STATISTICAL NORMALISATION OF NETWORK PROPAGATION METHODS FOR COMPUTATIONAL BIOLOGY

AUTHOR: SERGIO PICART-ARMADA
ADVISOR: ALEXANDRE PERERA-LLUNA

A thesis by compendium of publications
submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy
in Biomedical Engineering

in the

B2SLab
Centre de Recerca en Enginyeria Biomèdica
Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial
Universitat Politècnica de Catalunya

February 2020

ABSTRACT

STATISTICAL NORMALISATION OF NETWORK PROPAGATION METHODS FOR COMPUTATIONAL BIOLOGY

SERGIO PICART-ARMADA

The advent of high-throughput technologies and their decreasing cost have fostered the creation of a rich ecosystem of public database resources with molecular annotations and experimental data. In an era of affordable data acquisition and abundant pre-processing tools, the core challenge has shifted to improve data interpretation through algorithms and computational tools. The understanding of normal and disease states is a fundamental piece for generating novel and valuable biological insights. To that end, leveraging the current contextual knowledge in the form of annotations and biological networks can result in a powerful data amplifier and elucidate novel patterns and hypotheses.

Label propagation and diffusion are the cornerstone of the state of the art in network mining. They are driven by the *guilt by association* principle, which states that two interacting biological entities are prone to share functions and properties. In its simplest form, propagation algorithms predict the labels of a given node (for instance a gene, protein or metabolite) using those of its interactors. More elaborated approaches propagate beyond direct interactors, with robust performance in many areas within computational biology.

It has been pointed out that the topological structure of biological networks can bias propagation algorithms in such a way that best described entities experience a systemmatic advantage. Poorly known entities are therefore overlooked and harder to link to experimental findings, which in turn keeps them barely annotated. Some efforts try to break this circularity by statistically normalising the topological bias, albeit the properties of the bias and the real benefit of its removal are yet to be carefully examined.

The present thesis covers two general blocks. First of all, it seeks a proper characterisation of the bias in diffusion-based algorithms. Statistical normalisations are suggested, implemented and distributed within a scientific software package. The second block covers the application of such normalisation in classical computational biology problems that can be tackled from the network propagation standpoint. In particular, in biological pathway analysis for metabolomics data and in target gene prediction for drug development.

In the first block, the presence of the bias is confirmed and linked to the network topology, albeit dependent on which nodes have labels. Some equivalences are proven between diffusion processes with variations on their definitions, therefore easing its choice. Closed forms on the first and second statistical moments of the null distribution of the diffusion scores are provided, with resemblance to the spectral features of the network. Another

finding is that the normalisation can be detrimental in certain scenarios, e.g. if the bias favours nodes with positive labels. An ad-hoc study of the data and the expected properties of the findings is recommended for an optimal choice. To that end, this thesis contributes the *diffuStats* software package to a public repository. *diffuStats* eases the computation and benchmark of several diffusion scores, including normalised and unnormalised ones.

The second block starts with pathway analysis for metabolomics data. This choice is driven by the relative lack of computational solutions for metabolomics for being a younger discipline. The classical *over representation analysis* starts from a list of metabolites of interest, typically derived from an experimental study, and highlights a list of relevant biological pathways. Newer tools also use metabolic network data in their layout, but the interpretation still entails a demanding manual ad-hoc effort.

This block focuses on an enhanced interpretability by building and mining a richer knowledge network. The network connects the metabolites to the biological pathways through intermediate entities, like reactions and enzymes. Given the metabolites of interest, a propagation process is run to prioritise a relevant sub-network, suitable for manual inspection. The statistical normalisation is required due to the network design and properties. The usefulness of this approach is proven not only regarding pathway findings, but also examining the metabolites and reactions within the suggested sub-networks. The knowledge network construction and the propagation algorithm are distributed in the *FELLA* software package, with six case studies on human and animal datasets.

The second practical application is the prediction of plausible gene targets in disease by leveraging biological networks. Besides benchmarking the effect of the statistical normalisation on label propagation, particular care is put into obtaining meaningful performance estimates for practical drug development. Target data is usually known at the protein complex -or even family- level. Studies that overlook the structure of the protein complex data report overly optimistic performance estimates. In this thesis, this effect is corrected in an exhaustive comparison of prioritisation algorithms, networks, performance metrics and diseases. The results support that the statistical normalisation has a small but negative impact. In broad terms, even after correcting for the protein complex bias, network-based algorithms are still deemed useful and encouraged for drug discovery.

NORMALIZACIÓN ESTADÍSTICA SOBRE LOS ALGORITMOS DE PROPAGACIÓN EN REDES PARA BIOLOGÍA COMPUTACIONAL

SERGIO PICART-ARMADA

La aparición de tecnologías experimentales de alto rendimiento ha propiciado la creación de un rico entorno de bases de datos que aglomeran todo tipo de anotaciones moleculares. Dada la creciente facilidad para la adquisición de datos en varios niveles moleculares, el reto central de la biología computacional ha virado hacia la interpretación de dicho volumen de datos. La comprensión de los procesos de normalidad y enfermedad involucrados en los cambios observados en los estudios experimentales es el motor que expande la frontera del conocimiento humano. Para ello, es fundamental aprovechar la herencia de conocimiento previo, recogido en las bases de datos en forma de anotaciones y redes biológicas, y minarlo en busca de nuevos patrones e hipótesis.

Los algoritmos más extendidos para extraer conocimiento de las redes biológicas son los denominados métodos de propagación y difusión. Su trasfondo es el principio de *culpa por asociación*, que postula que las entidades biológicas que mantienen relación o interacción son más propensas a compartir funciones y propiedades. Dichos algoritmos aprovechan las interacciones conocidas, en formato de red, para predecir propiedades de nodos (por ejemplo genes, proteínas o metabolitos) usando las propiedades de sus interactores.

Existe evidencia de que la estructura topológica de las redes sesga los algoritmos de propagación, de forma que los nodos mejor descritos gozan de una ventaja sistemática. Los nodos menos conocidos quedan en desventaja, se entorpece el descubrimiento de su implicación en los experimentos, a su vez perpetuando nuestro pobre conocimiento sobre ellos. La literatura ofrece algunos estudios donde se normaliza dicho efecto, pero las propiedades intrínsecas del sesgo y el beneficio real de dicha normalización requiere un estudio más detallado.

El objeto de esta tesis tiene dos vertientes. Primero, la caracterización de la estadística del sesgo en los algoritmos de propagación, la concepción de normalizaciones estadísticas y su distribución como software científico. Segundo, la aplicación de dicha normalización en problemas clásicos de biología computacional. Concretamente, en el análisis de vías biológicas para datos de metabolómica y en la predicción de genes como dianas terapéuticas en el desarrollo de fármacos. Ambos problemas son abordables mediante técnicas de propagación y, por lo tanto, potencialmente sensibles al efecto del sesgo topológico.

En el primer bloque, se corrobora la existencia del sesgo y su dependencia no sólo de la estructura de la red, sino de los nodos en los que se define la propagación. Se demuestran equivalencias matemáticas entre ciertas variaciones en la definición de la propagación, facilitando así su elección. Se proporcionan expresiones cerradas sobre los momentos estadísticos de la difusión y se halla una conexión con las propiedades espectrales de las redes. Un punto importante es que la normalización no siempre ayuda, y su aplicabilidad dependerá de cada caso particular y de las hipótesis sobre la

topología de los nodos que deben ser descubiertos. Para ello, esta tesis deja como resultado *diffuStats*, un software disponible en un repositorio púlico, que permite calcular y comparar la propagación con ciertas variantes, y con presencia o ausencia de normalización.

En el segundo bloque, se escoge el análisis de vías en metabolómica dada la relativa juventud de los estudios metabolómicos y, por ende, su falta de herramientas informáticas dedicadas. El análisis de vías clásico parte de una lista de metabolitos de interés, normalmente procedentes de un estudio, y reporta una lista de vías o procesos metabólicos estadísticamente relacionados con ellos. Algunas variantes usan redes de metabolitos para dar más contexto biológico, pero la interpretación de los datos sigue requiriendo un extenso esfuerzo manual.

La aportación de esta tesis es la creación de una red de conocimiento que relaciona los metabolitos con las vías a través de las entidades intermedias anotadas, como reacciones y enzimas. Sobre dicha red se aplican algoritmos de propagación para identificar las entidades más relacionadas con los metabolitos de interés. La normalización estadística es necesaria, dada la estructura y las características de la red. Se demuestra no sólo la coherencia de las vías metabólicas propuestas, sino la de los metabolitos y las reacciones priorizadas. La publicación del software *FELLA* proporciona la construcción de la red de conocimiento y el algoritmo de difusión a la comunidad científica. *FELLA* va acompañado de seis casos de aplicación en estudios humanos y animales.

Por otro lado, se aborda el problema de predicción de genes para dianas terapéuticas a través de redes biológicas. Además de probar el efecto de la normalización estadística, se pone énfasis en estimar el desempeño real esperado en un escenario de desarrollo de fármacos. Los datos de dianas terapéuticas no se suelen conocer al nivel de proteína sino al de complejo o familia de proteínas. La mayoría de estudios no lo tiene en cuenta, llegando a estimaciones optimistas sobre el desempeño esperado. En esta tesis se propone un estudio exhaustivo que corrige el efecto de los complejos de proteínas, compara algoritmos de propagación con distintas métricas de rendimiento por su informatividad y explora el rol de la red biológica y de la enfermedad en cuestión. Se demuestra que la normalización estadística tiene poco efecto en el desempeño y que, en general, los métodos de propagación siguen siendo útiles en el desarrollo de fármacos después de corregir las estimaciones optimistas de su rendimiento.

ACKNOWLEDGEMENTS

On the scientific principle behind this thesis:

La teva tesi és de trileros

— Alexandre Perera Lluna

On how to properly manage a scientific project:

la cencia no se ace sola
ahi que acerla

— @cientefico

On how (not) to compromise one's mental health:

Your closest collaborator is you six months ago,
but you don't reply to emails.

— Karl Broman

This doctoral thesis contains many fingerprints, one for every person who helped, supported or beared me throughout the process. Actually, here are some numbers: this is the second time I write this section; my first attempt led to 5,757 characters, forming 978 words in 68 sentences that thanked exactly 107 people. As it felt a bit overwhelming to read, it was heavily summarised, but I must express my sincerest gratitude to all of them. Thank you.

I feel fortunate for having been a part of the B2SLab in Barcelona under the guidance of Alex Perera. You are gifted both within and outside the scientific scope, coupled with a selfless, educational, caring and easy-going personality. You know, you could probably make a living out of your coaching skills! You have shaped my critical thinking into the scientist I am now. Thank you for your academic and personal support throughout these years.

My deepest thanks to Francesc too, who can be seen as my older scientific brother, as we share our scientific father. Your support has been priceless and even now I struggle to find the right words to thank you properly. Despite your departure from the B2SLab (a sad moment for me), you kept in touch and eventually arranged an enriching internship in Takeda. I look forward to keep learning from you, as a professional and as a friend, and I wish you the very best as a scientist and as a dad.

Another key person is Alfonso Buil, my supervisor in my recurrent scientific stays in the Sct. Hans institute, Denmark. You have been outstandingly hospitable and attentive. You taught me statistics, genetics, but especially to keep calm and carry on, owing to your smooth attitude. Off topic: one could easily add a joke about bones in the previous sentence. Anyway, thank you for making it possible.

One name that deserves a special place is Pere Caminal. Not only for being one of the fathers of bioengineering research in Spain, but for your unconditional kindness and personal help. You brought us together and still show genuine interest for each one of us. Thank you.

Warm regards to the B2SLab, both the old and the young generations, for creating a cozy working environment. I wish the best to Pol S. We had an interesting time working together and our paths may encounter again in the future. A special mention to Josep, who took over some of my personal projects with enthusiasm.

I had the opportunity to visit external research centers and enjoyed an enriching experience. I want to thank Oscar Yanes and his lab for hosting me in Reus and having the perseverance to guide me on the first (and bumpy) stages of my research career. My warmest regards to the people in Sct. Hans, Denmark. Thank you Xabi for being a good friend and integrating me in your social circles. Thank you Camellia for being a warm neighbour and for your exceptional indian cuisine. Thank you Wes for listening to boring and convoluted statistical stuff. I would also like to acknowledge the people involved in the GAIT project. I also keep good memories of the Takeda people. Special thanks to Ben for keeping in touch and leading a fruitful collaboration with GSK.

To my University friends, highlighting Guillem B. for regularly checking on me and sharing our thoughts between beer and beer, Pablo, for our long calls and your caring attitude, and to Ferran, for the good old times.

To Stack Overflow and Wikipedia, no need to explain. To Sandwichez, where I spent too many hours (and money). To the Penyafort team and to the Galileu community, led by Jas, for our fond memories.

To Alvaro N., for being a good listener, preparing amazing food and keeping a relaxed atmosphere with his intelligent humour. To my closest people, especially to Xavier S. and Guillem M., who have accompanied me since high school.

To my parents Manuel and Pilar, who did not hesitate for a moment about my capabilities. Thank you for looking after me and for your genuine interest in what I do, even if most of it looks weird and hard to understand. To my family, especially to my grandparents.

To Angela, probably the one I annoyed the most, but with whom I shared the best moments too. I owe my mental sanity to you. You just knew how to take my mind out of vicious circles. You helped me keeping a work-life balance, putting everything in perspective and understanding the value of certain things. Thank you for being there for me.

Wrapping up this chapter in my life feels somewhere between unreal and relief. I will miss some aspects of the PhD life. Time to move on. Thank you.

CONTENTS

1 INTRODUCTION

1.1 OMICS SCIENCES

1.1.1 Introduction

The Human Genome Project (HGP) (Venter et al., 2001) marked the beginning of large scale biomedical data collection for public research. HGP was based on an automated protocol for Sanger sequencing (Sanger et al., 1977), a first generation method based on electrophoretic separation of chain-termination products. The paradigm of sequencing shifted with the advent of the second generation methods, or Next-Generation Sequencing (NGS), a massively parallel sequencing of shorter reads (Voelkerding et al., 2009). NGS diversified into technologies like Illumina CRT for whole genome sequencing, RNA-seq for transcriptome profiling, ChIP-seq for protein-DNA interaction, ATAC-seq for chromatin accessibility and methyl-seq for methylated DNA regions (Goodwin et al., 2016). The decreasing costs and the growing amount of measurable genomic features has been a key factor for the thorough annotation of thousands of organisms.

The high-throughput revolution is not restricted to the study of genomic data, but also available to other genome-scale data. The term *omics* sciences has been coined to refer to such technologies, which study the totality (suffix "-ome") of their subject (Joyce and Palsson, 2006). Omics technologies enabled a paradigm shift over traditional studies, typically reductionist and hypothesis-driven, by acquiring all the data in an agnostic way and generating hypotheses from its analysis (Horgan and Kenny, 2011). Omics studies promoted *systems biology*, the study of complex biological systems as a whole. Figure 1 illustrates the subject of the main omics sciences, explained hereafter, within the cell.

The leverage of omics data has promoted the population of specialised and comprehensive annotation databases, whose ultimate goal is to achieve a better understanding of biology. Such databases bring the opportunity to contextualise ongoing experimental studies using known molecular interactions – a key step that serves as a quality control and helps formulating new testable hypotheses (Gomez-Cabrero et al., 2014).

1.1.2 Genomics

Genomics is a mature discipline that studies genome sequences and the information encoded within them (Joyce and Palsson, 2006). The *genome* is defined as the total DNA of a cell or organism. Genotyping technologies enabled Genome Wide Association Studies (GWAS), a mainstream genomics analysis that seeks polymorphisms (variations in specific sites in the DNA) associated with the phenotype of interest. GWAS have contributed in our

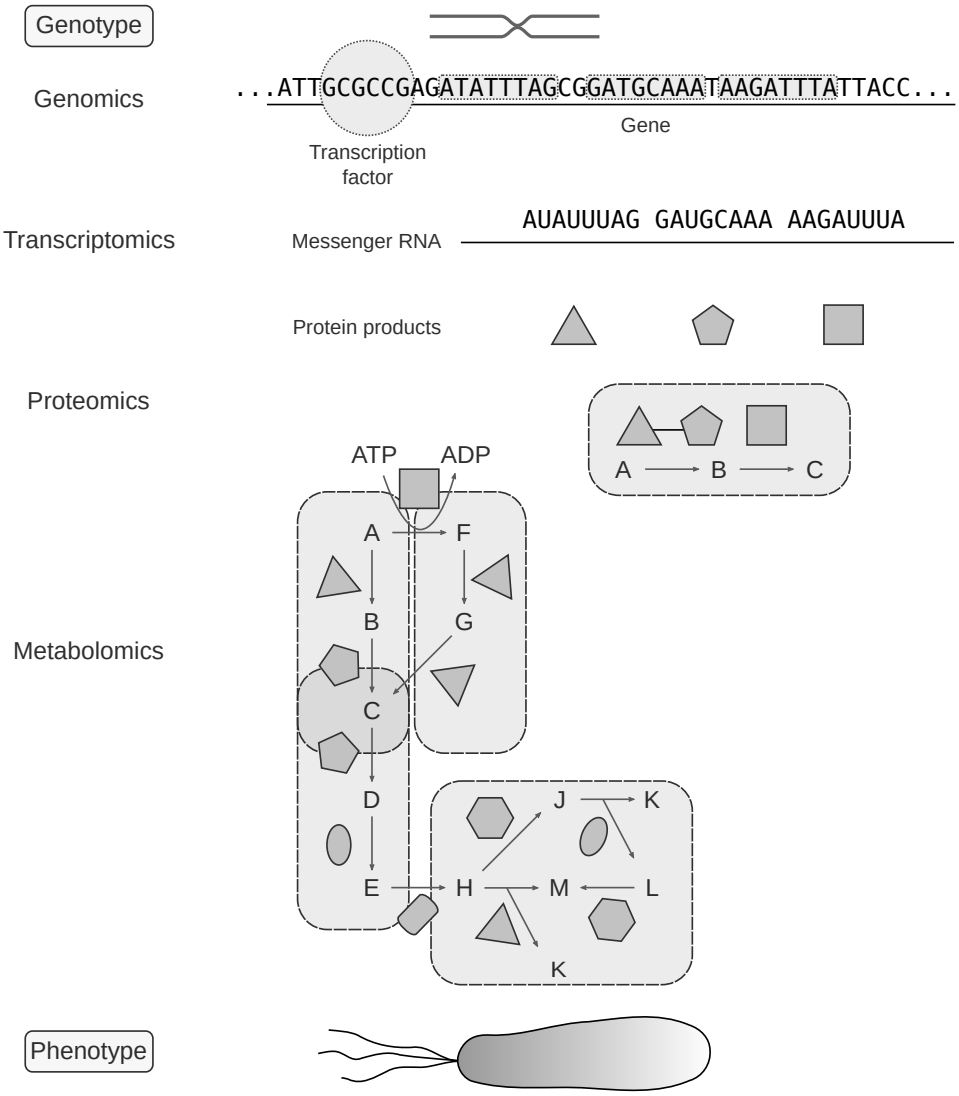


Figure 1: Overview of omics sciences. Genomics, trancriptomics, proteomics and metabolomics are the main omics sciences, whose measurements range from genotypic to phenotypic data. Figure adapted from 'Figure 1' in (Joyce and Palsson, 2006).

understanding of numerous complex traits through their findings (McCarthy et al., 2008). Currently, whole genome and whole exome (protein-coding regions) sequencing are improving our ability to discover genetic variants in human populations (Petersen et al., 2017).

1.1.3 Transcriptomics

Transcriptomics measures the presence and abundance of RNA transcripts (Joyce and Palsson, 2006). The total amount of messenger RNA in a cell or organism is called *transcriptome*. The assessment of differential gene expression was pioneering in the study of disease in the late 1990s and has generated a remarkable amount of biological knowledge. RNA-seq is a standard

technology to measure gene expression (Wang et al., 2009) and disposes of a rich array of tools to analyse its data. Despite being a valuable source of information, differences in transcript abundances do not necessarily imply the same changes at the protein level (Joyce and Palsson, 2006), meaning that transcriptomics alone is unable to explain the whole cellular state.

1.1.4 Proteomics

Proteomics focuses in identifing and quantifying the proteins within cells and tissues (Joyce and Palsson, 2006). The *proteome* is the set of all the expressed proteins in a cell or organism. Proteins orchestrate the metabolism, albeit their activity is in turn affected by the metabolic state of the cells. The proteome is a dynamic reflection of the combination of genetic and environmental factors and is considered an excellent source of disease biomarkers (Horgan and Kenny, 2011). Likewise, interaction events between proteins have been thoroughly studied and proven to be useful for applying network-based algorithms (Cowen et al., 2017).

1.1.5 Metabolomics

Metabolomics is the study of *metabolites*, the lightweight molecules that can be found within living organisms. The collection of metabolites found within cells or organisms is called the *metabolome*, also including those coming from the environment. Metabolite measurements are, in fact, quantitative phenotypes that give a snapshot of the functional readout of the cells (Joyce and Palsson, 2006). This is particularly appealing since it displays the actual effect of genomic or transcriptomic events. Compared with transcriptomics and proteomics, metabolomics data poses further statistical challenges due to its technical limitations (Joyce and Palsson, 2006), its physical and chemical complexity (Horgan and Kenny, 2011) and the unknown extent of the human metabolome (Wishart et al., 2012).

1.1.6 Other omics

Besides genomics, transcriptomics, proteomics and metabolomics, other omics sciences are emerging in terms of experimental techniques, public data and computational tools (Joyce and Palsson, 2006). *Metagenomics* is the analysis of genetic material from environmental samples, typically involving microbial communities. *Epigenomics* measures epigenetic events on the genetic material of cells, such as histone modification, DNA methylation and chromatin accessibility assays. The study of microRNA data is sometimes referred to as *miRNomics*. *Lipidomics* denotes the study of lipids, whereas *glycomics* revolves around carbohydrates and glycans. This list is not exhaustive but illustrative on the richness of current data acquisition and generation capabilities.

1.2 DATA INTERPRETATION

Leveraging omics data has provided key biological insights about normal and disease states, novel drug targets, drug response, biomarkers and predictive models for diagnosis and prognosis (Horgan and Kenny, 2011). The volume of high-throughput data generated in omics sciences has yielded high dimensional data that requires careful statistical treatment (Horgan and Kenny, 2011) and is challenging to interpret and understand (Joyce and Palsson, 2006). This issue first appeared with the advent of microarrays: a formal approach was needed to contextualise experimental results, usually extensive lists of differentially expressed genes.

The solution was named *functional analysis* and relied on classical statistical tests to assess if any known molecular function appeared more than expected within the gene list. Grouping genes that functioned in the same biological processes and finding dysregulated processes reduced the complexity of the data while providing richer mechanistical insights (Khatri et al., 2012). This is still a simple yet powerful approach, usually referred as *over representation analysis*, allowing the test of virtually any known annotation type. It plays a prominent role in translating differentially abundant genes, proteins or other molecular entities stemming from high-throughput technologies into biological knowledge (Mitrea et al., 2013), as illustrated in figure 2.

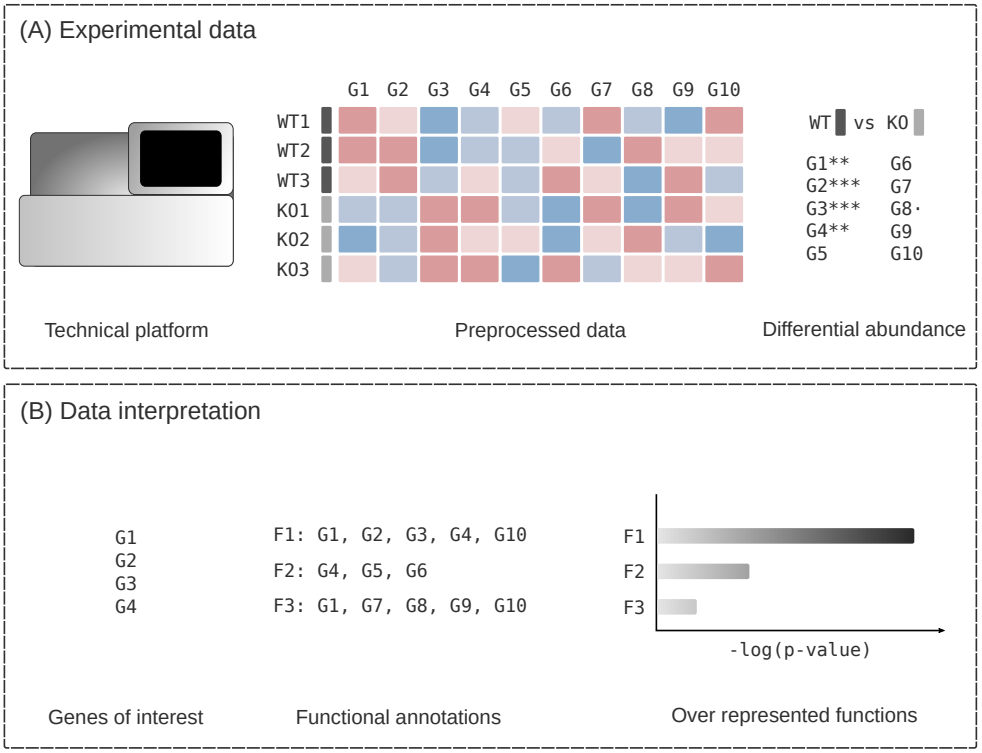


Figure 2: Examplary omics data analysis. This small example illustrates a hypothetical transcriptomics workflow in a case-control experiment. **(A)** Measurement of gene expression and discovery of differentially expressed genes between wild type (WT) and knockout (KO) experimental groups. **(B)** Discovery of highly occurring functions among the differentially expressed genes.

Data interpretation approaches have been extended to work on quantitative data and leverage biological network databases (Khatri et al., 2012), but the interpretability of their outputs is still an area of active research. The integration of different omics, which provide complementary views of a common reality, is a promising approach to attain a holistic picture of the molecular processes underlying disease (Ge et al., 2003).

REFERENCES

Cowen, Lenore, Trey Ideker, Benjamin J Raphael, and Roded Sharan
2017 "Network propagation: a universal amplifier of genetic associations", *Nature Reviews Genetics*, 18, 9, p. 551.

Ge, Hui, Albertha JM Walhout, and Marc Vidal
2003 "Integrating 'omic' information: a bridge between genomics and systems biology", *TRENDS in Genetics*, 19, 10, pp. 551-560.

Gomez-Cabrero, David, Imad Abugessaisa, Dieter Maier, Andrew Teschendorff, Matthias Merkenschlager, Andreas Gisel, Esteban Ballestar, Erik Bongcam-Rudloff, Ana Conesa, and Jesper Tegnér
2014 *Data integration in the era of omics: current and future challenges*.

Goodwin, Sara, John D McPherson, and W Richard McCombie
2016 "Coming of age: ten years of next-generation sequencing technologies", *Nature Reviews Genetics*, 17, 6, p. 333.

Horgan, Richard P and Louise C Kenny
2011 "'Omic' technologies: genomics, transcriptomics, proteomics and metabolomics", *The Obstetrician & Gynaecologist*, 13, 3, pp. 189-195.

Joyce, Andrew R and Bernhard Ø Palsson
2006 "The model organism as a system: integrating 'omics' data sets", *Nature reviews Molecular cell biology*, 7, 3, p. 198.

Khatri, Purvesh, Marina Sirota, and Atul J Butte
2012 "Ten years of pathway analysis: current approaches and outstanding challenges", *PLoS computational biology*, 8, 2, e1002375.

McCarthy, Mark I, Gonçalo R Abecasis, Lon R Cardon, David B Goldstein, Julian Little, John PA Ioannidis, and Joel N Hirschhorn
2008 "Genome-wide association studies for complex traits: consensus, uncertainty and challenges", *Nature reviews genetics*, 9, 5, p. 356.

Mitrea, Cristina, Zeinab Taghavi, Behzad Bokanizad, Samer Hanoudi, Rebecca Tagett, Michele Donato, Calin Voichita, and Sorin Draghici
2013 "Methods and approaches in the topology-based analysis of biological pathways", *Frontiers in physiology*, 4, p. 278.

Petersen, Britt-Sabina, Broder Fredrich, Marc P Hoeppner, David Ellinghaus, and Andre Franke
2017 "Opportunities and challenges of whole-genome and-exome sequencing", *BMC genetics*, 18, 1, p. 14.

Sanger, Frederick, Steven Nicklen, and Alan R Coulson
1977 "DNA sequencing with chain-terminating inhibitors", *Proceedings of the national academy of sciences*, 74, 12, pp. 5463-5467.

Venter, J Craig, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al.

2001 "The sequence of the human genome", *science*, 291, 5507, pp. 1304-1351.

Voelkerding, Karl V, Shale A Dames, and Jacob D Durtschi

2009 "Next-generation sequencing: from basic research to diagnostics", *Clinical chemistry*, 55, 4, pp. 641-658.

Wang, Zhong, Mark Gerstein, and Michael Snyder

2009 "RNA-Seq: a revolutionary tool for transcriptomics", *Nature reviews genetics*, 10, 1, p. 57.

Wishart, David S, Timothy Jewison, An Chi Guo, Michael Wilson, Craig Knox, Yifeng Liu, Yannick Djoumbou, Rupasri Mandal, Farid Aziat, Edison Dong, et al.

2012 "HMDB 3.0–the human metabolome database in 2013", *Nucleic acids research*, 41, D1, pp. D801-D807.

2 STATE OF THE ART

This chapter covers the concept of annotation databases, biological networks and propagation algorithms on them. Figure 3 pictures how contextual data, in network format, enriches experimental data by enabling predictions on a variety of domains. It also points out the logical order of the sections, whose topics are biological networks (data origin, definition and construction), propagation algorithms (graph theory definitions, the guilt-by-association principle, algorithm formulations) and two case studies (pathway analysis and disease gene prediction).

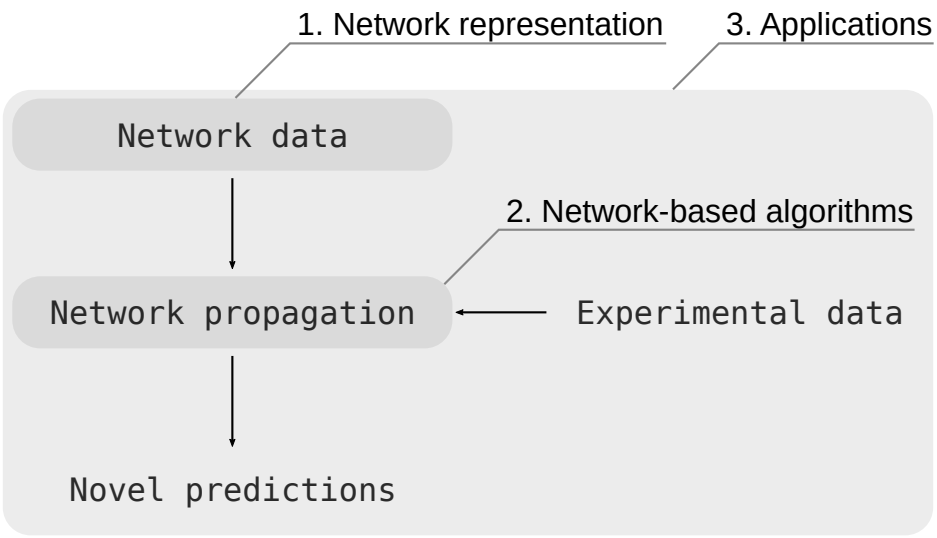


Figure 3: Overview of network propagation. Conceptual map of how network propagation takes advantage of network data to bring new insights from experimental data. The sections cover (1) the construction of biological networks from public data, (2) the definition and uses of network propagation algorithms and (3) two domains for the application of network propagation.

2.1 NETWORK REPRESENTATIONS IN BIOLOGY

Network data is a central concept in computational biology, both as a way to represent current knowledge and a corpus that enables new predictions. A proper knowledge representation is essential to provide sensible predictions through network propagation algorithms. Figure 4 displays these ideas, covered in this section. The logical order of the sections is as follows: section 2.1.1 introduces large projects that contribute abundant data to the public domain, section 2.1.2 covers specialised databases that annotate specific molecular levels, and both typically act as building blocks for network and

pathway databases (sections 2.1.3 and 2.1.4). Network-based algorithms can be applied to network data and to pathway annotations, provided that the latter are represented as networks.

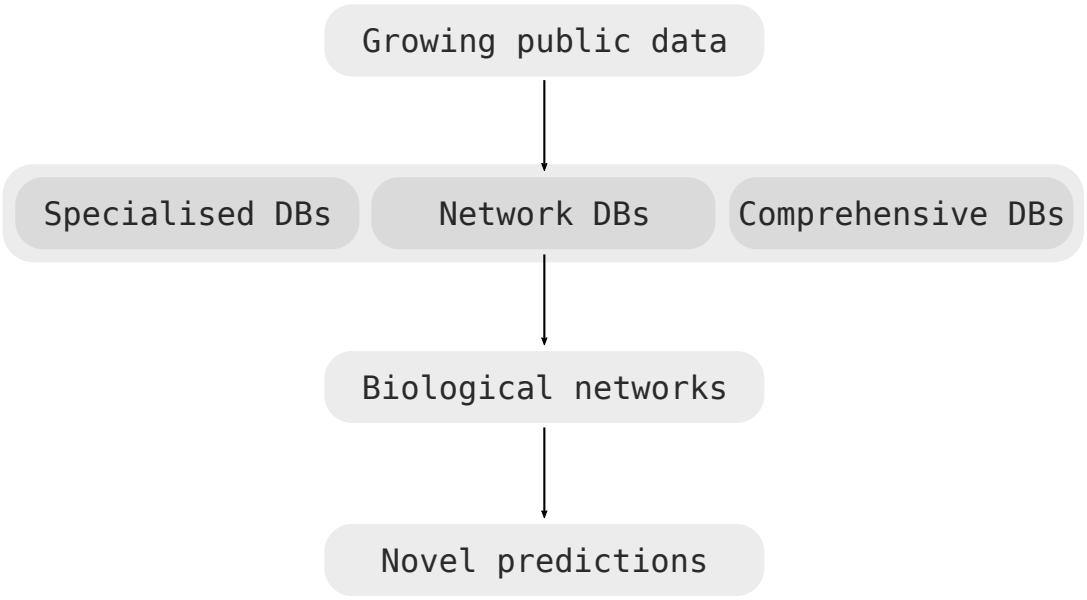


Figure 4: Overview of network representations. Conceptual map of public data is shaped into different types of database (DB), which in turn can be used to build biological networks. Network-based algorithms can be applied to biological networks to generate new knowledge.

2.1.1 Database resources

National and international consortia regularly foster large scale studies, which aim to translate large sample sizes into meaningful knowledge. This section contains a list of varied initiatives, to highlight the outstanding value of large scale data recollection.

One prominent example is the Encode project (ENCODE Project Consortium and others, 2012), aimed at annotating all the regions of the human genome throught the intregration of thousands of datasets. Likewise, the 1000 Genomes project (1000 Genomes Project Consortium and others, 2015) reconstructed 2,504 genomes from 26 populations in order to generate a high-quality reference panel of human genetic variation, annotating over 88 million variants. Another reference panel is provided by the UK10K project (UK10K consortium and others, 2015). Owing to the recording of several phenotypes, they further assess the contribution of genomic variation to a set of traits and causal mutations for disease. In the topic of mental disorders, the iPSYCH cohort (C. B. Pedersen et al., 2018) aims at finding novel genetic and environmental factors of conditions like schizophrenia, autism, attention-deficit/hyperactivity disorder or bipolar disorder. iPSYCH disposes of a Danish population-wide registry, granting an important statistical advantage.

The Genotype-Tissue Expression (GTEx) project (Lonsdale et al., 2013) is a systematic study on the effect of genetic variations on gene expression in

human tissues. Post-mortem samples were collected and analysed using genomics (whole genome sequencing) and transcriptomics (RNA sequencing). Scanned images and clinical data are also available. The incipient Human Cell Atlas (Regev et al., 2017) will explore cell types exhaustively with single cell technologies. Regarding oncology, The Cancer Genome Atlas (Tomczak et al., 2015) is a landmark cancer genomics program that characterised primary cancer and matched normal samples from 11,000 patients and 33 cancer types[1]. Genomic, epigenomic, transcriptomic, proteomic and clinical data was leveraged to publish marker manuscripts for each cancer type[2] and to elucidate key commonalities and differences across cancer types and tissues (Hoadley et al., 2014). The Connectivity Map, or CMap (Subramanian, Narayan, et al., 2017), aims at understanding cellular function by a large scale experiment on human cell lines with an array of perturbagens. Gene expression was quantified through L1000, a novel cost-effective profiling method that directly measures 978 landmark genes and accurately imputes 9,196 genes out of the 11,350 remaining transcripts. A total of 1,319,138 L1000 profiles are available and consolidated into 473,647 signatures that involve up to 77 cell lines.

Along with data from large initiatives, the number of scientific articles grows steadily too. Open science policies and protocols are gaining traction and encouraging data deposition on public online repositories with common protocols, such as Gene Expression Omnibus, or GEO (Clough and Barrett, 2016) and MetaboLights (Kale et al., 2016). Likewise, platforms like GitHub[3], Zenodo[4] and figshare[5] facilitate the storage of computer code and data.

Combining the increasing data and knowledge availability, there has been a need of annotation resources that centralise, format and curate data from the public domain. Two examples are the GWAS catalog (MacArthur et al., 2016) for genetic associations and the Expression Atlas (Papatheodorou et al., 2017) for transcriptomics studies. These efforts ease large-scale analyses and meta-studies, which leverage large sample sizes to unravel novel biological insights.

2.1.2 Specialised databases

A plethora of databases are publicly available, essentially at any molecular level ranging from genomic to phenotypic data. Table 1 displays a selection of such resources, some of which were already mentioned in section 2.1.1. Those cover, in order: genes, proteins, metabolites, compounds and phenotypes.

The Gene Ontology, or GO (G. O. Consortium, 2016) terms are a common choice to annotate gene function in three aspects: molecular function, cellular component and biological process. GO terms conform a hierarchy with

1 https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga/history. Accessed on 31/12/2019.
2 https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga/publications. Accessed on 31/12/2019.
3 https://github.com/. Accessed on 26/1/2020.
4 https://zenodo.org/. Accessed on 26/1/2020.
5 https://figshare.com/. Accessed on 26/1/2020.

Table 1: Selection of public databases covering a variety of biological data types. Their order ranges from genome-centered to phenotypic resources.

Resource name	Main subject	Reference
Ensembl	Genomes	(Zerbino et al., 2017)
GWAS catalog	Polymorphisms	(MacArthur et al., 2016)
Open Targets	Genetics and drugs	(Koscielny et al., 2016)
Gene Ontology	Gene products	(G. O. Consortium, 2016)
miRBase	MicroRNA	(Kozomara et al., 2018)
Expression Atlas	Gene expression studies	(Papatheodorou et al., 2017)
UniProt	Proteins	(U. Consortium, 2018)
Pfam	Protein families	(El-Gebali et al., 2018)
Brenda	Enzymes	(Jeske et al., 2018)
Human Metabolome Database	Metabolites	(Wishart, Feunang, Marcu, et al., 2017)
MetaboLights	Metabolomics studies	(Kale et al., 2016)
ChEMBL	Compounds	(Mendez et al., 2019)
DrugBank	Compounds	(Wishart, Feunang, An C Guo, et al., 2017)
Human Phenotype Ontology	Phenotypes	(Köhler et al., 2018)
Online Mendelian Inheritance in Man	Genetic phenotypes	(Amberger et al., 2018)

varying degrees of granularity (see figure 5). GO annotations involve a gene and a GO term. By the end of 2016, around 600,000 annotations derived from experimental evidence in 140,000 published articles, whereas 6 million stemmed from phylogenetic or computational inference (G. O. Consortium, 2016). Anontations for microRNA sequences are distributed in miRBase (Kozomara et al., 2018) and mapped to GO terms. miRBase v22 documents 271 organisms with 38,589 hairpin precursors and 48,860 mature microRNAs (from which 500 link to 5,000 GO terms).

Genetic associations can also be found at the genomic and transcriptomic level. For these purposes, the GWAS catalog (MacArthur et al., 2016) and the Expression Atlas (Papatheodorou et al., 2017) aggregate and curate public studies. The GWAS catalog encompassed 24,218 associations from 2,518 publications as for September 1st, 2016. The Expression Atlas contained 3,126 studies across 33 organisms by August 2017. The Open Targets platform (Koscielny et al., 2016), on the other hand, commits to the association between drug targets and diseases. The association strength is determined by combining several data sources, including GWAS catalog and Expression Atlas.

Protein data is annotated in resources like Uniprot (U. Consortium, 2018), with over 120 million proteins by 2018, and Pfam (El-Gebali et al., 2018), with data on 17,929 protein families in its release 32.0. More specifically, enzymes have dedicated databases, like the Brenda database (Jeske et al., 2018).

Metabolic signatures and reactions can be found in the Human Metabolome Database (Wishart, Feunang, Marcu, et al., 2017). Its metabolites are divided in four categories in release 4.0, in decreasing confidence: detected and quantified (18,557), detected but not quantified (3,271), expected metabolites (82,274) that have a known role and structure but await a formal identification, and predicted *in silico* (9,548).

For drug development, resources like ChEMBL (Mendez et al., 2019) and DrugBank (Wishart, Feunang, An C Guo, et al., 2017) gather activity data for small molecules. ChEMBL nourishes from Medicinal Chemistry journals and from clinical development and drug approval data. More than 15 million bioactivity measures for 1.8 million compounds on 3,600 organisms

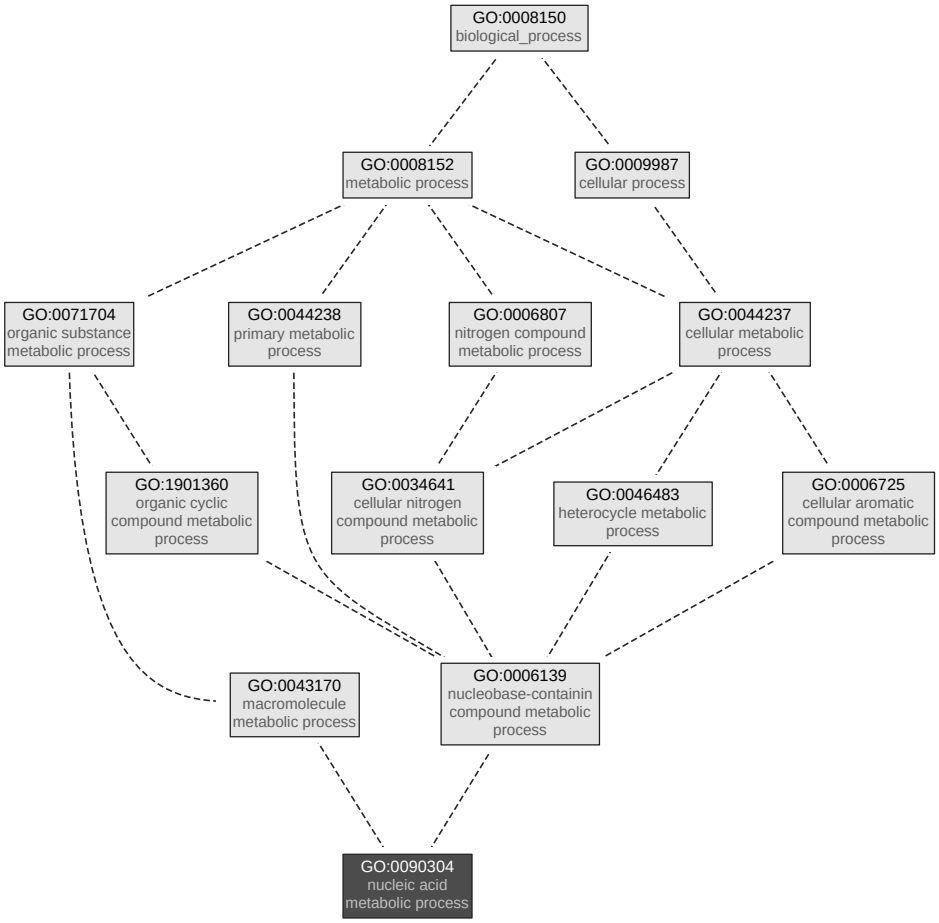


Figure 5: Gene ontology hierarchy. The GO term *nucleic acid metabolic process* (GO:0090304), inside a shaded frame, and all its parent terms. Terms deeper within the hierarchy refer to more specific annotations. The root term indicates that our GO term belongs to the *biological process* ontology (GO:0008150). Modified from a vector graphic file downloaded from `http://amigo.geneontology.org` by 7/3/2019.

have been extracted from over 67,000 publications and patents in ChEMBL release 24. Records feature 5,354 compounds that reached at least phase I clinical trials and 2,715 approved drugs. DrugBank 5.0 contains 2,358 approved drugs, 4,501 investigational drugs in phases I-II-III and 365,984 drug-drug interactions from 27,572 publications. Annotations include 4,563 drug targets (proteins, RNA, DNA and other molecules) and how their levels are modified by hundreds of drugs.

Phenotypic data is readily available as well. The Human Phenotype Ontology (Köhler et al., 2018) has created a standardised vocabulary to build computable definitions of over 7,000 diseases as September 2018. The Online Mendelian Inheritance in Man (Amberger et al., 2018) is an alternative in the field of medical genetics gathering over 24,600 entries, with a morbidity map linking 6,259 molecular phenotypes to 3,961 genes as September 2018.

2.1.3 Network databases

Annotations can often be regarded as connections between molecular entities. A natural way to understand them is as parts of a biological network, with the advantage of enabling automatic data mining through network-based algorithms (Carter et al., 2013). Network data is available for the majority of omics; table 2 provides a brief overview on several resources and their network type. The logical order of the networks in this section ranges from genotype to phenotype (genetics, gene regulatory events, gene expression, protein interactions, drugs and metabolism), although some integrative resources would fit in several categories. This classification is orientative and flexible, as many databases in table 1 can be understood from the network standpoint.

Table 2: Main network resources covering genetic, protein and metabolic data.

Resource name	Network connections	Main source	Reference
GIANT	Gene-gene regulation and interaction	Integrative	(Greene et al., 2015)
GTEx	Gene-gene co-expression	GTEx data	(Pierson et al., 2015)
HumanNet	Gene-gene relationships	Integrative	(Hwang et al., 2018)
DisGeNET	Gene/variant-disease associations	Integrative	(Piñero et al., 2016)
HMDD	MicroRNA-disease associations	Literature	(Z. Huang et al., 2018)
TRRUST	Transcription factor-target regulation	Literature	(H. Han et al., 2017)
BioGRID	Protein/drug-protein interactions	Literature	(Chatr-Aryamontri et al., 2017)
STRING	Protein-protein interaction	Integrative	(Szklarczyk, Gable, et al., 2018)
HIPPIE	Protein-protein interactions	Integrative	(Alanis-Lobato et al., 2016)
OmniPath	Protein-protein interactions	Integrative	(Türei et al., 2016)
STITCH	Protein-chemical interaction	Integrative	(Szklarczyk, Santos, et al., 2015)
CMap	Gene-drug-disease associations	L1000 data	(Subramanian, Narayan, et al., 2017)
Recon	Metabolite-reaction models	Integrative	(Swainston et al., 2016)

On the genetic scope, the GIANT project (Greene et al., 2015) provides a genome-scale integration of more than 61,400 experiments in 24,930 publications in its 2.0 release. Data from 283 tissues and cell types is acccessible for queries, visualisation and complementing quantitative genetics data. The added value of abundant tissue granularity is the potential of revealing finer, tissue-specific mechanisms though data mining algorithms.

As for transcriptomics, the RNA sequencing data collected in the GTEx project (Lonsdale et al., 2013) has enabled the inference of tissue-specific gene co-expression networks (Pierson et al., 2015) from the pilot data. Using a hierarchical algorithm for sharing data between related tissues, a total of 35 tissues have been released and analysed with regard to the topological properties of transcription factors and tissue-specific functional genes.

HumanNet (Hwang et al., 2018) provides a hierarchy of human disease networks with functional associations between human genes. Those are obtained through a Bayesian integration of varied omics data and resources: protein-protein interactions, co-citation, co-occurrence of protein domains, co-expression of genes, genomic context associations and interactions of evolutionary conserved proteins in model organisms. The fully extended HumanNet v2 hierarchy, HumanNet-XN, encompasses 17,929 genes and 525,537 links between them.

DisGeNET (Piñero et al., 2016), on the other hand, specialises on genotype-phenotype annotation. Several human, animal and chemical resources are fused in order to link genes to a controlled disease vocabulary and ontol-

ogy. Specifically, DisGeNET 4.0 connects 17,381 genes to 15,093 diseases in the form of 429,036 gene-disease associations, stemming from more than 289,000 scientific publications.

Driven by the growing evidence that microRNA elements carry regulatory roles in disease states, the HMDD database (Z. Huang et al., 2018) manually curates and incorporates microRNA-disease associations from the literature. HMDD v3.0 contains 32,281 microRNA-disease annotations supported by experimental evidence in 17,412 articles, involving 1,102 microRNAs and 850 diseases.

A complementary transcriptional regulation data source derives from the study of transcription factors. The TRRUST reference database (H. Han et al., 2017) aggregates connections of the form transcription factor-target, manually curated after an initial filtering by sentence-based text mining on the MEDLINE® database[6] abstracts. TRRUST v2 includes 8,444 regulatory interactions from 800 transcription factors in humans; respectively, 6,552 and 828 in mouse.

Several resources annotate protein interactions, either directly curating the literature or integrating primary sources offering all sorts of protein and gene-level annotations. BioGRID (Chatr-Aryamontri et al., 2017) is an example of the former: after a text mining step, expert curators extract genetic and protein interactions from peer-reviewed journals. BioGRID 3.4.140 (September 2016) gathers 470,810 protein interactions and 373,762 genetic interactions, both non-redundant, from 47,223 publications on 66 organisms. BioGRID is also populated with 38,559 connections from proteins to post-translational modification sites, and with 27,034 protein-chemical interactions from DrugBank (Wishart, Feunang, An C Guo, et al., 2017).

In contrast, STRING (Szklarczyk, Gable, et al., 2018) is a widely adopted protein-protein interaction resource that aggregates other resources, in the form of evidence codes, to provide a confidence about the veracity of the interactions. Figure 6 contains 10 interactors of the exonuclease EXD2 and illustrates some data channels: co-expression, text-mining, biochemical/genetic experimental data, previously curated pathway and protein-complex knowledge. Consequently, STRING attains a high coverage, currently of about 24.6 million proteins from 5,090 organisms in its version 11.0, and has held a robust performance in a recent benchmark (J. K. Huang et al., 2018).

HIPPIE (Alanis-Lobato et al., 2016), an integrated protein-protein interaction resource, focuses on building a thorough, context-rich and reliable representation of the human interactome. Gene Ontology terms (G. O. Consortium, 2016) are used to provide biological process and cellular component annotations. Tissue-specific networks are obtained by leveraing gene expression data from the GTEx project (Lonsdale et al., 2013), by dropping the genes that lack expression in the tissue of interest. HIPPIE v2.0 (June 2016) includes approximately 273,900 experimental interactions -42,600 of high confidence- among 17,000 proteins.

Similarly, OmniPath (Türei et al., 2016) is an integrated network database for human protein interactions, nourishing from 27 sources. OmniPath is stringent, seeking causal mechanisms by only including high-confidence sig-

6 See MEDLINE/PubMed data in https://www.nlm.nih.gov/databases/download/pubmed_medline.html. Accessed 17/11/2019.

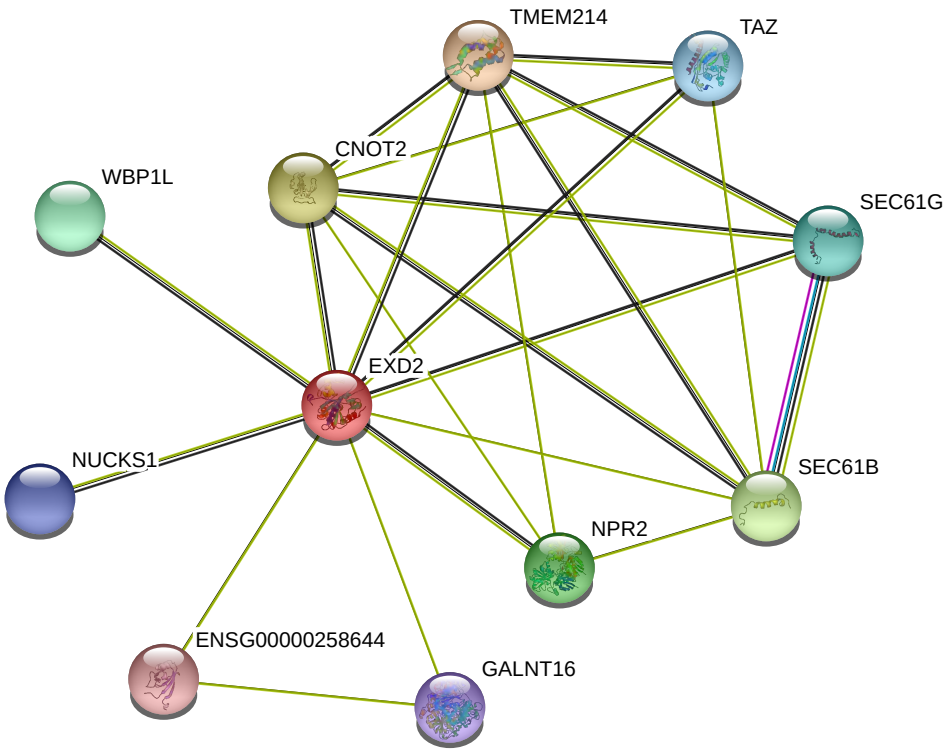


Figure 6: Small interactome from the STRING database. Each node represents a protein, with a thumbnail of its three-dimensional structure. Edges indicate associations through text mining (yellow), co-expression (black), curated databases (cyan) and experiments (magenta), with a confidence of at least 0.4 out of 1. Modified from a vector graphic file downloaded from `https://string-db.org` by 21/2/2019.

nalling interactions. OmniPath encompassed 7,984 proteins and 36,557 interactions from 41,237 references by November 2016[7].

The interactions between proteins and small chemicals is of outstanding interest for drug development. STITCH (Szklarczyk, Santos, et al., 2015) is a vast integrative resource for protein-chemical interactions. STITCH 5 involves around 9,600,000 proteins from 2,031 eukaryotic and prokaryotic organisms (shares the protein space with STRING v10) and 430,000 chemicals. More specifically, there are 4,740 high-confidence interactions between human proteins and chemicals.

The role of small molecules is also investigated at the gene expression level in CMap (Subramanian, Narayan, et al., 2017), already mentioned in section 2.1.1. Human cell lines are challenged with 19,811 compounds, 18,493 short hairpin RNAs, 3,462 complementary DNAs and 314 biologics. Novel mechanisms of action for small molecules can be elucidated by seeking similar or opposing gene expression signatures from perturbagens with known mechanisms. The CMap infrastructure can therefore be used to connect genes, drugs and diseases within or across cell lines throught their gene expression patterns.

Closing this list, the metabolome has also been the subject of network databases. Recon (Swainston et al., 2016) is a consensus global reconstruc-

7 `http://archive.omnipathdb.org/README.txt`. Accessed 17/11/2019.

tion of the human metabolism, intended to improve its computational modeling. Four metabolic resources are combined, obtaining 7,440 reactions that involve 5,063 metabolites. Furthermore, 65 cell type-specific models are provided through the intregration of protein expression data.

The vast number of resources, with varying scopes, data sources, curation and contextual information, poses a fundamental question of choice before applying the network-based algorithms described in section 2.2. A systematic benchmark of biological networks for the discovery of disease genes concludes that integrated networks such as GIANT and STRING are the best performers, and that the effects of broading the coverage outweight the extra false positive interactions (J. K. Huang et al., 2018). Another effort to address the resource heterogeneity is NDEx (Pratt et al., 2015), an online commons to centralise and publicly distribute molecular networks under a common programmatic interface.

Despite their successful application in computational biology, molecular networks come with limitations. Ascertainment bias is a major issue: the best studied genes and proteins are best represented in current networks, affecting any downstream network-based algorithm (Carter et al., 2013; W. Zhang et al., 2017). Molecular networks are generally incomplete (W. Zhang et al., 2017), in part due to biases in the experimental technologies (Carter et al., 2013). Contextual data -like protein isoforms, protein structural variations and cell populations- is still scarce, but necessary to understand disease mechanisms (W. Zhang et al., 2017).

2.1.4 Comprehensive databases

Comprehensive databases aim to annotate biological mechanisms in an exhaustive manner and understand biology at the systems level. Their linchpin is the concept of *biological pathway*, a delimited biological process typically involving proteins, reactions, metabolites, enzymes or genes (Bader et al., 2006). Biological pathways are human abstractions to describe and understand biological phenomena and stand as a central resource in computational biology (Wishart, Carin Li, et al., 2019). Linking experimental data to affected pathways is an essential part of the genomics, transcriptomics, proteomics (Khatri et al., 2012) and metabolomics (Chagoyen and Pazos, 2012) workflows.

Pathways can be understood as:

- Gene sets, i.e. lists of genes with a common function or association. Also applies to entities like proteins or metabolites.
- Network data, if the interactions between the molecular entities are available, possibly with directionality and metadata.

Conversely, not every biological network can be understood as a pathway.

- Pathways describe our mechanistic understanding of interaction and regulatory events.
- Certain networks contain observational data without any knowledge of the underlying biology. Gene co-expression networks are an example where connections (pairs of genes whose expression is highly

correlated) do not necessarily imply physical interaction or genetic regulation.

Table 3 displays a selection of pathway databases, further elaborated in this section and arranged according to their principal subject: genetic events, metabolism, signalling, disease and integrative.

Table 3: Selection of public pathway resources, sorted by their main focus: genetic, metabolic, general or integrative.

Resource name	Main subject	Reference
PANTHER	Genetics	(Mi et al., 2018)
SMPDB	Metabolic	(Jewison et al., 2013)
LIPID MAPS	Metabolic, lipids	(V. B. O'Donnell et al., 2019)
SwissLipids	Metabolic, lipids	(Aimo et al., 2015)
MetaCyc	signalling, metabolic	(Caspi et al., 2019)
KEGG	signalling, metabolic, disease	(Kanehisa et al., 2016)
Reactome	signalling, metabolic, disease	(Fabregat et al., 2017)
WikiPathways	signalling, metabolic, disease	(Slenter et al., 2017)
PathBank	signalling, metabolic, disease	(Wishart, Carin Li, et al., 2019)
Pathway Commons	Integrative: pathways, interactions	(Rodchenkov et al., 2019)
ConsensusPathDB	Integrative: pathways, interactions	(Herwig et al., 2016)
BioModels	Integrative: models	(Malik-Sheriff et al., 2019)

The PANTHER database (Protein ANalysis THrough Evolutionary Relationships) (Mi et al., 2018) provides evolutionary and functional annotations for genes in over 900 genomes. Besides nourishing from Gene Ontology (G. O. Consortium, 2016), PANTHER contains a collection of 177 pathways with 3,092 pathway components, 53,548 associated sequences and capturing 6,000 references (PANTHER™ Pathway 3.6.3, released December 2019).

Conversely, the SMPDB (Small Molecule Pathway Database) (Jewison et al., 2013) focuses in human pathways, for which small molecules play a central role. Its 2.0 version includes the following pathway types: metabolic (92), disease (221), drug action (232), drug metabolism (53), physiological action (5) and small molecule signalling (15). In most of them, the cellular location, tissue or organ where reactions take place are available.

Lipids encompass a fundamental part of the metabolism and contribute to its understanding, but the technical limitations hinder their characterisation (V. B. O'Donnell et al., 2019). The LIPID MAPS (Lipid Metabolites and Pathways Strategy) initiative (V. B. O'Donnell et al., 2019) categorised over 30,000 lipids from several organisms (Hartler, 2015) and contributed with 10 pathways, including the metabolism of cholesterol, eicosanoids, glycerolipids, omega fatty acids and sphingolipids[8]. Alternatively, SwissLipids (Aimo et al., 2015) features an *in silico* library of 244,155 feasible lipid structures, more than 2,000 curated enzymatic reactions linking to over 800 proteins and involving glycerophospholipids, glycerolipids, sphingolipids, sterols, fatty acids, fatty alcohols and wax esters (Aimo et al., 2015).

Certain pathway databases aim at a broader understanding of biological pathways, usually within multiple organisms, in the context of metabolism, signalling events, genetic regulation and disease states. They also rely on

8 http://www.lipidmaps.org/resources/pathways/index.php. Accessed 18/12/2019.

and link to multiple specialised databases for entities like sequences, proteins, enzymes, metabolites, lipids, drugs and diseases. MetaCyc (Caspi et al., 2019), the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al., 2016), Reactome (Fabregat et al., 2017) and WikiPathways (Slenter et al., 2017) are widespread public resources for this purpose. Commercial options include Ingenuity Pathway Analysis[9] and MetaCore[10] but are out of the scope of this thesis.

MetaCyc offers over 2,570 pathways based on experimental evidence from more than 54,000 publications, and involving 14,003 compounds and 15,691 reactions in version 21.1 (Caspi et al., 2019). Although MetaCyc mainly covers small molecule metabolism, the amount of data on macromolecular mecanism is increasing. Specifically, 35 species are annotated with 20 or more pathways (294 in Homo sapiens). By August 2017, almost 11,000 organism-specific semi-automatic metabolic networks were described in pathway/genome databases.

KEGG also offers curated organismal pathways, ranging from metabolic to signalling processes. KEGG contains the following database categories: systems information for pathway data, genomic information, chemical information for metabolites, KEGG LIGAND for reactions and enzymes, health information for diseases and KEGG MEDICUS for drugs (Kanehisa et al., 2016). As of October 2016, KEGG encompassed 496 manually drawn pathway reference maps. One example can be found in figure 7: the *photosynthesis* KEGG pathway, depicting its enzymatic reactions, metabolites and related pathways.

Reactome is a knowledge representation that describes human signal transduction, transport, DNA replication, metabolism in a single, consistent data structure (Fabregat et al., 2017). Reactome version 62 covers 10,719 genes, 24,704 protein forms (including post-translational modifications and cellular localisations), 1,768 metabolites and 11,302 reactions drawn from 27,526 scientific articles. 2,012 human pathways are divided into 26 superpathways that represent broad biological domains. Disease annotations are present in the form of 906 disease-specific reactions, annotated from 1,334 mutated variants in 285 gene products.

Wikipathways, on the other hand, is based on a crowdsourcing paradigm to annotate biological pathway data (Slenter et al., 2017). WikiPathways focuses on 25 reference species, annotating 2,614 pathways that were contributed by 634 individuals by September 2017. These involve 11,532 genes (7,982 related to metabolic processes) and 3,133 metabolites, with an increasing effort to improve the coverage of the latter.

PathBank (Wishart, Carin Li, et al., 2019) is a recent effort on 10 model organisms to provide a pathway for every protein and a map for every metabolite. Over 110,000 pathways containing 78,488 compounds (including metabolites and drugs), 8,993 proteins and 176,535 reactions/interactions are available for metabolism, signalling, disease, drugs and physiology. Their metadata provide subcellular locations, cofactors and protein quaternary structures. PathBank aims at improving the coverage of lipid syn-

9 https://www.qiagenbioinformatics.com/products/ingenuity-pathway-analysis/. Accessed 26/01/2020.

10 https://portal.genego.com/. Accessed 26/01/2020.

thesis, metabolite signalling, small molecule hormone signalling and small molecule drug action in KEGG and MetaCyc, while keeping the coverage of protein and cellular signalling from Reactome and Wikipathways.

Integrative efforts have emerged to ease searching, downloading, querying, browsing and analysing a collection of varied pathway databases. Pathway Commons (Rodchenkov et al., 2019) aggregates 22 public databases (including PANTHER, KEGG, Reactome and WikiPathways) into 4,794 human pathways (18,490 genes and 11,437 metabolites) and 2.3 milion interactions by February 2019. Likewise, ConsensusPathDB (Herwig et al., 2016) integrates 32 databases for human, 15 for mouse and 14 for yeast. In humans, 158,523 physical entities are annotated with 458,570 interactions and conform 4,593 pathway gene sets. ConsensusPathDB further allows tasks such as pathway analysis, heterogeneous network inference and module analysis, starting from genome-wide data or priority lists of genes, proteins or metabolites. BioModels (Malik-Sheriff et al., 2019) is based on a systems biology approach, storing about 2,000 mathematical models from the literature. Such models can predict the states of biological systems, ease the elaboration of novel hypotheses and improve our mechanistic understanding. Models on cell signalling, metabolic pathways and gene regulation are also contextualised with cross-references to standard data resources using machine-friendly controlled vocabularies.

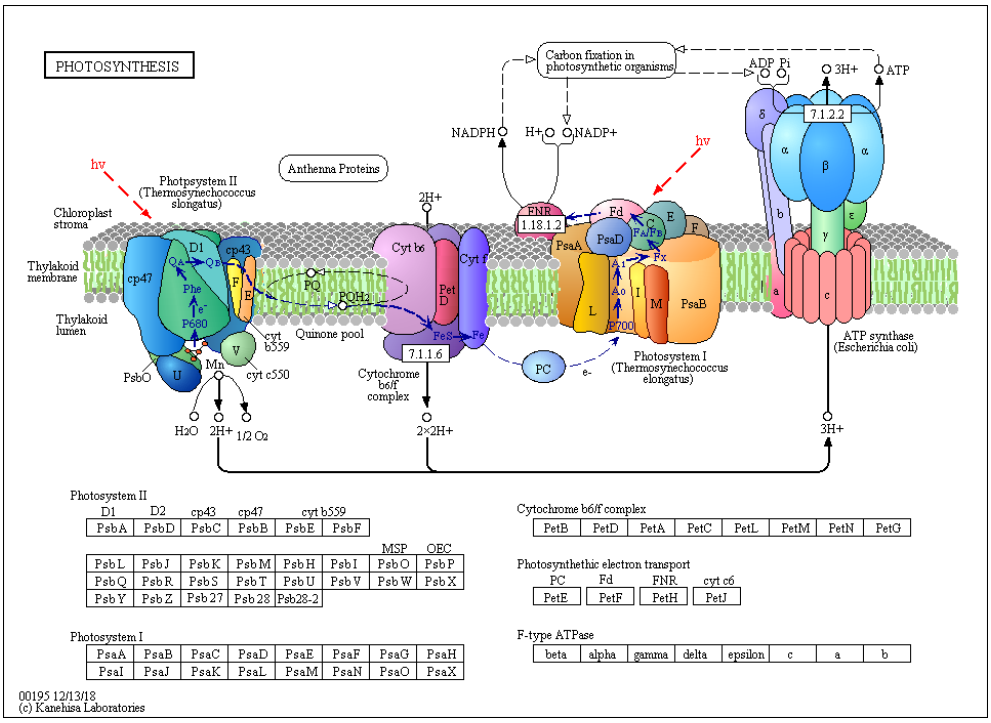


Figure 7: The photosynthesis KEGG pathway (identifier: map00195). Metabolites are represented by circles, reactions by arrows and their enzymes by superposed rectangles. Other neighbouring pathways are visible, like *Carbon fixation in photosynthetic organisms*. Note how also gene/protein names for ortholog groups are provided. Image downloaded from https://www.genome.jp by 9/1/2019.

Despite is usefulness, leveraging biological pathway data suffers from limitations. Pathways are under continuous construction and considered to

be highly incomplete (Ogris et al., 2016), limiting the statistical power of pathway-based approaches. Maintaining pathway databases is demanding, leading to an inability to keep up with the growing liteature or even to discontinuation (Rodchenkov et al., 2019). Another common critique arises from the fact that manual curation results in artificial borders between biological pathways. Consequently, notable differences exist between major pathway databases, in terms of focus, coverage, granularity and pathway definition (Domingo-Fernández et al., 2018), implying that the database choice has a considerable impact in any downstream analysis. In this context, the emergence of integrative resources provide a proxy to alleviate database-related biases in data mining. However, the degree of overlap, complementarity, pathway cross-talk and even disagreement needs careful investigation (Domingo-Fernandez et al., 2019). The PathMe platform (Domingo-Fernandez et al., 2019) is a pioneering effort to harmonise and understand the merging of the human data in KEGG, Reactome and WikiPathways under a common controlled vocabuary. Considering the wide spectrum of databases and organisms, there is still room to scale up pathway database harmonisation.

2.2 NETWORK PROPAGATION ALGORITHMS

Once network data is available, network propagation allows the integration of experimental (or annotated) data with contextual knowledge. This section, conceptualised in figure 8, covers the mathematical definition of the networks, the algorithms applied on them (through specific examples in computational biology) and the examination of their statistical properties.

2.2.1 Introduction to network propagation

High-throughput techniques are contributing in the prediction and identification of a vast collection of molecular interactions. This has led to a rich variety of biological network resources (section 2.1.3), such as protein-protein interaction, gene regulatory, co-expression and metabolic networks (J. K. Huang et al., 2018). Such networks are usually defined as a set of nodes and a set of edges that connect pairs of nodes. For example, nodes can be proteins and edges can be experimentally proven interactions. Both nodes and edges can have additional attributes, like edge directionality and weight.

On the other hand, the *guilt by association principle* states that interacting entities are more prone to share molecular functions. This basic concept has found ubiquitous application to problems like protein function prediction, disease gene prioritisation (figure 2B), module inference, cancer patients stratification, drug discovery and causal variants identification (Cowen et al., 2017).

The paradigm behind network propagation is to infer the labels of molecular entities using the neighbouring connections from a biological network and a set of known, labelled entities. The simplest approach, *neighbour vot-*

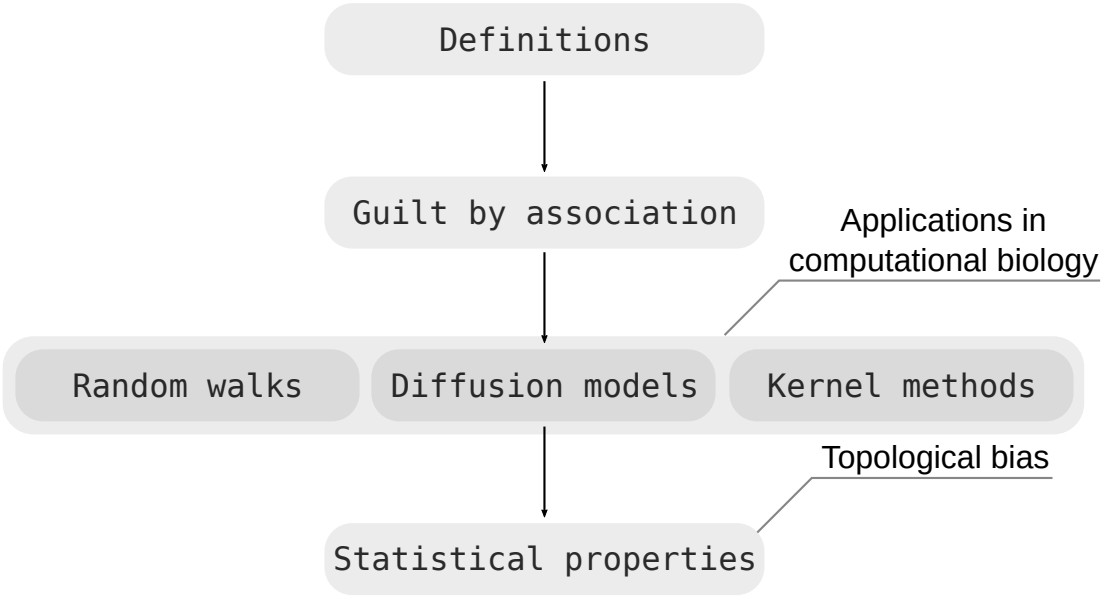


Figure 8: Overview of network propagation algorithms. Conceptual map of the section layout. Relying on basic network definitions, the *guilt by association principle* justifies the concept of network propagation, which accepts a variety of formulations based on random walks, heat (or another abstract entity) diffusion or graph kernels. The statistical description of the predictions of such methods raised concerns about the presence of a topological bias, i.e. related to the intrinsic properties of the networks.

ing, infers the label of an entity from the known labels of its neighbours. A use case could be as follows: to infer whether a protein is related to the obesity phenotype, one counts the proportion of obesity-related interacting proteins. Proteins with the highest proportions are suggested as potential associations.

More sophisticated *network propagation* or *diffusion* approaches allow the propagation to reach beyond direct neighbours and attain competitive performances in many computational biology applications (Cowen et al., 2017). Following the case study of obesity, one can conceive an abstract substance (heat, fluid, current) that flows from the obesity-related proteins to the rest of the network through the edges. The degree of association of each protein is then measured by the substance received at every protein.

In this regard, the diversity of problem formulations and computational biology applications is notable, sometimes causing the re-discovery of equivalent methods under different names and domains. The term "network propagation" therefore stands for a general purpose, heterogeneous but unifying formalism for network analysis.

2.2.2 Introduction to graph theory

This section covers basic notions on graph theory prior to the introduction of propagation methods in section 2.2.3. These definitions will be referenced throughout this thesis.

Defining a graph

The mathematical definition of a network or graph G, adapted from (Smola and Kondor, 2003), consists of two sets:

$$G = (V, E) \tag{1}$$

V is the set of vertices, or nodes, typically numbered from 1 to n. The number n of nodes in the graph is called the graph *order*. E is the set of edges, consisting of pairs of nodes (i, j) that indicate that there is a connection from i to j, meaning that they are *neighbours*. Here, only finite graphs without *multiple edges* (i.e. 'repeated') or *loops* (edges of the kind (i, i)) are considered. $G' = (V', E')$ is a *subgraph* of G if G' is a graph with $V' \subseteq V$ and $E' \subseteq E$.

If G is *weighted*, each edge (i, j) comes with a weight $W_{ij} \in \mathbb{R}$. The methods hereby discussed assume $W_{ij} \geqslant 0$: the greater the weight, the easier it is to traverse the edge. If G is *unweighted*, $W_{ij} = 1$ if $(i, j) \in E$ and $W_{ij} = 0$ otherwise.

If G is *undirected*, the edges (i, j) and (j, i) are identical and usually denoted as unordered pairs $\{i, j\}$, and $W_{ij} = W_{ji}$. Conversely, in a *directed* graph, the edge (i, j) can only be traversed from i to j and does not imply the existence of the edge (j, i).

A *simple* graph is an unweighted, undirected graph without multiple edges or loops (Diestel, 2000).

The $n \times n$ real matrix W is called the *adjacency matrix* of G. The *degree matrix* of an undirected graph G is the diagonal matrix D with $D_{ii} = \sum_{j=1}^{n} W_{ij}$. D_{ii} is called the *degree* of vertex i. Note that if G is directed, one can either use the *in-degree* $D_{ii} = \sum_{j=1}^{n} W_{ji}$ or the *out-degree* $D_{ii} = \sum_{j=1}^{n} W_{ij}$ (Bang-Jensen and Gutin, 2008). If G is simple, D_{ii} is the number of neighbours of the i-th node. Nodes with a high amount of neighbours are called *hubs* (Cowen et al., 2017), whereas nodes with no connections are *isolated*. Examples can be found in figure 9.

Walks, paths and connectivity

This section is based on the definitions in (Diestel, 2000) for simple graphs. A *walk* in a graph $G = (V, E)$ is a sequence of alternating nodes $v_i \in V$ and edges $e_j \in E$, represented as $v_1, e_1, v_2, e_2, \ldots, e_{k-1}, v_k$, starting and ending on nodes, with $e_l = (v_l, v_{l+1})$, i.e. every edge connects the nodes before and after it. A *path* is a walk without repeating nodes and its *length* is the number of edges in it.

A *shortest path* between two nodes v_i and v_j is a path whose length is minimum among those that start at v_i and end at v_j. If such a path exists, its length is called the *shortest path distance* $d_G(v_i, v_j)$ between v_i and v_j, which are denoted as *connected*. Otherwise, if no path exists between v_i and v_j, then $d_G(v_i, v_j) = \infty$ by convention and they are *disconnected*.

A simple graph G is *connected* if every possible pair of its nodes is connected. A *connected component* of a graph G is a maximal connected subgraph of G.

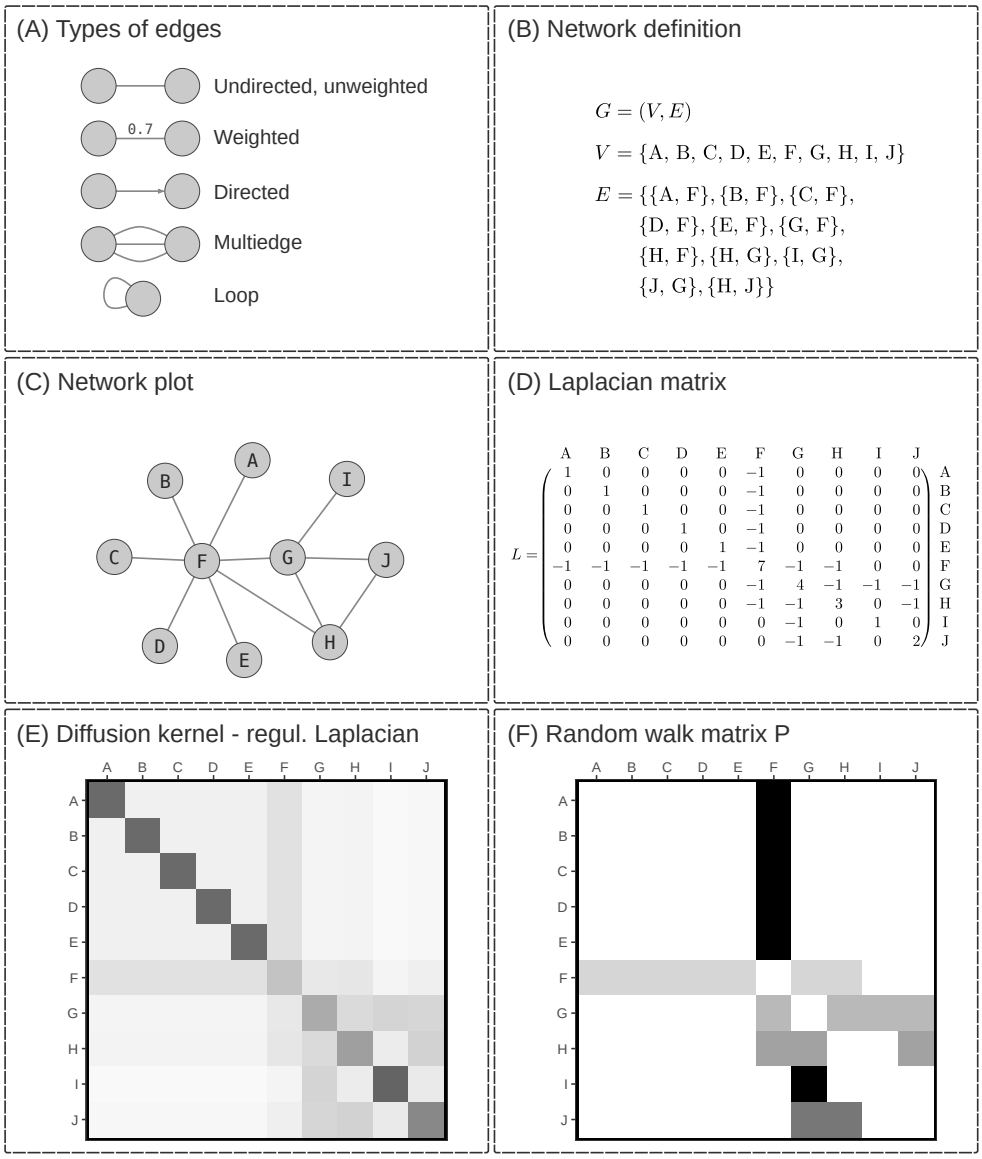


Figure 9: Examples on network definitions. Each sub-figure represents one stage in the network definition and analysis. **(A)** A network can be built using different types of edges. **(B)** Definition of the vertex and edge sets. **(C)** Plot of the network. **(D)** Unnormalised graph Laplacian matrix. The diagonal contains the node degree. Node F would be the equivalent of a biological hub, given its high degree. **(E)** Kernel matrix for label propagation, derived from the Laplacian matrix. Darker colours reflect a higher node similarity. Equation 14 details its formal definition, with $\sigma^2 = 1$ and using L instead of $\tilde{L}$. **(F)** Random walk matrix P from equation 4. The darker the colour, the higher the probability of transitioning from the row-indexed to the column-indexed vertex. Note how P is asymmetric.

The graph Laplacian matrix

The starting point of many network-based propagation methods is the graph Laplacian matrix, which comes in two flavours: the *unnormalised graph Laplacian* L and its *normalised* version $\tilde{L}$. These are defined for simple graphs but naturally work on weighted graphs (Smola and Kondor, 2003):

$$L := D - W \tag{2}$$

$$\tilde{L} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I_n - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \tag{3}$$

where I_n is the $n \times n$ identity matrix, being n the order of the graph. Isolated nodes typically require a special treatment in $\tilde{L}$ because their degree is 0. Figure 9D contains a small example on how to compute L. Because G is undirected, L and $\tilde{L}$ are symmetric and therefore diagonalisable. The spectral properties of L and $\tilde{L}$ have been extensively studied and are tightly connected to the topological properties of G.

Random walks

A notable branch of graph theory is the study of *random walks*, which are markov chains on graphs (Lovász, 1993). In each step of a random walk, a fictional walker sits on a node i and randomly chooses an edge to resume her random walk, or decides to start another walk. The stationary probability distributions of such processes exist and have been extensively studied.

The matrices in equations 4 and 5 are normalised versions of the adjacency matrix, commonly used to compute random walk-based scores on undirected graphs (Cowen et al., 2017). Again, isolated nodes require special treatment. An example of the former can be found in figure 9F.

$$P := WD^{-1} \tag{4}$$

$$\tilde{P} := D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{5}$$

2.2.3 Network propagation in computational biology

The starting point of network-based algorithms has its roots in the Guilt By Association (GBA) principle (Oliver, 2000). In a succint formulation, it states that molecular entities that interact are prone to share biological properties. An exemplary instance can be found in (Lavi et al., 2012): genes that appear co-expressed tend to be closer in a network of interactions.

The straightforward approach for GBA is neighbour voting, where the label of a given node is predicted by letting its neighbours vote with their own labels (Ballouz et al., 2016). Neighbour voting has been improved into the so-called label propagation and network diffusion approaches, which generally allow further propagation into higher-order neighbourhoods (Cowen et al., 2017).

Resorting to propagation beyond neighbour nodes is endorsed by the network parsimony principle, that supports that the underlying perturbations propagate through the shortest paths within the complex molecular networks (Massucci et al., 2016). However, purely shortest paths-based approaches suffer from the "small world" property of biological networks: most nodes can be reached from every other node in a small number of steps due to the presence of hubs, i.e. highly connected nodes (Cowen et al., 2017). The use of label propagation techniques has been extensively

reviewed for finding social communities (Zoidi et al., 2015) and genetic associations (Cowen et al., 2017).

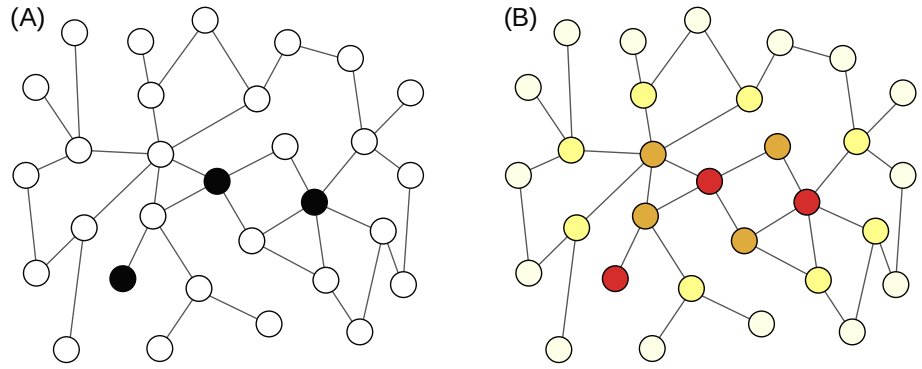


Figure 10: Network diffusion example. **(A)** Three seed nodes are labelled as positive before applying the network diffusion. **(B)** After the propagation, all the nodes earn a diffusion score. Its magnitude is represented by a heat colour scale: red scores are the highest whereas white are the lowest. The closer to the seed nodes, the higher the score.

The following sections provide several angles to tackle with the network diffusion and the random walks paradigms. A recurrent topic is how different approaches and physical models lead to equivalent formulations.

Physical models

An intuitive way to define diffusion-based approaches is through physical models, illustrated in figure 10. Equation 6 contains the equation of a fluid propagation model:

$$\frac{\partial f^s(t)}{\partial t} = -L_\gamma f^s(t) + b^s u(t) \tag{6}$$

$f^s(t)$ is the column vector containing the amount of fluid in every node at time t. $L_\gamma = L + \gamma I$, being L the graph Laplacian from equation 2, I the identity matrix and $\gamma \in \mathbb{R}$ a parameter to control the rate of fluid leaking in every node. b^s is the vector indicating the rate at which the fluid is pumped on the source nodes, and $u(t)$ is the unit step function. The fluid densities f^s in the stationary state ($t \to \infty$) are given by equation 7.

$$f^s = L_\gamma^{-1} b^s \tag{7}$$

HotNet (Vandin et al., 2011) uses the model in equation 6, placing sources on one gene at a time and regarding f^s in equation 7 as the influence of that gene to all the genes in the network. Influence values are used to build an influence graph, in order to find subnetworks with mutations in a statistically significant number of patients. In HotNet, the source node diffuses 1 positive unit and the rest of nodes diffuse 0 units of flow; therefore, b^s is a binary vector.

HotNet2 is a second iteration with the same purpose, based on insulated diffusion processes, which can also be formulated in terms of random walks with restarts (Mark DM Leiserson et al., 2015) as in equation 17.

Likewise, TieDIE (Paull et al., 2013) defines two diffusion processes to find 'linker genes' between two sets of genes: the source set (mutated genes) and the target set (transcription factors). The authors try three scoring schemes: the HotNet formulation (equation 7), PageRank (equation 17) and the pathway impact score from SPIA (Adi Laurentiu Tarca et al., 2008).

eQED is a modified version of an electrical model to accomodate for directionality (Suthram et al., 2008), based on the random walk matrix in equation 4 (Cowen et al., 2017). eQED prioritises causal genes, among those close to a genetic marker, for downstream gene expression changes.

GeneMANIA (Mostafavi et al., 2008) predicts gene functions through diffusion processes on multiple networks that represent complementary data sources. The networks are combined using weights that maximise a kind of kernel-target alignment (Cristianini et al., 2002). The diffusion process is solved as in equation 8, equivalent to equation 7 with $y = b^s$, $f = f^s$, $\gamma = 1$.

$$f = (I + L)^{-1} y \tag{8}$$

The input y is defined differently; nodes are divided into positives (genes with the property of interest), negatives (genes with other properties) and unlabelled (nodes to be prioritised). The positives diffuse 1 positive unit, like HotNet. However, negative nodes diffuse -1 units, whereas unlabelled nodes diffuse a bias term $k = \frac{n^+ - n^-}{n}$ that accounts for the balance between the number of positive n^+ and negative n^- instances over the number of nodes n.

On the other hand, the diffusion problem in equation 6 can be posed as the convex optimisation instance in equation 9 with $y = b^s$ and $f = f^s$ (Tsuda et al., 2005); the parameter c is a tradeoff between loss and smoothness.

$$\min_f (f - y)^T (f - y) + c f^T L f \tag{9}$$

Its solution is again similar to the classical diffusion problem (equation 7):

$$f = (I + cL)^{-1} y \tag{10}$$

The authors in (Tsuda et al., 2005) reformulated it to:

$$\min_{f,\gamma} (f - y)^T (f - y) + c\gamma \qquad f^T L f \leqslant \gamma \tag{11}$$

The convex problem was further generalised to accomodate k networks (equation 12), with an application to protein function prediction. Analogously to GeneMANIA, the negatives are forced to diffuse -1 units in this approach, whereas unlabelled nodes diffuse 0 units.

$$\min_{f,\gamma} (f - y)^T (f - y) + c\gamma \qquad f^T L_k f \leqslant \gamma;\ k = 1, \ldots, m \tag{12}$$

Similar formulations have been used to derive supervised and unsupervised classification algorithms that favour smooth predictors on the network. This idea has found its application in microarray data, in order to leverage a priori network data (Rapaport et al., 2007).

Graph kernels

A vast amount of diffusion instances can be formulated in terms of graph kernels, which define a similarity measure between nodes (Smola and Kondor, 2003). Specifically, one can define a kernel starting from the eigensystem $\{(\lambda_i, v_i)\}$ of the graph Laplacian L (equation 2) or $\tilde{L}$ (equation 3) by applying a regularisation function $r(\lambda)$ to its spectrum:

$$K = \sum_{i=1}^{m} \frac{1}{r(\lambda_i)} v_i v_i^T, \qquad \text{defining } \frac{1}{0} \equiv 0 \tag{13}$$

This formulation leads to commonly used propagation processes – their kernels are described in equations 14, 15 and 16. Note that equations 14 and 15 also apply to L. The parameter σ^2 controls the reach of the diffusion spreading.

$$K = (I + \sigma^2 \tilde{L})^{-1} \qquad \text{Regularised Laplacian} \tag{14}$$

$$K = e^{-\sigma^2/2\tilde{L}} \qquad \text{Diffusion equation} \tag{15}$$

$$K = (aI - \sigma^2 \tilde{L})^p,\ a \geqslant 2 \qquad \text{p-step random walk} \tag{16}$$

Note the equivalence between the regularised Laplacian kernel and several literature approaches (equations 7, 8 and 10).

The advantage of defining network propagation in terms of graph kernels is that it enables its usage on a broad range of kernel-based machine learning algorithms. Kernelised scores have been used to predict disease-gene associations, also allowing the integration of biological networks from various data sources (Valentini, Paccanaro, et al., 2014). Under this formalism, *positive-unlabelled learning* is a branch of machine learning that assumes the negative class is rather unlabelled than negative, becoming a *one-class learning* instance (Valentini, Armano, et al., 2016). This scenario is common among network biology, for example in the prediction of drug targets (Ferrero et al., 2017) and disease genes (Mordelet and Vert, 2011).

Random walks

Alternatively, random walks methods are based on a different formulation but share a considerable part of mathematical background with diffusion and kernel algorithms (Cowen et al., 2017). For instance, the p-step kernel in equation 16 is governed by random walks.

The PageRank web ranking algorithm (Page et al., 1999) is an early use of random walks with restart to model a web surfer. The surfer walks a graph whose nodes represent websites and edges are hyperlinks between them. In each step, she starts from a node and follows a random web link with a probability d (called the damping factor), or restarts her random walk with probability $1 - d$ in a random node. The PageRank scores are defined as the stationary probabilities of this process for each node (website). A website earns a high score if pointed to by many other websites, and if those also have high scores.

Its most common variant is called personalised PageRank due to the custom prior distribution p_0, which controls the frequency of the restarts in

each node. The classical PageRank just sets a uniform prior for p_0. The stationary state p can be computed using P (equation 4), a normalised version of the adjacency matrix, where α controls the damping, i.e. the tradeoff between the prior distribution and the network data (Cowen et al., 2017):

$$p = \alpha(I - (1 - \alpha)P)^{-1}p_0 \tag{17}$$

A key difference exists between personalised PageRank and other kernelised or fluid/heat propagation approaches. PageRank only normalises outgoing flow, whereas diffusion normalises both incoming and outgoing flow (Erten, Bebek, et al., 2011). As a consequence, PageRank is asymmetric and cannot be regarded as a kernel matrix (Cowen et al., 2017).

PageRank has since fostered numerous uses: finding relevant pathways while accounting for a protein-protein interaction network (Glaab et al., 2012), prioritising candidate disease genes (Erten, Bebek, et al., 2011; I. Lee et al., 2011), discovering protein targets using metabolic networks (Bánky et al., 2013), improving cancer classification by identifying risk-active pathways (W. Liu et al., 2013) and creating low-dimensional representations of multiple networks (H. Cho et al., 2016). TieDIE (Paull et al., 2013) and HotNet2 (Mark DM Leiserson et al., 2015), already mentioned in section 2.2.3, also use scoring functions based on random walks.

2.2.4 Statistical properties of network propagation

The question 'how high a diffusion score must be to be considered high?' logically arises from the biostatistics standpoint. Some authors have included various flavours of statistical adjustments into the diffusion scores, as an attempt to equalise nodes with systematically low or high scores. Figure 11 illustrates the concept of statistical adjustment or normalisation, typically involving the definition of a null or background distribution of scores.

Normalised scores

An early study pointed out the effect of node degree in diffusion processes (Erten and Koyutürk, 2010) and was later publised as a software called DADA (Erten, Bebek, et al., 2011). The authors claim that, although diffusion approaches gain power from considering indirect connections and path multiplicity, such methods systematically favour highly connected proteins. Association scores from random walks with restarts are proven to favour high-degree nodes – diffusion scores too, but to a lesser extent.

Several sampling schemes are suggested, accounting for the degree distribution of the input genes, to provide uniform (normalised) prioritisation schemes. The unnormalised prioritisation uses personalised PageRank (equation 17) to compute $\alpha(v, D)$, the association score between the node v and the disease D, determined by a set of seed nodes S. The normalised association $\alpha_{SD}(v, D)$ in equation 18 adjusts $\alpha(v, D)$ by the means of a z-score, whose mean μ_S and standard deviation σ_S come from the null distribution of $\alpha(v, D)$, i.e. when computed from random inputs. μ_S and σ_S are estimated with $1,000$ random samples matching the size and degree distribution of S.

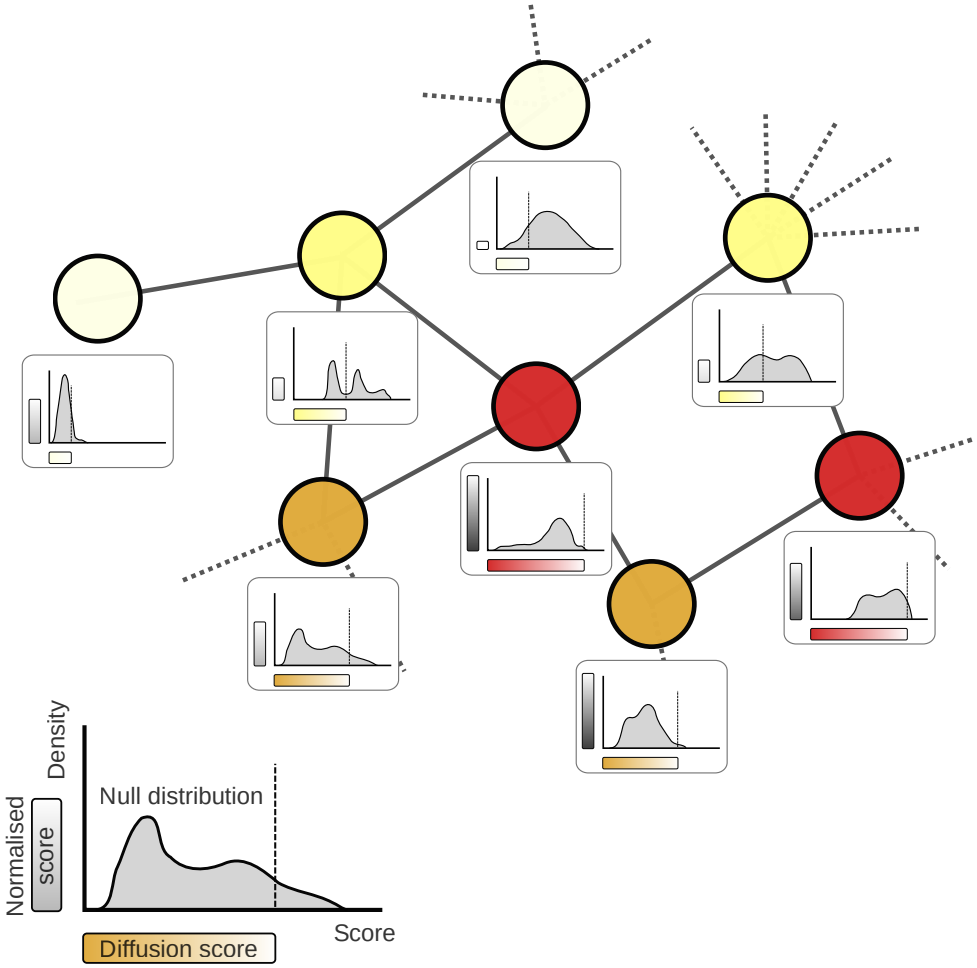


Figure 11: Statistical normalisation of diffusion scores. This small example illustrates the core idea of statistical normalisation. The heat colours represent the magnitude of diffusion scores after the network propagation, using any of the kernel, physical or random walk-based models. Normalised scores shuffle or resample the input to the diffusion process to obtain a null distribution and quantify how extreme each observed score is. The drive behind the normalisation is the presence of unwanted topological behaviours that favour certain nodes, in turn biasing the results and overshadowing novel findings.

$$\alpha_{SD}(v, D) = \frac{\alpha(v, D) - \mu_S}{\sigma_S} \tag{18}$$

The authors show that the uniform prioritisation is beneficial for the notable amount of low-degree disease nodes, albeit disease genes tend to have a higher degree than non-disease genes. Uniform ranking schemes better discover loosely connected disease genes missed by unnormalised scores, at the expenses of more false negatives within highly connected genes. Normalised and unnormalised prioritisations are finally combined in the so-called hybrid ranking strategies, in an attempt to keep the best of each.

In the study by (Cun and Fröhlich, 2013), the authors obtain t-statistics from gene expression data and smooth their absolute value with network diffusion. Equation 19 shows how to obtain the smoothed t-statistic $\tilde{t}$ from

t, the column vector with the absolute value of the original t-statistics, and a p-step random walk kernel K (equation 16).

$$\tilde{t} = t^T K \tag{19}$$

They further seek significant biomarkers within the top 10% genes, prioritised using $\tilde{t}$, by permuting t 1,000 times and computing empirical p-values for the top genes. Finally, the genes that remain significant after multiple test correction are used as features to train a support vector machine that predicts disease. This approach has been distributed within the netClass software package (Cun and Fröhlich, 2014).

A posterior study (Bersanelli et al., 2016) highlights differentially enriched modules from the analysis of high-throughput data. Special emphasis is put on exploring the impact of permutations and resampling. Input statistics x_0, analogously to t from equation 19, are smoothed into x^* using the regularised normalised Laplacian kernel (equation 14):

$$x^* = \gamma(\tilde{L} + \gamma I)^{-1} x_0 = \left(\frac{1}{\gamma}\tilde{L} + I\right)^{-1} x_0, \qquad \gamma > 0 \tag{20}$$

They introduce the so-called *network smoothing index* in equation 21, a quantitative measure of the relative change in the j-th gene before (x_{0j}) and after (x^*_j) the diffusion. The parameter ϵ balances the relative importance of initial and final diffusion states.

$$S_j(x_0) = \frac{x^*_j}{x_{0j} + \epsilon} \tag{21}$$

To compare experimental groups control u_1 and case u_2, the smoothing indices are subtracted:

$$\Delta S_j = S_j(u_2) - S_j(u_1) \tag{22}$$

This formulation pursues the mitigation of topological-only effects, like systematically low or high S_j, expected to cancel out. Likewise, if the inferential statistics (u) are already a contrast between experimental groups, the authors define the Sp value as a combination between the original network smoothing index S_j and its empirical p-value p_j (equation 23). p_j is obtained by comparing $S_j(u)$ to its null distribution by drawing random permutations of u and computing S_j on the null trials.

$$Sp_j(u) = -\log_{10}(p_j) S_j(u) \tag{23}$$

Both ΔS_j and $Sp_j(u)$ are explicit ways to addres the so-called hub effect, and later used as a proxy to identify differentially enriched modules.

Another possibility to normalise the scores is by randomising the network connections instead of the seed genes in the null model (Biran et al., 2019). Rewired networks preserve the degree distribution of the original network but randomly swap the endpoints of edge pairs. The personalised PageRank (equation 17) was used to score the network nodes for several gene prioritisation tasks. Empirical p-values were obtained for each node by comparing the actual score to that of rewired networks. A distinctive property of the

network rewiring null model is that it inherently controls for the degree distribution of the input nodes.

Scoring sets of nodes

Nework propagation has also been proven useful within the pathway analysis scope. Pathways, understood as *sets of nodes* (see section 2.1.4) from the network, are summarised into a single number that reflects their association to the input data. The inherently statistical nature of pathway analysis (see section 2.3.1 on pathway analysis for metabolomics) makes network propagation and statistical measures concur.

The EnrichNet algorithm (Glaab et al., 2012), mentioned in section 2.2.3, defines pathway-level statistics derived from random walks and adjusts for a reference histogram. The latter can be regarded as an attempt to account for topology-related biases. First, the personalised PageRank is computed using the input genes as a prior, and converted to a dissimilarity measure by subtracting it from 1. The scores of genes within every pathway, understood as a gene set, are binned into a histogram. A reference histogram is computed by the averaging of all the pathways. The Xd-distance of a pathway (equation 24) is defined as a weighted difference between the pathway and the reference histograms, and used to prioritise pathways.

$$Xd = \sum_{i=1}^{n} \frac{P_{ic} - P_{ia}}{in} \tag{24}$$

Specifically, P_{ic} is the percentage of dissimilarity scores for the target gene set and the pathway c within bin i, in relation to the total amount of genes in pathway c. P_{ia} is analogous defined, but using the background model. n is the number of bins, whereas i is the current bin – note how the first bins are upweighted, i.e. the ones with genes the most similar to the target gene set in terms of random walks. Xd-scores should display high (and positive) values for relevant pathways, because their genes are expected to be similar to those in the target gene set.

The authors from (L. Liu and Ruan, 2013) suggest a parametric approach to identify enriched biological pathways starting from a gene list. Analogously to EnrichNet, the random walk with restart is solved. For each pathway, regarded as a gene set, the observed mean similarity score D is computed and normalised as a z-score Z. The expected value μ_R and standard deviation σ_R of each pathway are estimated from $1,000$ input permutations. Pathways are finally prioritised by the z-scores.

$$Z = \frac{D - \mu_R}{\sigma_R} \tag{25}$$

An analogous z-score approach was applied to gene set proximity (Aguirre-Plans et al., 2018), based on shortest paths between two gene sets S and T, defining the proximity from S to T as:

$$d(S,T) = \frac{1}{|S|} \sum_{u \in S} \min_{v \in T} d(u,v)\,, \tag{26}$$

where $d(u, v)$ is the shortest path length between nodes u and v. The mean value and standard deviation in the z-score were estimated from random resamplings of S and T matching the original degree distributions and set sizes.

2.3 APPLICATIONS OF NETWORK PROPAGATION

This section contains specific problems in computational biology that have been addressed from a network analysis perspective.

2.3.1 Metabolomics data enrichment

The understanding and interpretation of experimental data is considered an essential challenge to generate biological knowledge from metabolomics data (Chagoyen and Pazos, 2012). This is the purpose of the so-called pathway analysis and enrichment techniques, conceived to contextualise experimental findings within known biological annotations. Following the review in (Khatri et al., 2012), pathway enrichment techniques can be classified in over representation analysis, set enrichment analysis and topology-based approaches.

Over representation analysis (ORA) is generally based on a statistical test to identify pathways with a high occurrence in a list of metabolites. Early methods for tackling the same problem in gene expression, such as GOstat (Beißbarth and Speed, 2004), made use of Fisher's exact test or a χ^2 test to obtain a measure of statistical significance (Everitt, 1992). Such simple approaches are still used up to date, available in tools like the web servers MetaboAnalyst (Chong et al., 2018) or IMPaLA (Kamburov et al., 2011). Limitations of ORA include its low discriminative power among certain sets and the sensitivity with respect to the threshold that generates the metabolite list (Glaab et al., 2012), although consistence among ORA methods has been reported in a recent comparison (Marco-Ramell et al., 2018). Other caveats include the generally assumed independence between pathways, and equivalence and independence between genes, in the analogous case of gene expression data (Khatri et al., 2012).

Set Enrichment Analysis (SEA) generalises ORA by dropping the requirement of a threshold to obtain a list of metabolites. Metabolite Set Enrichment Analysis (MSEA), adapted from a similar method for gene expression (Subramanian, Tamayo, et al., 2005), introduced the SEA paradigm in metabolomics. Metabolites can be sorted using a statistic (e.g. fold change) and metabolite sets are tested using Kolmogorov-Smirnov-like statistic. MSEA can be found in MetaboAnalyst. An altenative approach named PAPi (Aggio et al., 2010) defines pathway activity scores using relative metabolite abundance and the number of known and measured metabolites within each pathway. Despite SEA drops the cutoff parameter, it is unclear whether it outperforms ORA in real settings (Mitrea et al., 2013).

A complementary approach is provided by Topology-based analysis (TP). Metabolic pathways are sought by leveraging network data, accounting for

the distinct roles and properties of the metabolites within biological processes. MetPA (Xia and Wishart, 2010), part of MetaboAnalyst, computes centrality measures on a metabolite level and reports a topological impact of the metabolites in the user-provided list. mummichog predicts functional metabolic activity through module analysis within metabolic networks followed by pathway analysis (S. Li et al., 2013). Other tools revolve around curating and visualising network data. For instance, Metscape 2 (Karnovsky et al., 2011) buils and displays the so-called CREG networks, connecting compounds, reactions, enzymes and genes. TP methods are bounded by the network data limitations, such as biases and incompleteness (Bayerlová et al., 2015). Current challenges include accounting for pathway cross-talk (Donato et al., 2013), the consideration of organism-specific data and the interpretability of the results (Booth et al., 2013).

In a real scenario, however, TP analysis does not necessarily outpeform simpler tests in gene expression data (Bayerlová et al., 2015). The differential traits of certain enrichment methods are not an automatic guarantee of their optimality, which should be assessed by their ability to recover truly affected pathways (Mitrea et al., 2013). This is further hindered by the lack of standard datasets for the evaluation of such methods (Mitrea et al., 2013) and the statistical challenges of metabolomics, such as the unknown size of the metabolome and the sparsity of annotations compared to other omics sciences (Chagoyen and Pazos, 2012). In addition, current evidence supports the inexistence of a universally optimal pathway enrichment technique (Adi L Tarca et al., 2013), adding an extra layer of complexity when defining the direction of future efforts.

2.3.2 Disease gene identification

The identification of novel therapeutic targets is an area of active research. A wide spectrum of network-based approaches have been developed for this purpose or similar problems. Efforts include neighbour voting (Ballouz et al., 2016), semi-supervised learning (Valentini, Armano, et al., 2016), propagation and random walks (Vanunu et al., 2010), artificial neural networks (Muslu et al., 2019), supervised learning on diffusion-based features (H. Cho et al., 2016) or on diffusion-based distances (M. Cao et al., 2013).

Methods make use of a variety of networks, ranging from a single interactome (Vanunu et al., 2010) to supervised weighted combinations of networks from various data sources (Mostafavi et al., 2008; Tsuda et al., 2005; Valentini, Paccanaro, et al., 2014). The assessment of the real benefit of integrating multiple sources is endorsed by some authors (Valentini, Paccanaro, et al., 2014), whereas others have found only a marginal improvement, if any, over a plain averaging of the networks (Mordelet and Vert, 2011; Tsuda et al., 2005).

The impact of the network coverage has also been examined. In line with the robustness to noise in diffusion-based methods, which are able to downweigh spurious predictions (Cowen et al., 2017), it has been reported that the usage of a larger network outweighs the higher proportion of noisy and low-confidence edges (J. K. Huang et al., 2018).

Drug target data suffers from the true negative issue: reliable data about truly unsuccessful targets is extremely rare, so the negative class becomes fuzzier (Ferrero et al., 2017). Another limitation derives from the fact that targeting is usually known at the protein complex level (or even for a protein family) instead of the protein sub-unit level (Bento et al., 2014). The label is therefore shared by all the genes that code the proteins within that protein complex, implying that the data is structured. Usual cross validation techniques yield misleading performance estimates on structured data if uncorrected (D. R. Roberts et al., 2017). Similar problems with cross validation on structured data have been identified and corrected in other fields, namely ligand-target binding models (Lopez-del Rio et al., 2018) and ecology (D. R. Roberts et al., 2017).

2.4 OPEN ISSUES

Several limitations and challenges in the application of network-based and pathway analysis approaches in computational biology were mentioned throughout this chapter. The following sections highlight specific issues of special interest within the scope of this thesis.

2.4.1 Heterogeneity and biases in network propagation

One of the first steps when applying network propagation to a new problem is the decision of how to propagate the data on the network. The choice of the graph kernel, the treatment of positive, negative and unlabelled nodes and the need of a statistical normalisation are open questions. An implementation that allows a systematic benchmark of these options is still missing.

Besides, both the networks –for instance, protein-protein interaction networks (Edwards et al., 2002)– and the data that is propagated on them suffer from incompleteness and spurious associations. Despite their robutsness, diffusion-based approaches are still affected to a certain extent that has not been thoroughly characterised.

On the other hand, network topology has been proven to affect diffusion scores (Erten, Bebek, et al., 2011). A plethora of network data resources is publicly available, with differences in data sources, coverage, topology and confidence (J. K. Huang et al., 2018). The network choice greatly affects downstream analysis (J. K. Huang et al., 2018), including diffusion-based approaches, which were used in their original study.

In addition, the coexistence of well-studied and barely known genes or proteins, respectively turning into high and low-degree nodes, poses a challenge. In (Erten, Bebek, et al., 2011), propagation methods are shown to better predict highly connected proteins, but known disease genes are also biased towards highly connected genes. This circularity hampers the discovery of novel disease genes among the less studied ones.

Few publications have explored how sensitive diffusion scores are. The authors in (Bersanelli et al., 2016) quantify an empiric p-value by permuting the input labels, in order to account for nodes with systematically low

or high scores. A similar concept has been applied to gene expression t-statistics (Cun and Fröhlich, 2013) and to disease gene prioritisation (Erten, Bebek, et al., 2011), whereas (Biran et al., 2019) rewire network connections instead of permuting. There is enough evidence supporting the existence of topology-related biases and a noticeable impact upon their removal. The node degree seems to have a clear biasing role, but is unable to explain the whole casuistic of score behaviour (Hill et al., 2019). This further encourages its characterisation and quantification, including factors like the non-observability of certain nodes (i.e. due to experimental limitations).

2.4.2 Results interpretability in pathway analysis

Even though pathway analysis was conceived to improve the biological interpretation of experimental data, understanding a list of affected pathways still remains as an outstanding challenge due to pathway overlap and cross-talk effects (Donato et al., 2013). Active research lines include the addition of richer organism-specific contextual representations (Booth et al., 2013), the modelling of pathway cross-talk (Donato et al., 2013) and the creation of aggregated pathway databases that better reflect current knowledge (Domingo-Fernandez et al., 2019).

2.4.3 Performance overestimation in target gene prediction

Drug target data is often known for protein complexes instead of their individual protein sub-units (Bento et al., 2014). The presence of data structure can artificially inflate the peformance estimates from classical cross-validation (Lopez-del Rio et al., 2018; D. R. Roberts et al., 2017). In addition, the performance metrics require a careful consideration. Classical metrics like the Area Under the Receiver Operating Characteristic can be misleading in early-retrieval (Saito and Rehmsmeier, 2015), like the practical scenario in which only few targets can be tested. A comprehensive study controlling both factors is needed to obtain a realistic snapshot of the expected benefit, if any, of applying network propagation methods for drug discovery.

2.4.4 Free and open source software

Competitive algorithms can be found in the literature for most of the areas in computational biology. However, the availability of their software, source code and the interaction with the user is variable across their spectrum.

It is essential to provide the source code and the data that generates the conclusions of any manuscript to achieve reproducible science (Peng, 2011). The lack of the data or source code that support the findings hinders their replication and a wider method adoption.

Certain algorithms are available through a web server, preventing the user from customising its settings, modifying the algoritm or its application to other salient problems in computational biology. For instance, (Kamburov et al., 2011) offers a user-friendly web server for pathway enrichment that, on the other hand, does not contemplate changing the pathway libraries. En-

richNet (Glaab et al., 2012) is available on a web server that also offers an API using RESTful calls, enabling the user some programmatic options and network customisation. The web server MetaboAnalyst (Chong et al., 2018) has deployed a companion R package to provide batch analysis, reproducibility and transparency.

Another common practice is to provide the raw data and the scripts that were used in the publication, like in MashUp (H. Cho et al., 2016). This policy is commendable, albeit still limited by the lack of maintenance over time and the non-standard distribution of the software, sometimes depending on a private or institutional server.

Public repositories, either general purpose like CRAN[11] (R Core Team, 2018) or specialised like Bioconductor[12] (Huber et al., 2015), are a robust solution to endorse good coding practices, sofware maintenance and support, reproducibility, data availability and standard distribution channels for the R computing language. This is the case of RANKS (Valentini, Armano, et al., 2016), available in CRAN, and EGAD (Ballouz et al., 2016), published in Bioconductor. There is an analogous initiative for the python programming language community, called Biopython (Cock et al., 2009).

An effort is needed not only in software publication in public repositories, but also in proper maintenance and long term support. The netClass R package (Cun and Fröhlich, 2014) serves as an example: it was published in CRAN in 2013, but archived by July 29th, 2017 due to uncorrected check problems. This becomes an obstacle for its adoption and for reproducible science, since the package might need bug fixes to work on latest R versions.

11 https://cran.r-project.org. Accessed on 31/12/2019.
12 https://www.bioconductor.org. Accessed on 31/12/2019.

REFERENCES

1000 Genomes Project Consortium and others
2015 "A global reference for human genetic variation", *Nature*, 526, 7571, p. 68.

Aggio, Raphael BM, Katya Ruggiero, and Silas Granato Villas-Bôas
2010 "Pathway Activity Profiling (PAPi): from the metabolite profile to the metabolic pathway activity", *Bioinformatics*, 26, 23, pp. 2969-2976.

Aguirre-Plans, Joaquim, Janet Piñero, Jörg Menche, Ferran Sanz, Laura Furlong, Harald Schmidt, Baldo Oliva, and Emre Guney
2018 "Proximal pathway enrichment analysis for targeting comorbid diseases via network endopharmacology", *Pharmaceuticals*, 11, 3, p. 61.

Aimo, Lucila, Robin Liechti, Nevila Hyka-Nouspikel, Anne Niknejad, Anne Gleizes, Lou Götz, Dmitry Kuznetsov, Fabrice PA David, F Gisou van der Goot, Howard Riezman, et al.
2015 "The SwissLipids knowledgebase for lipid biology", *Bioinformatics*, 31, 17, pp. 2860-2866.

Alanis-Lobato, Gregorio, Miguel A Andrade-Navarro, and Martin H Schaefer
2016 "HIPPIE v2.0: enhancing meaningfulness and reliability of protein–protein interaction networks", *Nucleic acids research*, gkw985.

Amberger, Joanna S, Carol A Bocchini, Alan F Scott, and Ada Hamosh
2018 "OMIM.org: leveraging knowledge across phenotype–gene relationships", *Nucleic acids research*, 47, D1, pp. D1038-D1043.

Bader, Gary D, Michael P Cary, and Chris Sander
2006 "Pathguide: a pathway resource list", *Nucleic acids research*, 34, suppl 1, pp. D504-D506.

Ballouz, Sara, Melanie Weber, Paul Pavlidis, and Jesse Gillis
2016 "EGAD: ultra-fast functional analysis of gene networks", *Bioinformatics*, 33, 4, pp. 612-614.

Bang-Jensen, Jørgen and Gregory Z Gutin
2008 *Digraphs: theory, algorithms and applications*, Springer Science & Business Media.

Bánky, Dániel, Gábor Iván, and Vince Grolmusz
2013 "Equal opportunity for low-degree network nodes: a PageRank-based method for protein target identification in metabolic graphs", *PLoS One*, 8, 1, e54204.

Bayerlová, Michaela, Klaus Jung, Frank Kramer, Florian Klemm, Annalen Bleckmann, and Tim Beißbarth
2015 "Comparative study on gene set and pathway topology-based enrichment methods", *BMC bioinformatics*, 16, 1, p. 334.

Beißbarth, Tim and Terence P Speed

2004 "GOstat: find statistically overrepresented Gene Ontologies within a group of genes", *Bioinformatics*, 20, 9, pp. 1464-1465.

Bento, A Patrícia, Anna Gaulton, Anne Hersey, Louisa J Bellis, Jon Chambers, Mark Davies, Felix A Krüger, Yvonne Light, Lora Mak, Shaun McGlinchey, et al.

2014 "The ChEMBL bioactivity database: an update", *Nucleic acids research*, 42, D1, pp. D1083-D1090.

Bersanelli, Matteo, Ettore Mosca, Daniel Remondini, Gastone Castellani, and Luciano Milanesi

2016 "Network diffusion-based analysis of high-throughput data for the detection of differentially enriched modules", *Scientific reports*, 6, p. 34841.

Biran, Hadas, Martin Kupiec, and Roded Sharan

2019 "Comparative analysis of normalization methods for network propagation", *Frontiers in genetics*, 10, p. 4.

Booth, Sean C, Aalim M Weljie, and Raymond J Turner

2013 "Computational tools for the secondary analysis of metabolomics experiments", *Computational and structural biotechnology journal*, 4, 5, e201301003.

Cao, Mengfei, Hao Zhang, Jisoo Park, Noah M Daniels, Mark E Crovella, Lenore J Cowen, and Benjamin Hescott

2013 "Going the distance for protein function prediction: a new distance metric for protein interaction networks", *PloS one*, 8, 10, e76339.

Carter, Hannah, Matan Hofree, and Trey Ideker

2013 "Genotype to phenotype via network analysis", *Current opinion in genetics & development*, 23, 6, pp. 611-621.

Caspi, Ron, Richard Billington, Ingrid M Keseler, Anamika Kothari, Markus Krummenacker, Peter E Midford, Wai Kit Ong, Suzanne Paley, Pallavi Subhraveti, and Peter D Karp

2019 "The MetaCyc database of metabolic pathways and enzymes-a 2019 update", *Nucleic acids research*.

Chagoyen, Monica and Florencio Pazos

2012 "Tools for the functional interpretation of metabolomic experiments", *Briefings in bioinformatics*, 14, 6, pp. 737-744.

Chatr-Aryamontri, Andrew, Rose Oughtred, Lorrie Boucher, Jennifer Rust, Christie Chang, Nadine K Kolas, Lara O'Donnell, Sara Oster, Chandra Theesfeld, Adnane Sellam, et al.

2017 "The BioGRID interaction database: 2017 update", *Nucleic acids research*, 45, D1, pp. D369-D379.

Cho, Hyunghoon, Bonnie Berger, and Jian Peng

2016 "Compact integration of multi-network topology for functional analysis of genes", *Cell systems*, 3, 6, pp. 540-548.

Chong, Jasmine, Othman Soufan, Carin Li, Iurie Caraus, Shuzhao Li, Guillaume Bourque, David S Wishart, and Jianguo Xia

2018 "MetaboAnalyst 4.0: towards more transparent and integrative metabolomics analysis", *Nucleic acids research*.

Clough, Emily and Tanya Barrett

2016 "The gene expression omnibus database", in *Statistical Genomics*, Springer, pp. 93-110.

Cock, Peter JA, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al.

2009 "Biopython: freely available Python tools for computational molecular biology and bioinformatics", *Bioinformatics*, 25, 11, pp. 1422-1423.

Consortium, Gene Ontology

2016 "Expansion of the Gene Ontology knowledgebase and resources", *Nucleic acids research*, 45, D1, pp. D331-D338.

Consortium, UniProt

2018 "UniProt: a worldwide hub of protein knowledge", *Nucleic acids research*, 47, D1, pp. D506-D515.

Cowen, Lenore, Trey Ideker, Benjamin J Raphael, and Roded Sharan

2017 "Network propagation: a universal amplifier of genetic associations", *Nature Reviews Genetics*, 18, 9, p. 551.

Cristianini, Nello, John Shawe-Taylor, Andre Elisseeff, and Jaz S Kandola

2002 "On kernel-target alignment", in *Advances in neural information processing systems*, pp. 367-373.

Cun, Yupeng and Holger Fröhlich

2013 "Network and data integration for biomarker signature discovery via network smoothed t-statistics", *PloS one*, 8, 9, e73074.

2014 "Netclass: an r-package for network based, integrative biomarker signature discovery", *Bioinformatics*, 30, 9, pp. 1325-1326.

Diestel, Reinhard

2000 *Graph Theory*, Second edition, Graduate Texts in Mathematics, Springer, vol. 173.

Domingo-Fernández, Daniel, Charles Tapley Hoyt, Carlos Bobis-Álvarez, Josep Marín-Llaó, and Martin Hofmann-Apitius

2018 "ComPath: an ecosystem for exploring, analyzing, and curating mappings across pathway databases", *NPJ systems biology and applications*, 5, 1, p. 3.

Domingo-Fernandez, Daniel, Sarah Mubeen, Josep Marin-Llao, Charles Hoyt, and Martin Hofmann-Apitius

2019 "PathMe: Merging and exploring mechanistic pathway knowledge", *bioRxiv*, p. 451625.

Donato, Michele, Zhonghui Xu, Alin Tomoiaga, James G Granneman, Robert G MacKenzie, Riyue Bao, Nandor Gabor Than, Peter H Westfall, Roberto Romero, and Sorin Draghici

2013 "Analysis and correction of crosstalk effects in pathway analysis", *Genome research*.

Edwards, Aled M, Bart Kus, Ronald Jansen, Dov Greenbaum, Jack Greenblatt, and Mark Gerstein

2002 "Bridging structural biology and genomics: assessing protein interaction data with known complexes", *TRENDS in Genetics*, 18, 10, pp. 529-536.

ENCODE Project Consortium and others

2012 "An integrated encyclopedia of DNA elements in the human genome", *Nature*, 489, 7414, p. 57.

Erten, Sinan, Gurkan Bebek, Rob M Ewing, and Mehmet Koyutürk

2011 "DADA: degree-aware algorithms for network-based disease gene prioritization", *BioData mining*, 4, 1, p. 19.

Erten, Sinan and Mehmet Koyutürk

2010 "Role of centrality in network-based prioritization of disease genes", in *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, Springer, pp. 13-25.

Everitt, Brian S

1992 *The analysis of contingency tables*, Chapman and Hall/CRC, chap. 2x2 Contingency tables.

Fabregat, Antonio, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, et al.

2017 "The reactome pathway knowledgebase", *Nucleic acids research*, 46, D1, pp. D649-D655.

Ferrero, Enrico, Ian Dunham, and Philippe Sanseau

2017 "In silico prediction of novel therapeutic targets using gene–disease association data", *Journal of translational medicine*, 15, 1, p. 182.

El-Gebali, Sara, Jaina Mistry, Alex Bateman, Sean R Eddy, Aurélien Luciani, Simon C Potter, Matloob Qureshi, Lorna J Richardson, Gustavo A Salazar, Alfredo Smart, et al.

2018 "The Pfam protein families database in 2019", *Nucleic acids research*, 47, D1, pp. D427-D432.

Glaab, Enrico, Anaïs Baudot, Natalio Krasnogor, Reinhard Schneider, and Alfonso Valencia

2012 "EnrichNet: network-based gene set enrichment analysis", *Bioinformatics*, 28, 18, pp. i451-i457.

Greene, Casey S, Arjun Krishnan, Aaron K Wong, Emanuela Ricciotti, Rene A Zelaya, Daniel S Himmelstein, Ran Zhang, Boris M Hartmann, Elena Zaslavsky, Stuart C Sealfon, et al.

2015 "Understanding multicellular function and disease with human tissue-specific networks", *Nature genetics*, 47, 6, p. 569.

Han, Heonjong, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al.

2017 "TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions", *Nucleic acids research*, 46, D1, pp. D380-D386.

Hartler, Jürgen

2015 "LIPID MAPS: Tools and Databases", in *Encyclopedia of Lipidomics*, ed. by Markus R. Wenk, Springer Netherlands, Dordrecht, pp. 1-4, ISBN: 978-94-007-7864-1.

Herwig, Ralf, Christopher Hardt, Matthias Lienhard, and Atanas Kamburov

2016 "Analyzing and interpreting genome data at the network level with ConsensusPathDB", *Nature protocols*, 11, 10, p. 1889.

Hill, Abby, Scott Gleim, Florian Kiefer, Frederic Sigoillot, Joseph Loureiro, Jeremy Jenkins, and Melody K Morris

2019 "Benchmarking network algorithms for contextualizing genes of interest", *PLOS Computational Biology*, 15, 12, e1007403.

Hoadley, Katherine A, Christina Yau, Denise M Wolf, Andrew D Cherniack, David Tamborero, Sam Ng, Max DM Leiserson, Beifang Niu, Michael D McLellan, Vladislav Uzunangelov, et al.

2014 "Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin", *Cell*, 158, 4, pp. 929-944.

Huang, Justin K, Daniel E Carlin, Michael Ku Yu, Wei Zhang, Jason F Kreisberg, Pablo Tamayo, and Trey Ideker

2018 "Systematic Evaluation of Molecular Networks for Discovery of Disease Genes", *Cell systems*, 6, 4, pp. 484-495.

Huang, Zhou, Jiangcheng Shi, Yuanxu Gao, Chunmei Cui, Shan Zhang, Jianwei Li, Yuan Zhou, and Qinghua Cui

2018 "HMDD v3.0: a database for experimentally supported human microRNA–disease associations", *Nucleic acids research*, 47, D1, pp. D1013-D1017.

Huber, Wolfgang, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, et al.

2015 "Orchestrating high-throughput genomic analysis with Bioconductor", *Nature methods*, 12, 2, p. 115.

Hwang, Sohyun, Chan Yeong Kim, Sunmo Yang, Eiru Kim, Traver Hart, Edward M Marcotte, and Insuk Lee

2018 "HumanNet v2: human gene networks for disease research", *Nucleic acids research*, 47, D1, pp. D573-D580.

Jeske, Lisa, Sandra Placzek, Ida Schomburg, Antje Chang, and Dietmar Schomburg

2018 "BRENDA in 2019: a European ELIXIR core data resource", *Nucleic acids research*, 47, D1, pp. D542-D549.

Jewison, Timothy, Yilu Su, Fatemeh Miri Disfany, Yongjie Liang, Craig Knox, Adam Maciejewski, Jenna Poelzer, Jessica Huynh, You Zhou, David Arndt, et al.

2013 "SMPDB 2.0: big improvements to the Small Molecule Pathway Database", *Nucleic acids research*, 42, D1, pp. D478-D484.

Kale, Namrata S, Kenneth Haug, Pablo Conesa, Kalaivani Jayseelan, Pablo Moreno, Philippe Rocca-Serra, Venkata Chandrasekhar Nainala, Rachel A Spicer, Mark Williams, Xuefei Li, et al.

2016 "MetaboLights: An Open-Access Database Repository for Metabolomics Data", *Current protocols in bioinformatics*, 53, 1, pp. 14-13.

Kamburov, Atanas, Rachel Cavill, Timothy MD Ebbels, Ralf Herwig, and Hector C Keun

2011 "Integrated pathway-level analysis of transcriptomics and metabolomics data with IMPaLA", *Bioinformatics*, 27, 20, pp. 2917-2918.

Kanehisa, Minoru, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima

2016 "KEGG: new perspectives on genomes, pathways, diseases and drugs", *Nucleic acids research*, 45, D1, pp. D353-D361.

Karnovsky, Alla, Terry Weymouth, Tim Hull, V Glenn Tarcea, Giovanni Scardoni, Carlo Laudanna, Maureen A Sartor, Kathleen A Stringer, HV Jagadish, Charles Burant, et al.

2011 "Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data", *Bioinformatics*, 28, 3, pp. 373-380.

Khatri, Purvesh, Marina Sirota, and Atul J Butte

2012 "Ten years of pathway analysis: current approaches and outstanding challenges", *PLoS computational biology*, 8, 2, e1002375.

Köhler, Sebastian, Leigh Carmody, Nicole Vasilevsky, Julius O B Jacobsen, Daniel Danis, Jean-Philippe Gourdine, Michael Gargano, Nomi L Harris, Nicolas Matentzoglu, Julie A McMurry, et al.

2018 "Expansion of the Human Phenotype Ontology (HPO) knowledge base and resources", *Nucleic acids research*, 47, D1, pp. D1018-D1027.

Koscielny, Gautier, Peter An, Denise Carvalho-Silva, Jennifer A Cham, Luca Fumis, Rippa Gasparyan, Samiul Hasan, Nikiforos Karamanis, Michael Maguire, Eliseo Papa, et al.

2016 "Open Targets: a platform for therapeutic target identification and validation", *Nucleic acids research*, 45, D1, pp. D985-D994.

Kozomara, Ana, Maria Birgaoanu, and Sam Griffiths-Jones

2018 "miRBase: from microRNA sequences to function", *Nucleic acids research*, 47, D1, pp. D155-D162.

Lavi, Ofer, Gideon Dror, and Ron Shamir

2012 "Network-induced classification kernels for gene expression profile analysis", *Journal of Computational Biology*, 19, 6, pp. 694-709.

Lee, Insuk, U Martin Blom, Peggy I Wang, Jung Eun Shim, and Edward M Marcotte

2011 "Prioritizing candidate disease genes by network-based boosting of genome-wide association data", *Genome research*, 21, 7, pp. 1109-1121.

Leiserson, Mark DM, Fabio Vandin, Hsin-Ta Wu, Jason R Dobson, Jonathan V Eldridge, Jacob L Thomas, Alexandra Papoutsaki, Younhun Kim, Beifang Niu, Michael McLellan, et al.

2015 "Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes", *Nature genetics*, 47, 2, p. 106.

Li, Shuzhao, Youngja Park, Sai Duraisingham, Frederick H Strobel, Nooruddin Khan, Quinlyn A Soltow, Dean P Jones, and Bali Pulendran

2013 "Predicting network activity from high throughput metabolomics", *PLoS computational biology*, 9, 7, e1003123.

Liu, Lu and Jianhua Ruan

2013 "Network-based pathway enrichment analysis", in *Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on*, IEEE, pp. 218-221.

Liu, Wei, Chunquan Li, Yanjun Xu, Haixiu Yang, Qianlan Yao, Junwei Han, Desi Shang, Chunlong Zhang, Fei Su, Xiaoxi Li, et al.

2013 "Topologically inferring risk-active pathways toward precise cancer classification by directed random walk", *Bioinformatics*, 29, 17, pp. 2169-2177.

Lonsdale, John, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, et al.

2013 "The genotype-tissue expression (GTEx) project", *Nature genetics*, 45, 6, p. 580.

Lopez-del Rio, Angela, Alfons Nonell-Canals, David Vidal, and Alexandre Perera-Lluna

2018 "Evaluation of cross-validation strategies in sequence-based binding prediction using Deep Learning", *Journal of chemical information and modeling*.

Lovász, László et al.

1993 "Random walks on graphs: A survey", *Combinatorics, Paul erdos is eighty*, 2, 1, pp. 1-46.

MacArthur, Jacqueline, Emily Bowler, Maria Cerezo, Laurent Gil, Peggy Hall, Emma Hastings, Heather Junkins, Aoife McMahon, Annalisa Milano, Joannella Morales, et al.

2016 "The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog)", *Nucleic acids research*, 45, D1, pp. D896-D901.

Malik-Sheriff, Rahuman S, Mihai Glont, Tung VN Nguyen, Krishna Tiwari, Matthew G Roberts, Ashley Xavier, Manh T Vu, Jinghao Men, Matthieu Maire, Sarubini Kananathan, et al.

2019 "BioModels–15 years of sharing computational models in life science", *Nucleic acids research*.

Marco-Ramell, Anna, Magali Palau-Rodriguez, Ania Alay, Sara Tulipani, Mireia Urpi-Sarda, Alex Sanchez-Pla, and Cristina Andres-Lacueva

2018 "Evaluation and comparison of bioinformatic tools for the enrichment analysis of metabolomics data", *BMC bioinformatics*, 19, 1, p. 1.

Massucci, Francesco Alessandro, Jonathan Wheeler, Raúl Beltrán-Debón, Jorge Joven, Marta Sales-Pardo, and Roger Guimerà

2016 "Inferring propagation paths for sparsely observed perturbations on complex networks", *Science advances*, 2, 10, e1501638.

Mendez, David, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, et al.

2019 "ChEMBL: towards direct deposition of bioassay data", *Nucleic acids research*, 47, D1, pp. D930-D940.

Mi, Huaiyu, Anushya Muruganujan, Dustin Ebert, Xiaosong Huang, and Paul D Thomas

2018 "PANTHER version 14: more genomes, a new PANTHER GO-slim and improvements in enrichment analysis tools", *Nucleic acids research*, 47, D1, pp. D419-D426.

Mitrea, Cristina, Zeinab Taghavi, Behzad Bokanizad, Samer Hanoudi, Rebecca Tagett, Michele Donato, Calin Voichita, and Sorin Draghici

2013 "Methods and approaches in the topology-based analysis of biological pathways", *Frontiers in physiology*, 4, p. 278.

Mordelet, Fantine and Jean-Philippe Vert

2011 "ProDiGe: Prioritization Of Disease Genes with multitask machine learning from positive and unlabeled examples", *BMC bioinformatics*, 12, 1, p. 389.

Mostafavi, Sara, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris

2008 "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function", *Genome biology*, 9, 1, S4.

Muslu, Oezlem, Charles Tapley Hoyt, Martin Hofmann-Apitius, and Holger Froehlich

2019 "GuiltyTargets: Prioritization of Novel Therapeutic Targets with Deep Network Representation Learning", *BioRxiv*, p. 521161.

O'Donnell, Valerie B., Edward A. Dennis, Michael J. O. Wakelam, and Shankar Subramaniam

2019 "LIPID MAPS: Serving the next generation of lipid researchers with tools, resources, data, and training", *Science Signaling*, 12, 563, ISSN: 1945-0877.

Ogris, Christoph, Dimitri Guala, Thomas Helleday, and Erik LL Sonnhammer

2016 "A novel method for crosstalk analysis of biological networks: improving accuracy of pathway annotation", *Nucleic acids research*, 45, 2, e8-e8.

Oliver, Stephen

2000 "Proteomics: guilt-by-association goes global", *Nature*, 403, 6770, p. 601.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd

1999 *The PageRank citation ranking: Bringing order to the web.* Tech. rep., Stanford InfoLab.

Papatheodorou, Irene, Nuno A Fonseca, Maria Keays, Y Amy Tang, Elisabet Barrera, Wojciech Bazant, Melissa Burke, Anja Füllgrabe, Alfonso Muñoz-Pomer Fuentes, Nancy George, et al.

2017 "Expression Atlas: gene and protein expression across multiple studies and organisms", *Nucleic acids research*, 46, D1, pp. D246-D251.

Paull, Evan O, Daniel E Carlin, Mario Niepel, Peter K Sorger, David Haussler, and Joshua M Stuart

2013 "Discovering causal pathways linking genomic events to transcriptional states using Tied Diffusion Through Interacting Events (TieDIE)", *Bioinformatics*, 29, 21, pp. 2757-2764.

Pedersen, Carsten Boecker, Jonas Bybjerg-Grauholm, Marianne Gioertz Pedersen, Jakob Grove, Esben Agerbo, Marie Baekvad-Hansen, Jesper Buchhave Poulsen, Christine Soeholm Hansen, JJ McGrath, Thomas D Als, et al.

2018 "The iPSYCH2012 case–cohort sample: new directions for unravelling genetic and environmental architectures of severe mental disorders", *Molecular psychiatry*, 23, 1, p. 6.

Peng, Roger D

2011 "Reproducible research in computational science", *Science*, 334, 6060, pp. 1226-1227.

Pierson, Emma, Daphne Koller, Alexis Battle, Sara Mostafavi, GTEx Consortium, et al.

2015 "Sharing and specificity of co-expression networks across 35 human tissues", *PLoS computational biology*, 11, 5, e1004220.

Piñero, Janet, Àlex Bravo, Núria Queralt-Rosinach, Alba Gutiérrez-Sacristán, Jordi Deu-Pons, Emilio Centeno, Javier García-García, Ferran Sanz, and Laura I Furlong

2016 "DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants", *Nucleic acids research*, gkw943.

Pratt, Dexter, Jing Chen, David Welker, Ricardo Rivas, Rudolf Pillich, Vladimir Rynkov, Keiichiro Ono, Carol Miello, Lyndon Hicks, Sandor Szalma, et al.

2015 "NDEx, the network data exchange", *Cell systems*, 1, 4, pp. 302-305.

R Core Team

2018 *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Rapaport, Franck, Andrei Zinovyev, Marie Dutreix, Emmanuel Barillot, and Jean-Philippe Vert

2007 "Classification of microarray data using gene networks", *BMC bioinformatics*, 8, 1, p. 35.

Regev, Aviv, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, et al.

2017 "Science forum: the human cell atlas", *Elife*, 6, e27041.

Roberts, David R, Volker Bahn, Simone Ciuti, Mark S Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, et al.

2017 "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure", *Ecography*, 40, 8, pp. 913-929.

Rodchenkov, Igor, Ozgun Babur, Augustin Luna, Bulent Arman Aksoy, Jeffrey V Wong, Dylan Fong, Max Franz, Metin Can Siper, Manfred Cheung, Michael Wrana, et al.

2019 "Pathway Commons 2019 Update: integration, analysis and exploration of pathway data", *Nucleic acids research*.

Saito, Takaya and Marc Rehmsmeier

2015 "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets", *PloS one*, 10, 3, e0118432.

Slenter, Denise N, Martina Kutmon, Kristina Hanspers, Anders Riutta, Jacob Windsor, Nuno Nunes, Jonathan Mélius, Elisa Cirillo, Susan L Coort, Daniela Digles, et al.

2017 "WikiPathways: a multifaceted pathway database bridging metabolomics to other omics research", *Nucleic acids research*, 46, D1, pp. D661-D667.

Smola, Alexander J and Risi Kondor

2003 "Kernels and regularization on graphs", in *Learning theory and kernel machines*, Springer, pp. 144-158.

Subramanian, Aravind, Rajiv Narayan, Steven M Corsello, David D Peck, Ted E Natoli, Xiaodong Lu, Joshua Gould, John F Davis, Andrew A Tubelli, Jacob K Asiedu, et al.

2017 "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles", *Cell*, 171, 6, pp. 1437-1452.

Subramanian, Aravind, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al.

2005 "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles", *Proceedings of the National Academy of Sciences*, 102, 43, pp. 15545-15550.

Suthram, Silpa, Andreas Beyer, Richard M Karp, Yonina Eldar, and Trey Ideker

2008 "eQED: an efficient method for interpreting eQTL associations using protein networks", *Molecular systems biology*, 4, 1.

Swainston, Neil, Kieran Smallbone, Hooman Hefzi, Paul D Dobson, Judy Brewer, Michael Hanscho, Daniel C Zielinski, Kok Siong Ang, Natalie J Gardiner, Jahir M Gutierrez, et al.

2016 "Recon 2.2: from reconstruction to model of human metabolism", *Metabolomics*, 12, 7, p. 109.

Szklarczyk, Damian, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al.

2018 "STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets", *Nucleic acids research*, 47, D1, pp. D607-D613.

Szklarczyk, Damian, Alberto Santos, Christian von Mering, Lars Juhl Jensen, Peer Bork, and Michael Kuhn

2015 "STITCH 5: augmenting protein–chemical interaction networks with tissue and affinity data", *Nucleic acids research*, 44, D1, pp. D380-D384.

Tarca, Adi L, Gaurav Bhatti, and Roberto Romero

2013 "A comparison of gene set analysis methods in terms of sensitivity, prioritization and specificity", *PloS one*, 8, 11, e79217.

Tarca, Adi Laurentiu, Sorin Draghici, Purvesh Khatri, Sonia S Hassan, Pooja Mittal, Jung-sun Kim, Chong Jai Kim, Juan Pedro Kusanovic, and Roberto Romero

2008 "A novel signaling pathway impact analysis", *Bioinformatics*, 25, 1, pp. 75-82.

Tomczak, Katarzyna, Patrycja Czerwińska, and Maciej Wiznerowicz

2015 "The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge", *Contemporary oncology*, 19, 1A, A68.

Tsuda, Koji, Hyunjung Shin, and Bernhard Schölkopf

2005 "Fast protein classification with multiple networks", *Bioinformatics*, 21, suppl_2, pp. ii59-ii65.

Türei, Dénes, Tamás Korcsmáros, and Julio Saez-Rodriguez

2016 "OmniPath: guidelines and gateway for literature-curated signaling pathway resources", *Nature methods*, 13, 12, p. 966.

UK10K consortium and others

2015 "The UK10K project identifies rare variants in health and disease", *Nature*, 526, 7571, p. 82.

Valentini, Giorgio, Giuliano Armano, Marco Frasca, Jianyi Lin, Marco Mesiti, and Matteo Re

2016 "RANKS: a flexible tool for node label ranking and classification in biological networks", *Bioinformatics*, 32, 18, pp. 2872-2874.

Valentini, Giorgio, Alberto Paccanaro, Horacio Caniza, Alfonso E Romero, and Matteo Re

2014 "An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods", *Artificial Intelligence in Medicine*, 61, 2, pp. 63-78.

Vandin, Fabio, Eli Upfal, and Benjamin J Raphael

2011 "Algorithms for detecting significantly mutated pathways in cancer", *Journal of Computational Biology*, 18, 3, pp. 507-522.

Vanunu, Oron, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan

2010 "Associating genes and protein complexes with disease via network propagation", *PLoS computational biology*, 6, 1, e1000641.

Wishart, David S, Yannick Djoumbou Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al.

2017 "DrugBank 5.0: a major update to the DrugBank database for 2018", *Nucleic acids research*, 46, D1, pp. D1074-D1082.

Wishart, David S, Yannick Djoumbou Feunang, Ana Marcu, An Chi Guo, Kevin Liang, Rosa Vázquez-Fresno, Tanvir Sajed, Daniel Johnson, Carin Li, Naama Karu, et al.

2017 "HMDB 4.0: the human metabolome database for 2018", *Nucleic acids research*, 46, D1, pp. D608-D617.

Wishart, David S, Carin Li, Ana Marcu, Hasan Badran, Allison Pon, Zachary Budinski, Jonas Patron, Debra Lipton, Xuan Cao, Eponine Oler, et al.

2019 "PathBank: a comprehensive pathway database for model organisms", *Nucleic acids research*.

Xia, Jianguo and David S Wishart

2010 "MetPA: a web-based metabolomics tool for pathway analysis and visualization", *Bioinformatics*, 26, 18, pp. 2342-2344.

Zerbino, Daniel R, Premanand Achuthan, Wasiu Akanni, M Ridwan Amode, Daniel Barrell, Jyothish Bhai, Konstantinos Billis, Carla Cummins, Astrid Gall, Carlos García Girón, et al.

2017 "Ensembl 2018", *Nucleic acids research*, 46, D1, pp. D754-D761.

Zhang, Wei, Jeremy Chien, Jeongsik Yong, and Rui Kuang

2017 "Network-based machine learning and graph theory algorithms for precision oncology", *NPJ precision oncology*, 1, 1, pp. 1-15.

Zoidi, Olga, Eftychia Fotiadou, Nikos Nikolaidis, and Ioannis Pitas

2015 "Graph-based label propagation in digital media: A review", *ACM Computing Surveys (CSUR)*, 47, 3, p. 48.

3 GOALS

3.1 MAIN OBJECTIVE

Diffusion scores are used in every discipline of computational biology that involves biological networks. On the other hand, concerns have arisen about the existence of a bias within the scores, potentially reaching numerous areas of active research. The main objective of this thesis is to develop, characterise and implement a statistical normalisation of diffusion scores. The potential benefits (or the absence thereof) will be assessed for salient problems in computational biology: pathway enrichment for metabolomics data and novel gene target discovery.

3.2 DETAILED OBJECTIVES

The main objective of this thesis can be achieved through three conceptual steps. First, a generic formulation of the normalisation used to address the bias. Then, its application to two computational biology domains: metabolomics data enrichment and prediction of sensible disease gene targets.

3.2.1 Conception of the statistical normalisation

- Characterise and understand the bias in diffusion scores.
- Define statistical models to normalise diffusion scores, focusing on providing a deterministic formulation.
- Give a general guideline about when and how should diffusion scores be normalised.

3.2.2 Application to metabolomics data enrichment

- Build a contextual representation linking metabolites to pathways as a knowledge graph.
- Define a diffusion-based enrichment method, examine the need of a statistical normalisation.
- Validate the method on in-house and public datasets.

3.2.3 Application to gene target discovery

- Define a validation framework suitable for structured data and a performance metric oriented to drug development.

- Benchmark diffusion-based methods on a protein interaction network, including unnormalised and normalised scores.
- Quantify the impact of the network choice and the disease under study.

3.3 EXPECTED CONTRIBUTIONS

The main contribution will revolve around quantifying the presence of bias within the diffusion algorithms and providing ways to address it. On the other hand, the knowledge graph for metabolomics data enrichment has its interest per se, as it delivers a new paradigm for data interpretation.

Every detailed objective is expected to lead to one or more publications in indexed scientific journals. To encourage open and reproducible science, all the algorithms and models will be released as free, open source tools. They will be encapsulated in R packages with extensive documentation and published in the Bioconductor repository.

4 STATISTICAL PROPERTIES

THE EFFECT OF STATISTICAL NORMALISATION ON DIFFUSION SCORES IN COMPUTATIONAL BIOLOGY

Network diffusion and label propagation are fundamental tools in computational biology, with applications like gene-disease association, protein function prediction and module discovery. More recently, several publications have introduced a permutation analysis after the propagation process, due to concerns that network topology can bias diffusion scores. This opens the question of the statistical properties and the presence of bias of such diffusion processes in each of its applications. In this work, we characterised some common null models behind the permutation analysis and the statistical properties of the diffusion scores. We benchmarked seven diffusion scores on three case studies: synthetic signals on a yeast interactome, simulated differential gene expression on a protein-protein interaction network and prospective gene set prediction on another interaction network. For clarity, all the datasets were based on binary labels, but we also present theoretical results for quantitative labels.

Diffusion scores starting from binary labels were affected by the label codification, and exhibited a problem-dependent topological bias that could be removed by the statistical normalisation. Parametric and non-parametric normalisation addressed both points by being codification-independent and by equalising the bias. We identified and quantified two sources of bias -mean value and variance- that yielded performance differences when normalising the scores. We provided closed formulae for both and showed how the null covariance is related to the spectral properties of the graph. Despite none of the proposed scores systematically outperformed the others, normalisation was preferred when the sought positive labels were not aligned with the bias. We conclude that the decision on bias removal should be problem and data-driven, i.e. based on a quantitative analysis of the bias and its relation to the positive entities. The code is publicly available at https://github.com/b2slab/diffuBench

4.1 INTRODUCTION

The guilt by association principle states that two proteins that interact with one another are prone to participate in the same, or related, cellular functions (Oliver, 2000). This cornerstone fact has motivated the exploration

This chapter is a reproduction of the following preprint, with minor section title changes: Picart-Armada, Sergio, Wesley K. Thompson, Alfonso Buil, and Alexandre Perera-Lluna. "The effect of statistical normalisation on network propagation scores". *BioRxiv (2020)*.

of network algorithms on interaction networks for protein function prediction (Sharan et al., 2007). Network analysis has further proven its usefulness in other computational biology problems, such as prioritising candidate disease genes (Barabási et al., 2011), finding modular structures (Mitra et al., 2013) and modelling organisms (Aderem, 2005).

Network propagation is a fundamental formalism to leverage network data in computational biology. Its theoretical basis revolves around graph spectral theory, graph kernels and random walks (Smola and Kondor, 2003). The central concept is that nodes carry abstract labels that, following the guilt by association principle, are propagated to the neighbouring nodes (Zoidi et al., 2015). Unlabelled nodes can therefore be inferred a label based on the available data of their neighbours. Label propagation can be defined in several ways, such as the heat diffusion, the electrical model or random walks with restarts (RWR), some of which lead to equivalent formulations (Cowen et al., 2017).

One of the most common diffusion formulations relies on the regularised Laplacian graph kernel (Smola and Kondor, 2003) - examples are provided throughout this paragraph. HotNet (Vandin et al., 2010) is a tool for finding modules with a statistically high number of mutated genes in cancer, after propagating the labels of mutated genes. The authors in (Bersanelli et al., 2016) have found relevant modules from gene expression and mutation data, based on a diffusion process followed by an automatic subgraph mining. GeneMANIA (Mostafavi et al., 2008) is a web server that predicts gene function by optimising a combination of knowledge networks and running a diffusion process on the resulting network. TieDIE (Paull et al., 2013) defines two diffusion processes in order to connect two sets of genes, applied to link perturbation in the genome with changes in the transcriptome. More generally, the predictive power of label propagation using graph kernels has been benchmarked in gene-disease association (Guala and Sonnhammer, 2017; Lee et al., 2011; Valentini et al., 2014).

Some studies have pointed out biases in diffusion scores and explored the effect of their removal. The authors of DADA (Erten et al., 2011) have found that prioritisation using RWR favours highly connected genes and suggest several normalisation strategies. One of them computes a z-score that adjusts for the mean value and standard deviation estimated from propagation scores from random degree-preserving inputs. Another possibility is to normalise diffusion scores into empirical p-values, as used in the diffusion of t-statistics derived from gene expression (Cun and Fröhlich, 2013). The aim was to quantify robust biomarkers, whose diffusion score is unlikely to arise from a permuted input. In the discovery of enriched modules (Bersanelli et al., 2016), the effect of the topology has been mitigated by combining diffusion scores with their empirical p-values. Similarly, exact z-scores and empirical p-values have been used for pathway analysis of metabolomics data (Sergio Picart-Armada, Fernández-Albert, et al., 2017). A recent study (Biran et al., 2019) has normalised RWR into an empirical p-value, obtained from edge rewiring. Specifically, random degree-preserving networks have been built to re-run the propagation and draw values from the null distributions of scores. Another recent manuscript (Hill et al., 2019) highlights biases in certain network propagation algorithms, related to the node degree.

Overall, a variety of measures to address the bias have emerged, but a systematic quantification and evaluation of the biases is missing. The normalisation can potentially backfire, for instance by missing highly connected nodes that are associated with the property under study (Erten et al., 2011). The goal of this manuscript is to provide a quantitative way to assess the presence of the bias and its alignment with the node labels, in order to understand the impact and adequateness of the normalisation.

4.2 APPROACH

Here, we address basic statistical properties of the normalisation of single-network diffusion scores to remove topology-related biases. We define and quantify two sources of bias. Both are derived from a statistical standpoint, based on the exact means and variances of the null distributions of the diffusion scores under input permutation. Differences in mean values between nodes should be the first indicator of systematic advantages: nodes with highest means will often be prioritised over those with lowest means. In their absence, differences in variances should be examined instead, as nodes with highest spread can be more likely to reach extreme scores. We compare classical and normalised propagation, as implemented in diffuStats (Sergio Picart-Armada, Thompson, et al., 2017), in data with and without bias. The main results are derived for the commonly used regularised Laplacian kernel, although most of them apply to other graph kernels and, to a lesser extent, to random walks with restarts. Special emphasis is placed on identifying scenarios under which normalisation is beneficial or detrimental and on understanding the underlying reasons why.

4.3 METHODS

We include seven diffusion scores that are part of the diffuStats package (Sergio Picart-Armada, Thompson, et al., 2017): f_{raw}, f_{ml}, f_{gm}, f_{ber_s}, f_{mc}, f_z and f_{ber_p}. These scores are variations of the original diffusion model with a regularised unnormalised Laplacian kernel (Smola and Kondor, 2003). Labelled nodes are referred to as positives if they have the property of interest, and negatives otherwise.

4.3.1 Unnormalised scores

The starting point is the f_{raw} score, which requires a graph kernel K (Smola and Kondor, 2003) and input vector y_{raw} and is computed as:

$$f_{raw} = K y_{raw} \tag{27}$$

This work focuses on the unnormalised, regularised Laplacian kernel for K, for being a widespread choice in the computational biology literature (electrical model, heat or fluid propagation). The values in y_{raw} reflect

the weights of each type of node: 1 for positives and 0 for negative and unlabelled entities.

f_{ml} and f_{gm} differ from f_{raw} by setting a weight of -1 on negative nodes. f_{gm} also weighs unlabelled nodes with a bias term adapted from GeneMANIA (not to be confused with the diffusion bias). On the other hand, f_{ber_s} measures the relative change between f_{raw} and y_{raw}, with a moderating parameter ϵ:

$$f_{ber_s}(i) = \frac{f_{raw}(i)}{y_{raw}(i) + \epsilon} \tag{28}$$

4.3.2 Normalised scores

Normalised scores attempt to equalise nodes that systematically show low or high scores, regardless of the input and due to the specific topology of the network. The lynchpin of normalisation is the null distribution of the diffusion scores under a random permutation π of the labelled nodes. The null scores arise from applying f_{raw} to a randomised input $X_y = \pi(y_{raw})$ and comparing, for the i-th node, $f_{raw}(i)$ to its null distribution $X_f(i)$, where $X_f = KX_y$. An empirical p-value can be computed throught Monte Carlo trials for the i-th node on N trials:

$$p(i) = \frac{r_i + 1}{N + 1}, \tag{29}$$

where r_i is the number of randomised trials having an equal or higher diffusion score in node i. In order to assign high scores to relevant nodes, the score is defined as $f_{mc}(i) = 1 - p(i)$. We also include a parametric alternative to f_{mc} by computing z-scores for each node i:

$$f_z(i) = \frac{f_{raw}(i) - E(X_f(i))}{\sqrt{Var(X_f(i))}} \tag{30}$$

The expected value and variance of the null distributions are analytically determined (see Supplement 1). Thus, f_z has a computational advantage over Monte Carlo trials.

Finally, a hybrid combining an unnormalised and a normalised score is provided, inspired by how (Bersanelli et al., 2016) moderated the effect of hubs: f_{raw}: $f_{ber_p}(i) = -\log_{10}(p(i)) f_{raw}(i)$.

4.3.3 Metrics and baselines

Two baseline methods were used. First `pagerank`, regarded as an input-naïve centrality measure (default damping factor of 0.85), to measure the predictive power of a basic network property. Second, a `random` predictor, to set an absolute baseline. Performances were quantified with two metrics: the area under the Receiver Operating Characteristic curve (AUROC) and the area under the Precision-Recall curve (AUPRC), as implemented in the precrec package (Saito and Rehmsmeier, 2017). For clarity, the ranking (ordering) of the nodes for any given score and instance was normalised to lie

in $[0, 1]$ by dividing it by the number of ranked nodes, so that top suggestions corresponded to ranks close to 0.

4.3.4 Bias quantification

The reference expected value of the i-th node $b^{\mathcal{K}}_{\mu}(i)$ (eq. 33) was defined as proportional to the expected value of its null distribution $X_f(i)$ (eq. 31). Reference expected values that vary across nodes can indicate systematic differences in the diffusion scores of such nodes.

In the absence of differences in the reference expected value, variance-related bias was analysed instead. The reference variance of the i-th gene $b^{\mathcal{K}}_{\sigma^2}(i)$ (eq. 34) was defined as, up to an additive constant, the base 10 logarithm of the variance of $X_f(i)$, straightforward to obtain from the covariance matrix (eq. 32). The rationale is that the scores of nodes with varying dispersion measures should not be compared directly.

4.3.5 Performance explanatory models

Explanatory models have found use in the formal description of differences in performance as a function of design factors (Lopez-del Rio et al., 2019; S Picart-Armada et al., 2019). Following (S Picart-Armada et al., 2019), the trends in AUROC and AUPRC were described through logistic-like quasibinomial models with a logit link function, as a generalisation of logistic models to prevent over and under-dispersion issues.

Table 4 presents the main model for each case study. The categorical regressors were: `method`, `metric` (AUROC or AUPRC), `biased` (refers to the signal, true or false), `strat` (labelled, unlabelled or overall), `array` (ALL or Lym), and the parameters `k`, `r` and `p_max` for the second case study. `path_var_ref` was quantitative, equal to the reference pathway variance $bp^{\mathcal{K}}_{\sigma^2}$ (eq. 35). The responses were either AUROC, AUPRC, or both mixed, the latter denoted by `Performance`.

4.4 MATERIALS

The evaluation of the diffusion scores was performed on three datasets of different nature, as described in Table 4: (1) synthetic signals on a yeast interactome, (2) pathway-based synthetic signals on a human network and (3) real signals on another human network.

Table 4: Case studies for characterising biases and benchmarking diffusion scores. Interactions in explanatory models are denoted by a colon.

Case	Network	Positive nodes	Signal	Bias type	Purpose	Explanatory model for hypothesis testing
(1)	Yeast	Synthetic	Synthetic, bias-based	Mean value	Proof of concept	Performance ~ method + method : biased + metric
(2)	HPRD	KEGG pathways	Pathway sub-sampling	Mean value	Background influence in bias	AUPRC ~ method + method : strat + array + k + r + p_{max}
(3)	BioGRID	KEGG pathways	Prospective pathway prediction	Variance	Bias in a common scenario	AUROC ~ method + method : path_var_ref

4.4.1 Networks

Yeast network

A small yeast network was used to demonstrate the casuistic of diffusion scores properties. Medium and high confidence interactions from several sources were provided by the original study (Von Mering et al., 2002), as found in the igraphdata R package (Csardi, 2015). It contains 2,617 proteins and 11,855 unweighted edges, but we worked only with its largest connected component (2,375 proteins, 11,693 edges).

HPRD network

The diffuse large B-cell lymphoma study, available in the R package DLBCL (Dittrich and Beisser, 2010), contains a differential expression dataset accompanied by a human interactome network extracted from the Human Protein Reference Database, HPRD (Mishra et al., 2006). The original network encompasses 9,385 proteins with 36,504 interactions, whose largest connected component (8,989 nodes, 34,325 interactions) was extracted to compute the diffusion scores.

We derived two gene backgrounds based on expression arrays. The first background (Lym) was taken from the expression data from 2,557 genes (2,482 in the network) in the lymphoma study (Rosenwald et al., 2002). The second background (ALL) was based on the acute lymphocytic leukemia array (Chiaretti et al., 2004), available in the ALL R package (Li, 2009), encompassing 6,133 genes (5,921 in the network).

BioGRID network

The Biological General Repository for Interaction Datasets (BioGRID) (Chatraryamontri et al., 2017) is a public database with curated genetic and protein interaction from Homo sapiens and other organisms. BioGRID was retrieved in January 2017, but only keeping interactions dating from 2010 or older. The interactions were weighted according to (Cao et al., 2014), under the assumptions that more publications about an interaction boost its confidence and that low-throughput technologies are more reliable that high-throughput ones. The network encompassed 11,394 nodes and 67,573 edges and was connected.

4.4.2 Datasets

Synthetic bias-based dataset

100 biased and 100 unbiased instances of positive, negative and unlabelled nodes were generated in dataset (1) from table 4, by sampling positive nodes with probabilities proportional to biased and unbiased scores. By construction, the frequencies of the positives drawn for biased signals were positively correlated with the reference expected value, whereas those of the unbiased signals were uncorrelated with it.

Nodes were partitioned into three equally sized pools, from which positive nodes were drawn: (a) labelled nodes that were fed to the diffusion methods, (b) target nodes, the ones to be ranked and whose ground truth was known, and (c) filler nodes that were neither target nor labelled.

For each instance, a fixed fraction of labelled nodes x_e were uniformly sampled as positives, the rest of labelled nodes were deemed negatives and the target and filler nodes were left unlabelled. This input served two purposes: generate the ground truth in target nodes, and be the input for all the diffusion scores.

To generate the ground truth in target nodes of biased signals, the `raw` diffusion scores were computed from the input above. A fixed fraction of target nodes x_s was sampled with probabilities proportional to their raw scores, i.e. $p(i) \propto f_{raw}(i)$, to become positives. The remaining target nodes would remain negatives, completing the ground truth. The regularised unnormalised Laplacian kernel is endorsed by physical models that ensure $f_{raw}(i) > 0$ provided that inputs have one or more positives and the graph is connected. Analogously, unbiased signals were generated by sampling a fraction of target nodes x_s, but with probabilities roughly proportional to the unbiased diffusion scores `mc`: $p(i) \propto f_{mc}(i) + \frac{1}{N+1}$. By definition, the frequency of appearance of the target nodes was independent of the bias, and the small offset ensured $p(i) > 0$.

In both cases, after sampling the ground truth, the same input was used again for all the diffusion scores, in order to rank the target nodes and compute the corresponding AUROC and AUPRC.

Pathway sub-sampling dataset

Synthetic gene expression statistics were generated, based on pathways in the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al., 2017), and on two array-based gene backgrounds described within the HPRD network. Genes outside the background were hidden (unlabelled), and genes inside were given p-values for differential expression.

Each signal derived from k random KEGG pathways. The pathways were assumed to be affected as a whole, but only a sampled portion of r genes showed differential expression patterns. The p-values of the differential expressed genes were uniformly sampled from $[0, p_{max}]$, whereas the rest of genes were uniform in $[0, 1]$, following a previously study (Rajagopalan and Agarwal, 2004).

For both expression arrays, genes with an FDR $< 10\%$ within their background were used as positives, the remaining background genes as negatives and the hidden nodes were deemed unlabelled. Notice that, by definition, this procedure generated false positives and false negatives among the input genes.

The target genes were those belonging to the k affected pathways, including those with no apparent differential expression and those among the unlabelled nodes. Methods were compared using the AUROC and AUPRC, computed separately on labelled, unlabelled genes, and overall, on a grid of parameters: $k \in \{1, 3, 5\}$, $r \in \{0.3, 0.5, 0.7\}$ and $p_{max} \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. For each combination of parameters, $N = 50$ instances were simulated.

Prospective pathway dataset

The input lists consisted of the genes in 139 KEGG pathways from 14th March, 2011. The target genes were the newly added genes in the same KEGG pathways in 18th August, 2018 release. The 139 pathways had new genes in the latter release after mapping to the network.

AUROC and AUPRC were computed on each pathway, always excluding the input positive genes. The bias was examined at the pathway level, assessing whether the properties of their new genes differed from those of the rest of network genes. It was defined as the median reference variance of its new genes minus the median reference variance of all the genes besides old and new pathway genes (eq. 35).

4.5 RESULTS

4.5.1 Properties of diffusion scores

Some of the diffusion scores are equivalent in certain scenarios. In the absence of unlabelled nodes and using kernels based on the unnormalised graph Laplacian, f_{raw}, f_{ml} and f_{gm} lead to an identical node prioritisation. More generally, the results using only two classes (and therefore two real values $y^+ > y^-$ as weights) always lead to the same ranking as f_{raw}. An analogous result holds for the weights of the positives and the unlabelled, $y^+ > y^u$, in the absence of negative nodes.

The normalised scores f_{mc} and f_z are invariant to changes in the weights of the positive and negative examples, regardless of the presence of unlabelled nodes and the graph kernel. This property simplifies the diffusion setup and leads to weight-independent results. Along with eqs. 31 and 32, this holds even if the matrix K in eq. 27 is not a kernel, like the random walk similarity matrices in (Cowen et al., 2017).

We also provide the closed form of the null expected value and covariance matrix of the raw scores, governed by the identifiers of the n_l labelled nodes (out of n). If $\mathcal{K}$ contains only their corresponding columns from K, and $\mathcal{Y}$ is the input vector y_{raw} restricted to them, then:

$$\mathbb{E}(X_f) = \mu_y \mathcal{K} \mathbb{1}_{n_l} \tag{31}$$

$$\Sigma(X_f) = \sigma_y^2 \mathcal{K} M_{n_l} \mathcal{K}^T \tag{32}$$

$\mu_y = \frac{1}{n_l}\sum_{i=1}^{n_l} y_i$ and $\sigma_y^2 = \frac{1}{n_l - 1}\sum_{i=1}^{n_l}(y_i - \mu_y)^2$ are the mean and variance of the labels. $M_k = I_k - \frac{1}{k}\mathbb{1}_k\mathbb{1}_k^T$, being I_k the $k \times k$ identity matrix and $\mathbb{1}_k$ the column vector with k ones.

If a graph kernel based on the unnormalised Laplacian is used, the covariance of the null distribution (eq. 32) is closely related to the spectral properties of the labelled nodes. In particular, in the absence of unlabelled nodes, the leading eigenvector of the null covariance is, up to a sign change, the Fiedler-vector, commonly used for graph clustering (Smola and Kondor, 2003). The statistical normalisation is therefore endowed with a topological

basis. This sheds light on prior empirical observations that, even when the bias can relate to the node degree, there must be further topological factors involved (Hill et al., 2019).

Because μ_y and σ^2_y are multiplicative constants and inherent to the labels, the topology-related mean value and variance references of the i-th node are defined as follows. We assume $n_l \geqslant 2$ because if $n_l \in \{0, 1\}$ there is nothing to permute.

$$b^{\mathcal{K}}_{\mu}(i) := [\mathcal{K}\mathbb{1}_{n_l}]_{i1} = \sum_{j=1}^{n_l} \mathcal{K}_{ij} \tag{33}$$

$$b^{\mathcal{K}}_{\sigma^2}(i) := \log_{10}\left([\mathcal{K}M_{n_l}\mathcal{K}^T]_{ii}\right) = \log_{10}\left(\sum_{j=1}^{n_l}\left(\mathcal{K}_{ij} - \frac{b^{\mathcal{K}}_{\mu}(i)}{n_l}\right)^2\right) \tag{34}$$

Eq. 31 implies that there are two scenarios free of the expected value bias: $\mu_y = 0$ (centered input), or $n_l = n$ and a kernel K based on the unnormalised Laplacian, rendering $b^{\mathcal{K}}_{\mu}$ constant (see Supplement 1). The i-th null variance (eq. 32) can be exactly zero, either because $\sigma^2_y = 0$ (constant input), or because the topology forces $[\mathcal{K}M_{n_l}\mathcal{K}^T]_{ii} = 0$. In practice, the latter is expected to happen in small connected components without any labelled nodes. Both cases render the i-th score constant, therefore lacking interest, and leave f_z undefined.

In the retrospective dataset, the reference of a given pathway P, conceived to summarise its properties into a single number, was defined by subtracting the median reference of its new genes, new(P) to that of the genes that never belonged to it, others(P):

$$bp^{\mathcal{K}}_{\sigma^2}(P) := \underset{i\in\text{new}(P)}{\text{median}}\{b^{\mathcal{K}}_{\sigma^2}(i)\} - \underset{i\in\text{other}(P)}{\text{median}}\{b^{\mathcal{K}}_{\sigma^2}(i)\} \tag{35}$$

The mathematical proofs of the properties and illustrative examples can be found in Supplement 1.

4.5.2 Synthetic signals in yeast

Bias in diffusion scores

Supported by eq. 31, the presence of unlabelled nodes originated different expected values among the nodes. We hypothesised that f_{raw} would be biased to favour nodes with high $b^{\mathcal{K}}_{\mu}$, whereas f_{mc} and f_z would prioritise in a more unbiased manner. Figure 12A confirms both trends. The data imbalance (negatives outnumbered positives in the input) had the opposite biasing effect on f_{ml}, favouring nodes with low $b^{\mathcal{K}}_{\mu}$.

Performance

In biased signals, target nodes with higher $b^{\mathcal{K}}_{\mu}$ were sampled as positives more often (see Supplement 2), which (i) benefited the unnormalised scores `raw` over `z`, and (ii) endowed the `pagerank` baseline with predictive power. Unbiased signals led to a uniform density of positives across $b^{\mathcal{K}}_{\mu}$, which (iii)

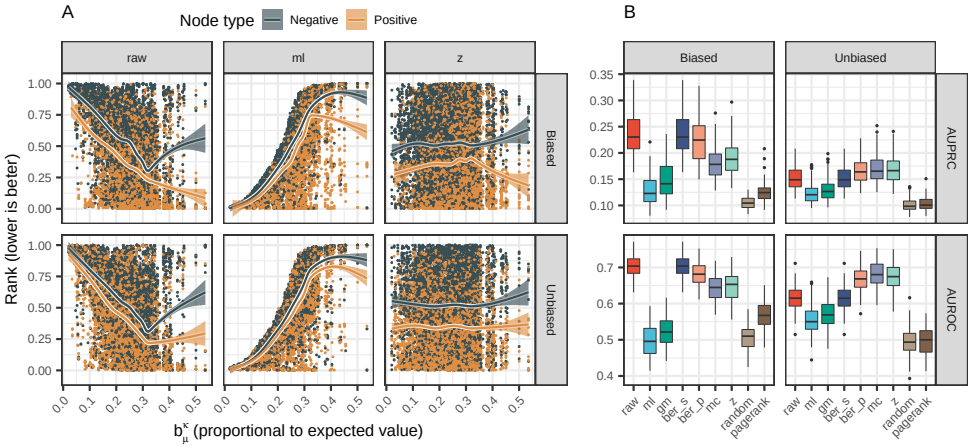


Figure 12: Analysis of biased and unbiased synthetic signals on the yeast network. Nodes showed a mean value-related bias, see Supplement 2. **(A)** Effects of the mean value bias in on the average node ranking, under biased and unbiased signals. Lines correspond to Generalised Additive Models with $y \sim s(x, bs = \text{"cs"})$ and 0.95 confidence intervals. `raw` and `ml` tended to find positives with high and low $b_\mu^\mathcal{K}$, respectively. `z` found positives in a more uniform manner. **(B)** Performance in terms of AUROC and AUPRC. `raw` was better suited for biased signals, for which the `pagerank` baseline also outperformed a random predictor. Conversely, `z` worked best on unbiased signals.

was better handled by `z` than by `raw` (figure 12B). Claims (i), (ii) and (iii) were statistically significant for AUROC and AUPRC (Tukey's method, FDR $< 10^{-10}$ in all cases, see Supplement 2). Also, f_{ber_p} was a good compromise between `raw` and `z`.

Based on these results, we suggest a systematic criterion to choose whether to normalise in the general case, by assessing (1) the presence of the expected value-related bias by checking if $b_\mu^\mathcal{K}$ is constant among the nodes to be prioritised, and (2) the expected or hypothetical dependence between $b_\mu^\mathcal{K}$ and the labels to be predicted. In this proof of concept, differences in $b_\mu^\mathcal{K}$ bias were present and normalisation was discouraged when $b_\mu^\mathcal{K}$ was aligned with the positives. If $b_\mu^\mathcal{K}$ is constant, $b_{\sigma^2}^\mathcal{K}$ should be examined instead, see the retrospective pathway dataset.

4.5.3 Simulated differential expression

Bias in diffusion scores

Analogously to the yeast dataset, the presence of unlabelled nodes led to differences in $b_\mu^\mathcal{K}$ among nodes, see figure 13A. We hypothesised that the main source of bias would arise from such heterogeneity, i.e. that unnormalised scores would be prone to find positives among highest expected values. In both arrays, the nodes belonging to one or more pathways had, compared with nodes outside, (i) larger $b_\mu^\mathcal{K}$ within the unlabelled genes, but (ii) lower $b_\mu^\mathcal{K}$ within the labelled nodes. Overall, (iii) labelled genes showed

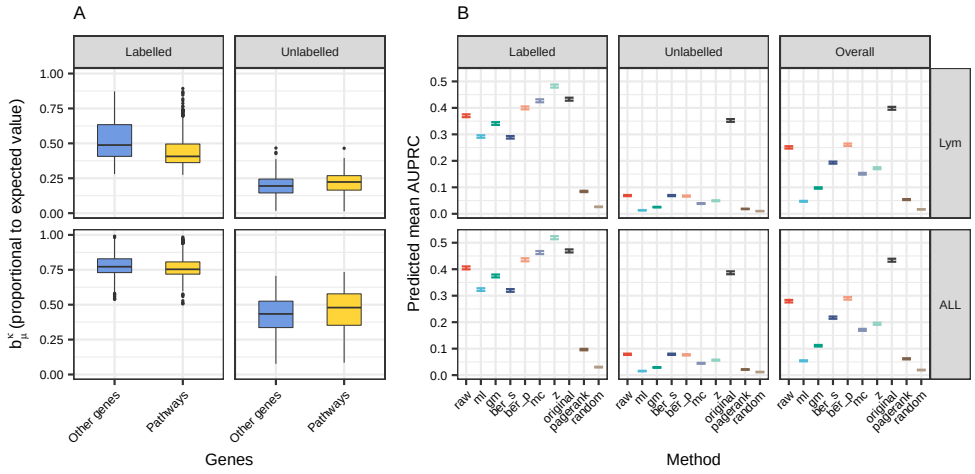


Figure 13: Performance in the DLBCL dataset. **(A)** Expected value-related bias. Within the labelled genes of both arrays, those in pathways had lower $b_\mu^{\mathcal{K}}$ that those outside. Within the unlabelled genes, this tendency was inverted. Overall, labelled genes had higher $b_\mu^{\mathcal{K}}$ than unlabelled genes. **(B)** Predicted AUPRC (0.95 confidence interval) using the explanatory model in Table 4 and Supplement 3. Besides diffusion scores, three baselines were included: `original` (ranking by the p-values), `pagerank` and `random`. In both arrays (*ALL* and *Lym*), `raw` outperformed `z` in unlabelled nodes and overall, while `z` was preferable in the labelled genes.

larger $b_\mu^{\mathcal{K}}$ than unlabelled genes. Figure 13A portrays the claims (i), (ii) and (iii) in both arrays – the six statements were significant with $p < 10^{-16}$, two-sided Wilcoxon test (see Supplement 3).

Performance

The performance, as predicted by the explanatory models, was influenced by the background used to compute the metrics, especially for AUPRC. Taking as reference f_{raw} and f_z, `raw` performed best in the unlabelled background and overall whereas `z` was preferable in the labelled background (figure 13B). The three claims were significant in both arrays (Tukey's method, $p < 10^{-10}$, see Supplement 3).

Differences in performance were consistent with the expected value-related bias: potential positives suffered from lower $b_\mu^{\mathcal{K}}$ in the labelled genes and benefited from greater $b_\mu^{\mathcal{K}}$ in the unlabelled part. In views of this, the natural choices were `z` and `raw`, respectively.

To understand why `raw` outperformed `z` in overall performance, note how by hypothesis the top candidates from `raw` should come from the labelled genes due to their high $b_\mu^{\mathcal{K}}$ against the unlabelled genes, whilst `z` should equalise predictions from both backgrounds. Predictions from the labelled part were more reliable owing to the presence of prior data on the genes (figure 13B). `z` equalised both backgrounds, shuffling reliable and unreliable predictions, and undermined overall performance.

Finally, an indirect assessment of the bias (PageRank centrality) fell short to explain performance differences in (i) and suggested that biased scores

were preferrable in the three cases, see Supplement 3. This highlights the importance of using a precise quantification of the bias.

4.5.4 Prospective pathway prediction

Bias in diffusion scores

Here, $b_{\mu}^{\mathcal{K}}$ was constant among all the nodes, as a consequence of using the unnormalised Laplacian without unlabelled nodes (see Supplement 1). Differences still existed in terms of $b_{\sigma^2}^{\mathcal{K}}$ (figure 14A), implying that the normalisation would make a difference.

However, the interpretation of the normalisation impact was not as straightforward as for the expected value bias. With the paradigm of the z-scores z, deviations from the expected value exacerbate under small variances and shrink under large variances. Notice how this does not imply the natural hypothesis that nodes with larger variances (resp. smaller) must drop (resp. rise) in the ranking, because ranking modifications take place around the mean. Figure 14B reflects how z actually recovered more high-variance positive nodes than `raw`.

Similarly to prior observations from figure 12A, the normalised scores tended to find the positives in a less biased manner. Positive nodes with a high variance were rarely found by `raw`, whereas z distributed them more evenly along the ranking (figure 14B). This improvement came at the cost of missing positives with lower variances.

Performance

The properties of the diffusion scores helped simplify this case study, as f_{ml}, f_{gm} and f_{ber_s} were left out for being redundant with f_{raw}. f_{ml} and f_{gm} for using the unnormalised Laplacian without unlabelled nodes, and f_{ber_s} because the genes to be prioritised were always labelled as negative in the input (see corollary 1 and proposition 3 in Supplement 1).

The prospective prediction of pathway genes was a challenging task, given the low predicted AUPRCs for all the methods (see Supplement 4). On the other hand, AUROC conveyed a richer view of the differences between methods. The explanatory model (figures 14C, 14D) showed that unnormalised scores were more affected by the presence of bias, reflected in the larger magnitude of their interaction terms (-1.387 for `raw` against -0.484 for z, $p < 10^{-4}$, Tukey's method). Overall, the casuistic among the bias of new pathway genes favoured z over `raw` (FDR $= 5.39 \cdot 10^{-9}$, two-sided paired Wilcoxon test). This conclusion did not apply to early retrieval, as it could not be proven for AUPRC (FDR $= 0.701$).

The negative sign of the interaction terms was also insightful: all the proper methods encountered more difficulties in finding loosely connected genes. This was expected, since there is less network data involving such genes, translating into unreliable predictions.

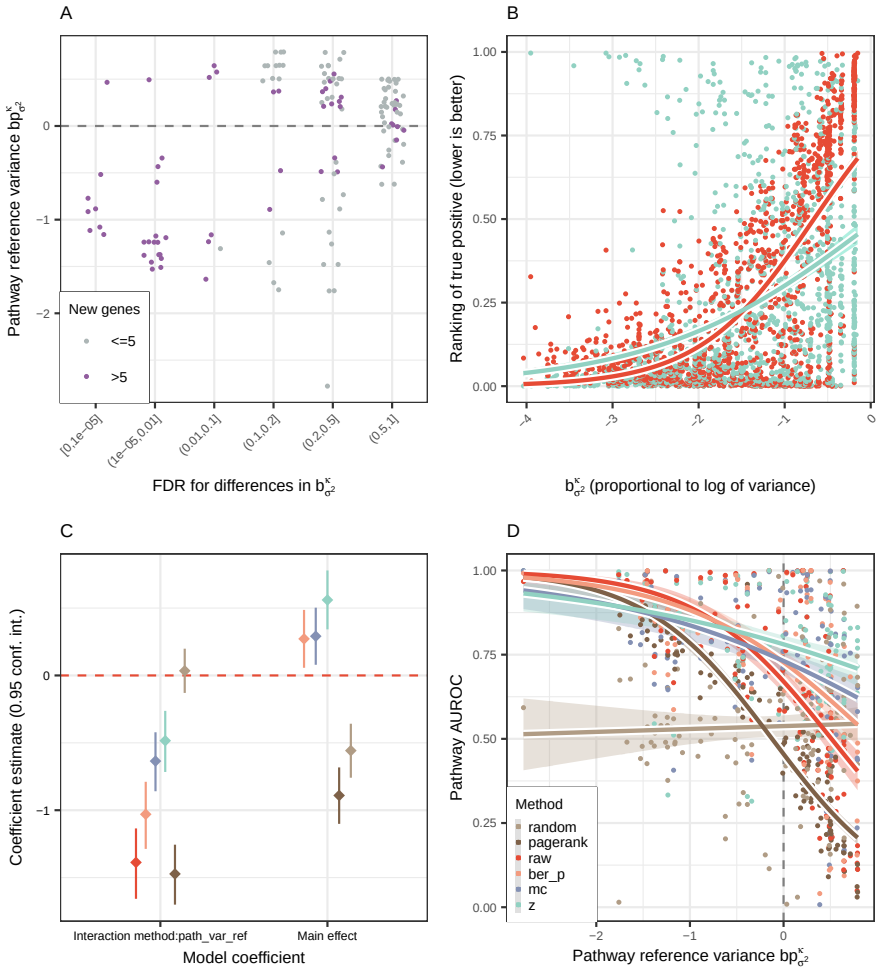


Figure 14: Analysis of the prospective dataset. **(A)** Pathway-wise comparison of new genes against the remaining genes outside the pathway, in terms of $b_{\sigma^2}^{\mathcal{K}}$. Several pathways showed significant differences in both directions (two-sided Wilcoxon test). The x axis was jittered for clarity. **(B)** Ranking of the positives using `raw` and `z`. Each data point is the relative ranking of a positive gene in one of the pathways, i.e. before computing pathway-level metrics. Lines correspond to a quasi-logistic fit with a 0.95 confidence interval. `raw` scores were more sensitive at low standard deviations, whereas `z` stood more uniform. **(C)** Coefficients of the model AUROC ~ method + method : path_var_ref with a 0.95 confidence interval, where the interaction term involved the variance bias. The main effect of `raw` was not depicted because it was the reference level of `method`. **(D)** Predicted AUROC across all the pathways, as a function of the bias. `z` was less sensitive to the bias, due to its interaction term in **(C)** being closer to 0. Lines correspond to a quasi-logistic fit with a 0.95 confidence interval.

4.6 CONCLUSION

In this study, we ratified that diffusion scores are biased due to the graph topology. We introduced two direct quantifications of the bias, in terms of the expected value and variance of the null distribution of the diffusion scores under input permutation. We analysed the benefits and pitfalls of using unbiased, statistically normalised scores and discussed several choices of the label weights when defining the diffusion process.

We proved equivalences between scores under certain conditions, helping simplify the setup of the diffusion, and discovered that normalised alternatives are invariant under label weights changes. We found an explicit link between principal directions of the null covariance and the spectral features of the network.

We applied the diffusion-based prioritisation on three scenarios: two with a mean value-related bias and one with a variance-related bias. Class imbalance and node topology had an impact in unnormalised scores, whereas normalised scores were more robust to both phenomena given their weight-independent definition. The parametric normalisation requires no permutations compared to Monte Carlo trials and performed equally or better, providing a convenient way to normalise. While mean value bias was straightforward to characterise, variance bias was less intuitive albeit of noticeable impact. In general terms, the statistical normalisation is advised if the positives are not aligned with the bias, and discouraged otherwise. The statistical background, i.e. which nodes are permuted, is a key piece that should be clearly stated in every application. Bias assessment should be carried through its direct quantification instead of indirect indicators, which can be misleading.

We conclude that the statistical normalisation can be beneficial or detrimental, and the decision should follow from the dependence between the node bias and the hypothetical or desired properties of the new positives. Topology-related bias can manifest in different ways (mean value- or variance-related bias) and each instance should be properly characterised.

ACKNOWLEDGEMENTS

SP thanks Guillem Belda-Ferrín for reviewing the mathematical proofs. SP thanks Imanol Morata Martínez and Camellia Sarkar for fruitful discussions and useful suggestions.

Conflict of Interest: none declared

FUNDING

This work was supported by the Spanish Ministry of Economy and Competitiveness (MINECO) [TEC2014-60337-R and DPI2017-89827-R to A.P.] and the National Institutes of Health (NIH) [R01GM104400 to W.T.]. AP and SP thank CIBERDEM and CIBER-BBN for funding, both initiatives of the Spanish ISCIII. SP thanks the AGAUR FI-scholarship programme.

REFERENCES

Aderem, Alan

2005 "Systems biology: its practice and challenges", *Cell*, 121, 4, pp. 511-513.

Barabási, Albert-László, Natali Gulbahce, and Joseph Loscalzo

2011 "Network medicine: a network-based approach to human disease", *Nature reviews. Genetics*, 12, 1, p. 56.

Bersanelli, Matteo, Ettore Mosca, Daniel Remondini, Gastone Castellani, and Luciano Milanesi

2016 "Network diffusion-based analysis of high-throughput data for the detection of differentially enriched modules." *Scientific Reports*, 6, August, p. 34841.

Biran, Hadas, Martin Kupiec, and Roded Sharan

2019 "Comparative analysis of normalization methods for network propagation", *Frontiers in genetics*, 10, p. 4.

Cao, Mengfei, Christopher M Pietras, Xian Feng, Kathryn J Doroschak, Thomas Schaffner, Jisoo Park, Hao Zhang, Lenore J Cowen, and Benjamin J Hescott

2014 "New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence", *Bioinformatics*, 30, 12, pp. i219-i227.

Chatr-aryamontri, Andrew, Rose Oughtred, Lorrie Boucher, Jennifer Rust, Christie Chang, Nadine K Kolas, Lara O'Donnell, Sara Oster, Chandra Theesfeld, Adnane Sellam, et al.

2017 "The BioGRID interaction database: 2017 update", *Nucleic acids research*, 45, D1, pp. D369-D379.

Chiaretti, Sabina, Xiaochun Li, Robert Gentleman, Antonella Vitale, Marco Vignetti, Franco Mandelli, Jerome Ritz, and Robin Foa

2004 "Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival", *Blood*, 103, 7, pp. 2771-2778.

Cowen, Lenore, Trey Ideker, Benjamin J Raphael, and Roded Sharan

2017 "Network propagation: a universal amplifier of genetic associations", *Nature Reviews Genetics*, 18, 9, pp. 551-562.

Csardi, Gabor

2015 *igraphdata: A Collection of Network Data Sets for the 'igraph' Package*, R package version 1.0.1, https://CRAN.R-project.org/package=igraphdata.

Cun, Yupeng and Holger Fröhlich

2013 "Network and Data Integration for Biomarker Signature Discovery via Network Smoothed T-Statistics", *PLoS One*, 8, 9.

Dittrich, Marcus and Daniela Beisser

2010 *DLBCL: Diffuse large B-cell lymphoma expression data*, R package version 1.16.0, http://bionet.bioapps.biozentrum.uni-wuerzburg.de/.

Erten, Sinan, Gurkan Bebek, Rob M Ewing, and Mehmet Koyutürk

2011 "DADA: degree-aware algorithms for network-based disease gene prioritization", *BioData mining*, 4, 1, p. 19.

Guala, Dimitri and Erik LL Sonnhammer

2017 "A large-scale benchmark of gene prioritization methods", *Scientific Reports*, 7.

Hill, Abby, Scott Gleim, Florian Kiefer, Frederic Sigoillot, Joseph Loureiro, Jeremy Jenkins, and Melody K Morris

2019 "Benchmarking network algorithms for contextualizing genes of interest." *PLoS Computational Biology*, 15, 12.

Kanehisa, Minoru, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima

2017 "KEGG: new perspectives on genomes, pathways, diseases and drugs", *Nucleic acids research*, 45, D1, pp. D353-D361.

Lee, Insuk, U Martin Blom, Peggy I Wang, Jung Eun Shim, and Edward M Marcotte

2011 "Prioritizing candidate disease genes by network-based boosting of genome-wide association data", *Genome Research*, 21, 7, pp. 1109-1121.

Li, Xiaochun

2009 *ALL: A data package*, R package version 1.20.0.

Lopez-del Rio, Angela, Alfons Nonell-Canals, David Vidal, and Alexandre Perera-Lluna

2019 "Evaluation of cross-validation strategies in sequence-based binding prediction using Deep Learning", *Journal of chemical information and modeling*, 59, 4, pp. 1645-1657.

Mishra, Gopa R, M Suresh, K Kumaran, N Kannabiran, Shubha Suresh, P Bala, K Shivakumar, N Anuradha, Raghunath Reddy, T Madhan Raghavan, et al.

2006 "Human protein reference database—2006 update", *Nucleic acids research*, 34, suppl 1, pp. D411-D414.

Mitra, Koyel, Anne-Ruxandra Carvunis, Sanath Kumar Ramesh, and Trey Ideker

2013 "Integrative approaches for finding modular structure in biological networks", *Nat. Rev. Genet.*, 14, 10, pp. 719-732.

Mostafavi, Sara, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris

2008 "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function." *Genome Biology*, 9 Suppl 1, S4.

Oliver, Stephen

2000 "Guilt-by-association goes global", *Nature*, 403, February, pp. 601-603.

Paull, Evan O., Daniel E. Carlin, Mario Niepel, Peter K. Sorger, David Haussler, and Joshua M. Stuart

2013 "Discovering causal pathways linking genomic events to transcriptional states using Tied Diffusion Through Interacting Events (TieDIE)", *Bioinformatics*, 29, 21, pp. 2757-2764.

Picart-Armada, S, SJ Barrett, DR Willé, A Perera-Lluna, A Gutteridge, and BH Dessailly

2019 "Benchmarking network propagation methods for disease gene identification", *PLoS Comput Biol*, 15, 9, e1007276.

Picart-Armada, Sergio, Francesc Fernández-Albert, Maria Vinaixa, Miguel A Rodríguez, Suvi Aivio, Travis H Stracker, Oscar Yanes, and Alexandre Perera-Lluna

2017 "Null diffusion-based enrichment for metabolomics data", *PloS one*, 12, 12, e0189012.

Picart-Armada, Sergio, Wesley K Thompson, Alfonso Buil, and Alexandre Perera-Lluna

2017 "diffuStats: an R package to compute diffusion-based scores on biological networks", *Bioinformatics*, 34, 3, pp. 533-534.

Rajagopalan, Dilip and Pankaj Agarwal

2004 "Inferring pathways from gene lists using a literature-derived network of biological relationships", *Bioinformatics*, 21, 6, pp. 788-793.

Rosenwald, Andreas, George Wright, Wing C Chan, Joseph M Connors, Elias Campo, Richard I Fisher, Randy D Gascoyne, H Konrad Muller-Hermelink, Erlend B Smeland, Jena M Giltnane, et al.

2002 "The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma", *New England Journal of Medicine*, 346, 25, pp. 1937-1947.

Saito, Takaya and Marc Rehmsmeier

2017 "Precrec: fast and accurate precision–recall and ROC curve calculations in R", *Bioinformatics*, 33, 1, pp. 145-147.

Sharan, Roded, Igor Ulitsky, and Ron Shamir

2007 "Network-based prediction of protein function", *Molecular systems biology*, 3, 1, p. 88.

Smola, Alexander J and Risi Kondor

2003 "Kernels and regularization on graphs", in *Learning theory and kernel machines*, Springer, pp. 144-158.

Valentini, Giorgio, Alberto Paccanaro, Horacio Caniza, Alfonso E. Romero, and Matteo Re

2014 "An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods", *Artificial Intelligence in Medicine*, 61, 2, pp. 63-78.

Vandin, Fabio, Eli Upfal, and Benjamin J. Raphael

2010 "Algorithms for detecting significantly mutated pathways in cancer", *Lect. Notes Comput. Sci.*, 6044 LNBI, 3, pp. 506-521.

Von Mering, Christian, Roland Krause, Berend Snel, Michael Cornell, Stephen G Oliver, Stanley Fields, and Peer Bork

2002 "Comparative assessment of large-scale data sets of protein–protein interactions", *Nature*, 417, 6887, p. 399.

Zoidi, Olga, Eftychia Fotiadou, Nikos Nikolaidis, and Ioannis Pitas

2015 "Graph-Based Label Propagation in Digital Media: A Review", *ACM Computing Surveys*, 47, 3, 48:1-48:35.

5 THE R PACKAGE DIFFUSTATS

DIFFUSTATS: AN R PACKAGE TO COMPUTE DIFFUSION-BASED SCORES ON BIOLOGICAL NETWORKS

Label propagation and diffusion over biological networks are a common mathematical formalism in computational biology for giving context to molecular entities and prioritising novel candidates in the area of study. There are several choices in conceiving the diffusion process (involving the graph kernel, the score definitions and the presence of a posterior statistical normalisation) which have an impact on the results.

This manuscript describes diffuStats, an R package that provides a collection of graph kernels and diffusion scores, as well as a parallel permutation analysis for the normalised scores, that eases the computation of the scores and their benchmarking for an optimal choice. The R package diffuStats is publicly available in Bioconductor, `https://bioconductor.org`, under the GPL-3 license.

5.1 INTRODUCTION

Network analysis can help finding therapeutic targets and understanding biology in networks obtained from protein-protein interactions, gene regulation and metabolic reactions. In this context, label propagation and diffusion algorithms (Zoidi et al., 2015) address a general problem of molecular entity ranking according to a seed node list. Examples include finding significantly mutated subnetworks in cancer (Vandin et al., 2010), predicting gene function (Mostafavi et al., 2008), prioritising genome-wide association hits (Lee et al., 2011) and classifying proteins (Tsuda et al., 2005).

In general, the mentioned methods involve diffusion processes with ad-hoc parameter and network settings, making comparisons fundamentally difficult. The RANKS R package (Valentini et al., 2016) is an effort to collect a range of diffusion kernels and scores, but the effects of label codification and a recently proposed statistical normalisation (Bersanelli et al., 2016) have not been explored. To that end, we introduce the diffuStats R package gathering diffusion kernels, input codifications and statistical normalisations to benchmark single-network diffusion settings.

This chapter is a postprint of the following journal article: Picart-Armada, Sergio, Wesley K. Thompson, Alfonso Buil, and Alexandre Perera-Lluna. "diffuStats: an R package to compute diffusion-based scores on biological networks". *Bioinformatics 34, no. 3 (2018): 533-534.*

Table 5: Implemented diffusion scores. f_{raw}, f_{ml} and f_{gm} differ on the weights of the positive, negative and unlabelled nodes. f_{ber_s} quantifies the change of the f_{raw} scores relative to the input scores. f_{ber_p}, f_{mc} and f_z are statistically normalised by permuting the labelled examples, but not the unlabelled*. f_{mc} derives from an empirical p-value, whereas f_{ber_p} combines f_{mc} and f_{raw}. f_z is a parametric alternative to f_{mc} requiring no stochastic permutations. Quantitative inputs are allowed in all the scores except f_{ml} and f_{gm}.

Score	y^+	y^-	y^u	Normalised	Stochastic	Quantitative	Reference
raw	1	0	0	No	No	Yes	(Vandin et al., 2010)
ml	1	-1	0	No	No	No	(Tsuda et al., 2005)
gm	1	-1	k	No	No	No	(Mostafavi et al., 2008)
ber_s	1	0	0	No	No	Yes	(Bersanelli et al., 2016)
ber_p	1	0	0*	Yes	Yes	Yes	(Bersanelli et al., 2016)
mc	1	0	0*	Yes	Yes	Yes	(Bersanelli et al., 2016)
z	1	0	0*	Yes	No	Yes	(Harchaoui et al., 2013)

5.2 METHODS

The diffuStats R package offers scoring schemes for diffusing a label vector on a network, determined by (a) the graph kernel, (b) the translation of labels into a numeric vector y to be smoothed, and (c) the statistical normalisation. In general, diffusion scores f are based on modifications of the quantity $f = K \cdot y$, where y are the input labels, f the diffusion scores and K is a graph kernel.

Regarding (a), most of the cited applications use the regularised Laplacian kernel, but our package also offers the diffusion kernel, the p-step random walk kernel, the inverse cosine kernel (Smola and Kondor, 2003) and the commute time kernel (Yen et al., 2007). In practice, they differ on the reach and the behaviour of the spreading inside the network - further detail for its choice can be found in the documentation. The decision can be data-driven, based on prior studies on the subject or on desirable properties of the kernel. For (b) and (c), the implemented scores are variations of the propagation of a binary vector whose ones are the positive labels y^+ and whose zeroes are the negative y^- and unlabelled y^u entities (Table 5). The statistical normalisation (c) compares the diffusion scores with the distribution of scores that arise from a permuted input, in order to spot nodes whose score is systematically high or low regardless of the input. However, not all normalised scores require stochastic simulations.

The diffuStats R package contains proper documentation and unit testing to facilitate its development. Its algorithms are written in R except the permutations, which use C++; further details on the implementation can be found in the supplementary materials. Manipulating networks with more than 10,000 nodes might require extra RAM memory and computational power due to the kernel matrix size.

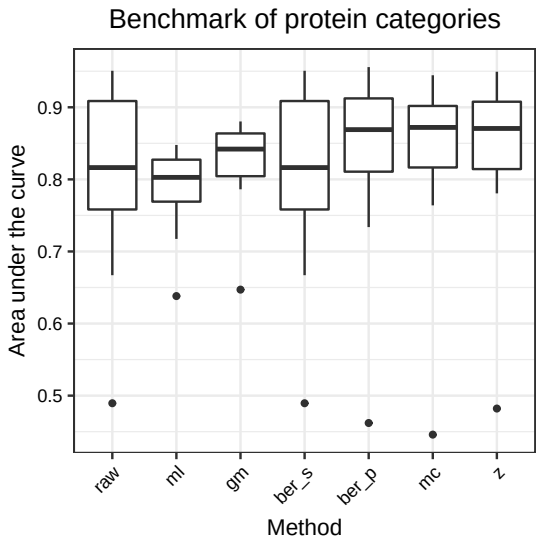


Figure 15: Performance comparison for diffusion scores in predicting 12 functions on half of the proteins using the area under the Receiver Operating Characteristic curve.

5.3 RESULTS

The example data is a yeast interactome with 12 annotated protein functions (Von Mering et al., 2002). The functionalities of our package are demonstrated by (i) obtaining a prioritised list of annotations given a set of labels, and (ii) benchmarking all the available diffusion scores in a dataset containing validation data. For both analyses, half of the proteins in the interactome will be treated as unlabelled and will receive a score from the propagation of the other half using the default regularised Laplacian kernel. Regarding (i), the function "diffuse" allows to compute the desired diffusion scores with a starting set of scores (labels) and a network:

```
scores_diff <- diffuse(graph = yeast,
  scores = scores, method = "raw")
```

When assessing the performance of different diffusion scores in a given dataset (ii), the desired metrics involving the diffusion scores and the validation can be computed on a grid of parameters:

```
performance <- perf(graph = yeast, scores = scores,
  validation = validation, grid_param = grid_param)
```

The results are returned as a table that can be directly plotted (Fig. 15). In this case study, statistically normalised scores f_{mc}, f_z and f_{ber_p} seem preferable than their unnormalised counterparts comparing the area under the ROC curves. For instance, f_z outperforms f_{raw}, f_{ml}, f_{gm} and f_{ber_s} (FDR < 25%, Wilcoxon test), thus highlighting the usefulness of a prior screening in score performance and its potential impact.

In summary, the R package diffuStats gathers diffusion kernels and scores with statistical normalisation that are object of active research in bioinformatics, like functional prediction or module identification. In addition, it

facilitates benchmarking diffusion scoring methods to find the optimal configuration for the application of interest.

FUNDING:

This work was supported by the Spanish Ministry of Economy and Competitiveness (MINECO) [TEC2014-60337-R to A.P.] and the National Institutes of Health (NIH) [R01GM104400 to W.T.]. AP. and S.P. thank CIBER-DEM and CIBER-BBN for funding, both initiatives of the Spanish ISCIII. SP. thanks the AGAUR FI-scholarship programme.

Conflict of Interest: none declared

REFERENCES

Bersanelli, Matteo, Ettore Mosca, Daniel Remondini, Gastone Castellani, and Luciano Milanesi

2016 "Network diffusion-based analysis of high-throughput data for the detection of differentially enriched modules", *Sci. Rep.*, 6.

Harchaoui, Zaid, Francis Bach, Olivier Cappe, and Eric Moulines

2013 "Kernel-based methods for hypothesis testing: A unified view", *IEEE Signal Process Mag*, 30, 4, pp. 87-97.

Lee, Insuk, U Martin Blom, Peggy I Wang, Jung Eun Shim, and Edward M Marcotte

2011 "Prioritizing candidate disease genes by network-based boosting of genome-wide association data", *Genome Res.*, 21, 7, pp. 1109-1121.

Mostafavi, Sara, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris

2008 "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function", *Genome Biol.*, 9, 1, S4.

Smola, Aj Alexander J and Risi Kondor

2003 "Kernels and Regularization on Graphs", *Mach. Learn.*, 2777, pp. 1-15.

Tsuda, Koji, HyunJung J. Shin, and Bernhard Schölkopf

2005 "Fast protein classification with multiple networks", *Bioinformatics*, 21, SUPPL. 2, pp. 59-65.

Valentini, Giorgio, Giuliano Armano, Marco Frasca, Jianyi Lin, Marco Mesiti, and Matteo Re

2016 "RANKS: A flexible tool for node label ranking and classification in biological networks", *Bioinformatics*, 32, 18, pp. 2872-2874.

Vandin, Fabio, Eli Upfal, and Benjamin J. Raphael

2010 "Algorithms for detecting significantly mutated pathways in cancer", *Lect. Notes Comput. Sci.*, 6044 LNBI, 3, pp. 506-521.

Von Mering, Christian, Roland Krause, Berend Snel, Michael Cornell, Stephen G Oliver, Stanley Fields, and Peer Bork

2002 "Comparative assessment of large-scale data sets of protein–protein interactions", *Nature*, 417, 6887, pp. 399-403.

Yen, Luh, Francois Fouss, Christine Decaestecker, Pascal Francq, and Marco Saerens

2007 "Graph nodes clustering based on the commute-time kernel", *Advances in Knowledge Discovery and Data Mining*, pp. 1037-1045.

Zoidi, Olga, Eftychia Fotiadou, Nikos Nikolaidis, and Ioannis Pitas

2015 "Graph-Based Label Propagation in Digital Media: A Review", *ACM Comput. Surv.*, 47, 3, 48:1-48:35.

6 METABOLOMICS ENRICHMENT

NULL DIFFUSION-BASED ENRICHMENT FOR METABOLOMICS DATA

Metabolomics experiments identify metabolites whose abundance varies as the conditions under study change. Pathway enrichment tools help in the identification of key metabolic processes and in building a plausible biological explanation for these variations. Although several methods are available for pathway enrichment using experimental evidence, metabolomics does not yet have a comprehensive overview in a network layout at multiple molecular levels. We propose a novel pathway enrichment procedure for analysing summary metabolomics data based on sub-network analysis in a graph representation of a reference database. Relevant entries are extracted from the database according to statistical measures over a null diffusive process that accounts for network topology and pathway crosstalk. Entries are reported as a sub-pathway network, including not only pathways, but also modules, enzymes, reactions and possibly other compound candidates for further analyses. This provides a richer biological context, suitable for generating new study hypotheses and potential enzymatic targets. Using this method, we report results from cells depleted for an uncharacterised mitochondrial gene using GC and LC-MS data and employing KEGG as a knowledge base. Partial validation is provided with NMR-based tracking of ^{13}C glucose labelling of these cells.

6.1 INTRODUCTION

Metabolomics is the science that studies the chemical reactions taking place in a living organism by measuring their lightweight reactants and products, also called metabolites. Metabolomics is used in the study of human disease, biomarker identification, drug evaluation and treatment prognosis (Nicholson et al., 2002). Metabolomics datasets are generated from the identification and quantification of the metabolites in a sample. Afterwards, statistical analysis of the datasets enables researchers to devise a plausible explanation for the changes identified and to understand the underlying biological processes involved (Chagoyen and Pazos, 2013).

Current methods to measure metabolites mainly rely on Nuclear Magnetic Resonance (NMR) and Mass Spectrometry (MS) technologies (Weckwerth,

This chapter is a postprint of the following journal article: Picart-Armada, Sergio, Francesc Fernández-Albert, Maria Vinaixa, Miguel A. Rodríguez, Suvi Aivio, Travis H. Stracker, Oscar Yanes, and Alexandre Perera-Lluna. "Null diffusion-based enrichment for metabolomics data". *PloS one 12, no. 12 (2017)*.

2003), the latter consisting of two broad categories: Liquid Chromatography and Gas Chromatography coupled to MS (LC/MS and GC/MS). Raw data processing, also known as primary analysis, can be achieved using tools including MeltDB(Kessler et al., 2013), MetaboAnalyst (Xia, Sinelnikov, et al., 2015), MAIT (Fernández-Albert et al., 2014), along with spectral databases (Vinaixa et al., 2015) like the Human Metabolome Database (Wishart et al., 2013), resulting in a table of relative metabolite abundances.

Data interpretation, known as secondary analysis, benefits from the identification of metabolic pathways to draw conclusions, encouraging the use of so-called pathway enrichment techniques. Their purpose is to provide the metabolites with their biological context, drawing from comprehensive databases like Kyoto Encyclopedia of Genes and Genomes, KEGG (Kanehisa et al., 2008), Reactome (Croft et al., 2014), WikiPathways (Kelder et al., 2012) and the Small Molecule Pathway Database (Kelder et al., 2012). Enrichment outputs can be further analysed by manual network manipulation through tools such as Cytoscape (Smoot et al., 2011), whose plug-in MetScape (Karnovsky et al., 2012) builds networks containing compounds, reactions, enzymes and genes. In this work, pathway enrichment techniques will be divided into three generations, following the review in (Khatri et al., 2012).

The first generation of enrichment techniques is based on Over Representation Analysis (ORA), a statistical test that assesses whether the occurrence of a label within a subset is greater than expected by chance in the background population. Applied to metabolomics, it takes as input the identifiers of affected metabolites (previously determined through a statistical test involving conditions) and assesses a p-value for each pathway. ORA is available through the web tools IMPaLA (Kamburov et al., 2011), MetaboAnalyst, MBRole and MPEA (Chagoyen and Pazos, 2011; Kankainen et al., 2011). Limitations of ORA include an oversimplification of the biology, a thresholding decision issue when generating the input metabolite list and a lower power for capturing subtle and coordinated changes within a pathway (Subramanian et al., 2005).

A second generation of enrichment methods, Functional Class Scoring (FCS), avoids the cutoff choice in generating the affected metabolite list and claims the capability of capturing subtle but consistent changes in concentration (Alonso et al., 2015; Chagoyen and Pazos, 2013). This concept was imported from Gene Set Enrichment Analysis (Subramanian et al., 2005) and is available through MSEA (Xia and Wishart, 2010) in MetaboAnalyst and IMPaLA. A shortcoming of FCS methods is that they ignore the network nature of biological pathways (Khatri et al., 2012). As biological datasets are heterogeneous, and as no method is always best, the researcher's expertise and prior knowledge remain key factors when choosing between ORA and FCS (Huang et al., 2009).

The third generation of enrichment techniques attempts to incorporate topological data on the underlying biological networks. This concept was applied early to genetic data through ScorePAGE (Rahnenführer et al., 2004) and is available in current tools like Pathway-Express (Draghici et al., 2007). For metabolomics data, MetaboAnalyst assigns each metabolic pathway a topological score accounting for the centrality of measured metabolites.

Pathway enrichment techniques face challenges, such as dealing with pathway crosstalk and overlap (Khatri et al., 2012) or generating comprehensive outputs rather than pathway p-value lists (Huang et al., 2009). Statistical tests that account for pathway crosstalk and overlap have been proposed for gene data (Donato et al., 2013; Tarca et al., 2012). Although pathway analysis techniques constitute essential resources for metabolomics secondary analysis, the abstract and artificial borders between pathways may not faithfully reflect biological mechanisms (Chagoyen and Pazos, 2013). This issue can be bypassed using sub-network analysis, a secondary analysis procedure to infer relevant biological modules under the condition of study (Mitra et al., 2013) without being limited by pathway definitions. Sub-network analysis has also been applied to the canonical pathways to obtain enrichment in a sub-pathway scale for gene and protein data (Haynes et al., 2013; Li et al., 2015). Some methods, such as jActiveModules (Ideker et al., 2002), define scores and attempt to find optimally scoring sub-networks. Likewise, diffusion kernels and random walk algorithms that score the nodes of a network, such as PageRank (Page et al., 1999), have been applied to genetic data (Paull et al., 2013; Vandin et al., 2011) and metabolic networks (Faust et al., 2010).

The HotNet algorithm (Vandin et al., 2011), applied to gene networks, computes pairwise influence measures from node g_s to node g_i, by introducing a flow on g_s and allowing it to leave through all the nodes. The diffusion score of node g_i, f_i^s, is interpreted as the influence $i(g_s, g_i)$. A new undirected graph is built using the weights $w(g_j, g_k) = \min[i(g_j, g_k), i(g_k, g_j)]$, in which sub-networks encompassing a large number of gene mutations are sought. TieDIE (Paull et al., 2013) applies a similar concept, aiming to connect a source and a target gene set. Flow is introduced between the source and the target sets, giving rise to two diffusion processes that score all the nodes. The linking score of each node, defined as the minimum of its two diffusion scores, serves as a ranking to apply a global threshold and report the resulting sub-network.

Here we describe the development of an innovative methodology that combines the usefulness of pathway enrichment with the flexibility of sub-network analysis. Starting from summary metabolomics data, we apply a null diffusive process over a network-based representation of the KEGG database and derive a relevant sub-network. Besides offering an overview in the form of a list of affected pathways, we propose a novel sub-pathway representation at several molecular levels that justifies the reported pathways through additional biological entities (reactions, enzymes and KEGG modules) to identify candidates for further study. All of the reported entries, along with their annotations, are drawn in a heterogeneous network layout.

6.2 MATERIALS AND METHODS

6.2.1 Overview

An overall scheme of the proposed methodology is presented (Fig 16): on the one hand, we retrieve knowledge from KEGG as a graph object; on the other hand, the input to our algorithm is a list of significantly affected

metabolites from an experimental study, obtained for example by applying a non-parametric Wilcoxon test to each metabolite's abundance. Afterwards, the graph is regarded as a meshed object in which the nodes representing the affected metabolites introduce unitary flow. The resulting node scores are normalised using a null diffusive model, and the top scores define an interpretable relevant subgraph. All this work has been implemented in the R language (R Core Team, 2015) and the network algorithms rely on the igraph R package (Csardi and Nepusz, 2006). Our R code is under active development and available at `https://github.com/b2slab/FELLA`.

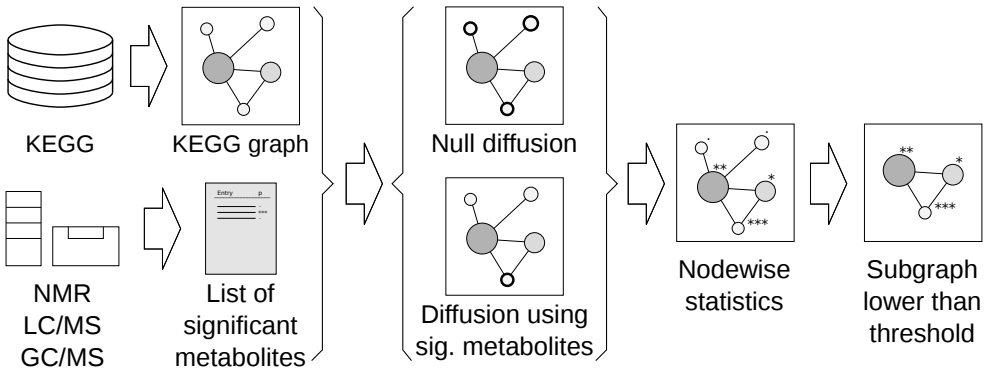


Figure 16: Workflow summary. Contextual knowledge is extracted from KEGG as a graph object while experimental data is introduced as a list of affected metabolites. A null diffusive model assesses, and reports in a subgraph, which part of the KEGG graph is relevant for the input metabolites.

Contextual knowledge is depicted according to the KEGG database (Fig 16), through the following categories: compounds, reactions, enzymes, modules and pathways. This network is specific for Homo sapiens and its construction is detailed in S1 Appendix.

6.2.2 Scoring algorithms

We derived scores for all the nodes through random walks on the KEGG graph, in order to assess their importance relative to the metabolites in the input. Performing random walks on the undirected graph is equivalent to running a diffusion process; specifically, we model heat diffusion. Conversely, if the graph is directed, the problem matches the PageRank algorithm for website ranking. Both the undirected and the directed versions are applied and referred to as diffusive processes (Fig 16).

In the undirected graph case, using a heat diffusion model, we model the biological perturbation in the KEGG graph as heat flow that traverses our KEGG graph. It is important to emphasise that this heat diffusion approach is purely a knowledge propagation abstraction, in no way simulating heat diffusion on the actual biological entities. Heat is forced to flow from nodes corresponding to affected metabolites and through database annotations, leading to a score for each node in the KEGG graph: its stationary temperature (Eq. 36). The rationale behind this approach is that nodes lying close to the affected metabolites, which are heat sources, will hold a higher stationary temperature. This can happen due to great proximity to

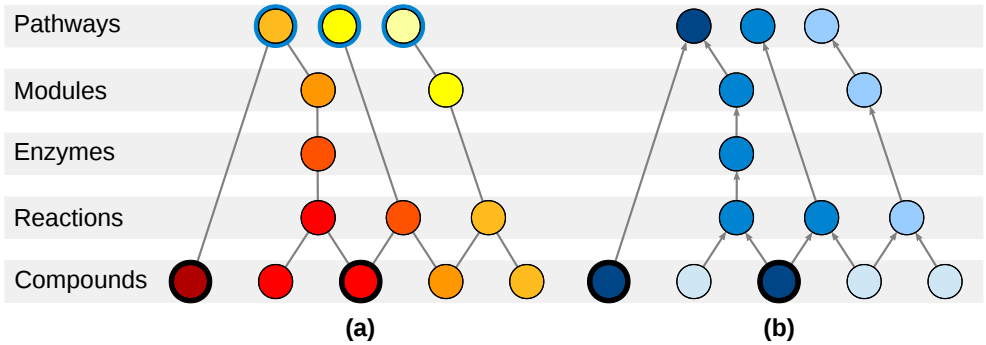


Figure 17: Nodes arrangement for **(a)** heat diffusion and **(b)** PageRank. The affected metabolites are highlighted with a black ring. For heat diffusion **(a)**, affected metabolites are forced to generate unitary flow. Every pathway is highlighted with a blue ring, representing its connection to a cool boundary node. In equilibrium, the highest temperature pathways (and nodes) will have the greatest heat flow, suggesting a relevant role in the experiment. For PageRank **(b)**, affected metabolites are the start of random walks. PageRank scores, represented by the intensity of the blue colour, will attain higher values in the frequently reached random walk nodes.

a particular heat source or to overall closeness to multiple ones. In order to determine the temperatures, we apply the finite difference formulation (Reddy and Gartling, 2010) of the heat equation, using the explicit method, applied to a meshed object (Fig 17a) (Bonals, 2005).

$$T = -KI^{-1} \cdot G = R_{HD} \cdot G \tag{36}$$

On the one hand, *KI* is the conductance matrix, where $KI = L + B$, L being the unnormalised graph Laplacian and B the diagonal adjacency matrix with $B_{ii} = 1$ if node *i* is a pathway and $B_{ii} = 0$ otherwise. The matrix B ensures that flow can leave the graph through pathways nodes. The matrix R_{HD} is defined as $-KI^{-1}$, the linear mapping to compute the temperatures. On the other hand, *G* is the heat generation vector, whose entries G_i are unitary if i is an affected metabolite and 0 otherwise.

In our node arrangement (Fig 17a), the affected metabolites constantly introduce heat flow into the structure and only the nodes in the top level (metabolic pathways) are allowed to disperse it. Further details are available in S2 Appendix.

In the directed graph case, the PageRank scoring algorithm is a web model that assigns each website a score reflecting the number of incoming hyperlinks as well as the quality of their respective websites. The web surfer performs random walks on a directed graph, with an initial probability distribution over the nodes. In each step, the surfer resumes his random walk with probability d and restarts it with probability $1 - d$, where d is the damping factor. If the surfer continues, he or she will choose an edge with a probability proportional to its weight. The default computation of PageRank scores is iterative for efficiency reasons, although a formula similar to (Eq. 36) can be derived and will be used in the proposed methods. The damping factor is set to $d = 0.85$ as in the original publication.

The arrangement of nodes for the PageRank calculation is identical to the one for diffusion (Fig 17b), being edges directed towards the upper levels. Random walks start only at the affected metabolites and explore all the reachable nodes. Further details are available in S3 Appendix.

6.2.3 Null models

The ranking of the network nodes is not achieved through raw scores, due to potential biases related to topological features. This is also the case in classical over-representation analysis, as it can be rephrased as a particular case of heat diffusion (Fig 18) where the observed statistic is the node temperature and its null distribution is the hypergeometric distribution. In view of this, our approach also includes a permutation analysis in the input, leading to a null distribution of scores for each node. Node scores are normalised using their null distributions and ranked, allowing a subgraph (Fig 16) to be extracted. Further details can be found in S4 Appendix.

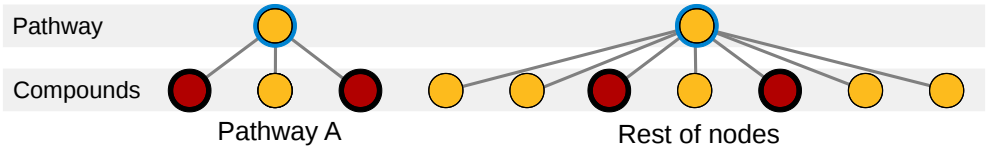


Figure 18: Toy example of an over-representation analysis of a hypothetical "pathway A" containing 3 metabolites out of a total of 10. The list to be enriched contains 4 metabolites, showing 2 hits in the pathway. The corresponding (Fisher's exact test) over-representation can be understood as a diffusion process on the depicted network followed by a null model. The temperature of pathway A is always coincident with the number of hits in the pathway, implying that its null distribution is the hypergeometric distribution, to which a one-tailed temperature comparison is made.

The null model will be introduced in the heat diffusion scenario (the PageRank case is analogous). Let n_{in} be the number of compounds in the input. Then, exactly n_{in} different KEGG compounds are chosen at random following dependent Bernoulli distributions, so that $X_i = 1$ if i is chosen and $X_i = 0$ otherwise. Normalisation can be performed using (i) the theoretical mean and variance of the scores, which can be obtained from Eq. 36, using the fact that, for the null model, G is a random vector X with known mean and covariance matrix:

$$\mathbb{E}(T_{null}) = R_{HD} \cdot \mathbb{E}(X) \tag{37}$$

$$\Sigma(T_{null}) = R_{HD} \cdot \Sigma(X) \cdot R_{HD}^{T} \tag{38}$$

The normalised score (z-score) of node i is defined in terms of the expected value $\mu_i = \mathbb{E}(T_{null})_i$ and standard deviation $\sigma_i = \sqrt{\Sigma(T_{null})_{i,i}}$

$$z_i = \frac{T_i - \mu_i}{\sigma_i} \tag{39}$$

Then, nodes with the top k scores are kept and reported. Alternatively, scores can be normalised through (ii) Monte Carlo simulations with n_{perm} permutations, which provide an estimate of the probability p_i that the null distribution attains a score greater than or equal to the observed one. Estimation of p_i involves the empirical cumulative distribution function with a small correction (North et al., 2002), r_i being the number of permutations in which the null score of node i is greater or equal than T_i:

$$p_i = \frac{r_i + 1}{n_{perm} + 1} \tag{40}$$

A consensus solution is derived from n_{vote} independent sets of Monte Carlo trials, each trial reporting the top k nodes. The consensus solution may therefore report a node count not exactly equal to k.

6.2.4 NMR validation

The reported subgraphs contain entities other than pathways and compounds that can be useful for the researchers. Among these, the highlighted reactions have been partially validated by quantifying their distance to an independent second set of affected metabolites.

In order to analyse the reactions in the scope of a metabolic network, distances are computed on the unweighted, maximal connected subgraph containing all the compounds and reactions from the KEGG graph, referred to as the reaction-compound graph. The validation metric is the resistance distance, previously used in the chemical literature (Bapat, 2004). Under these settings, the reported reactions are compared to all the reactions that involve the input metabolites (their nearest neighbours) in terms of their resistance distance to the second set of metabolites.

6.2.5 Evaluation with synthetic signals

In order to deploy an analysis of true and false positive pathway identifications, we opted to statistically characterize the pathway prioritisation induced by the diffusion scores. Artificial pathway signals have been generated to (a) find biases in the absence of a signal that might cause false positives, and to (b) quantify the ability to recover true positive pathways. The proposed methods are not directly compared to IMPaLA and MetaboAnalyst due to the lack of a batch analysis mode, but instead to their underlying distribution using Fisher's exact test. Our Monte Carlo approaches have not been aggregated into consensus solutions. The performance metric is the pathway rank in the list ordered by a method, where $\frac{1}{n_p}$ is the best rank and 1 is the worst one, n_p being the number of pathways in the KEGG graph. Ranks in Fisher's exact test are computed using the raw p-values, so that top ranked pathways correspond to lowest p-values. To compute the p-values, a metabolite is considered to belong to a pathway if it can be reached via the pathway in our directed KEGG graph (Fig 17).

In (a), noisy signals are generated and the ranks of all the pathways are calculated within signals. Then, the mean rank of a specific pathway i is

computed across all the signals. This measure can reveal pathways that tend to have an extreme rank irrespective of the input.

In (b), a target pathway generates the signal and its rank is used as the metric of interest. Methods able to recover the signal will show low ranks in general terms.

6.2.6 Description of the experimental data

Our method has been tested using data from a case-control experiment aimed at determining the function of an uncharacterised mitochondrial protein by silencing the gene using short hairpin RNAs (shRNA). Metabolites abundances were determined from five replicates of cell cultures expressing either control or experimental shRNA.

Metabolite measurements were performed by Metabolon platform (www.metabolon.com) using GC/MS (Thermo-Finnigan Trace DSQ single quadrupole) and LC/MS (Waters ACQUITY UPLC and a Thermo-Finnigan LTQ-FT). The proprietary Metabolon analysis reported 168 quantified metabolites annotated in the KEGG database.

In addition, we have used NMR following the labelling of the same cells with [U-^{13}C] glucose (DeBerardinis et al., 2007) to trace carbon atoms, in order to further validate the conclusions of our new method. The reported reactions are evaluated in terms of their resistance distance to the affected metabolites found by NMR.

6.2.7 Description of the synthetic data

All the signals generate a list with fixed length $n_{in} = 35$ for each one of the n_p pathway nodes in the KEGG graph. Three sampling types have been defined – differences arise in the specification of how much more probable compounds in the target pathway are.

The first signal is a uniform sampling of n_{in} compounds that imitates noise: the probability of drawing a compound j within pathway i, $p_{i,j}$, is $k_i = 1$ times more likely to be drawn than compounds outside the pathway, and thus does not depend on the pathway.

In the second signal, compounds belonging to pathway i are $k_i = 10$ times more likely to be drawn. Therefore, there are two different probability values: inside pathway and outside pathway. This sampling is affine to the assumptions in Fisher's exact test from ORA.

As for the third signal, $p_{i,j}$ is proportional to the quantity $R_{HD\,ij}$, which is greater in compounds close to the pathway. This takes into account the whole KEGG graph, thus being influenced by indirect connections and compound specificity.

6.3 RESULTS

6.3.1 Input for the algorithms

After the curation step, our knowledge base graph contains 10,183 nodes and 31,539 edges. The nodes are stratified in 288 pathways, 178 modules, 1,149 enzymes, 4,699 reactions and 3,869 compounds. The degree distribution of its vertices follow a scale-free network model, where $P(k) \sim k^{-\gamma}$, with $\gamma = 2.084 \in [2, 3]$, see S1 Appendix.

On the other hand, MS led to 168 quantified metabolites from KEGG. Two identifiers that each appeared twice have been dropped, as well as a KEGG drug, excluded from the KEGG compound category. The remaining 163 metabolites have been tested between both conditions, leading to 38 significant metabolites (two-tailed non-parametric Wilcoxon, FDR < 0.05), of which 33 have been mapped to our KEGG graph.

The 33 MS-derived compounds served as input for each of the proposed enrichment algorithms. Heat Diffusion (*HD*) and PageRank (*PR*) are followed by *norm* (z-score normalisation) or *sim* (Monte Carlo permutations). Normalised scores have been computed through the null models with $n_{in} = 33$, followed with subgraph selection with a desired number of nodes $k = 250$. For simulated methods, a consensus subgraph using $n_{vote} = 9$ runs of $n_{perm} = 10,000$ permutations each has been derived by majority vote on each node.

Regardless of the specific details, high diffusion scores are an indicator of overall closeness to the MS-derived metabolites and potential relevance in the condition being studied. This intuition, known as guilt-by-association, can be phrased in the context of heat diffusion: high temperatures are found close to the heat sources. Therefore, warm nodes are candidates for further study as they are easily reached through database annotations from the input metabolites.

6.3.2 Null model impact

The impact of using the null model in HD and an overview of the random temperatures behaviour is described in Fig 19. The null model is closely related to the graph structure and node topology, quantified through the vertex degree. In Fig 19a, the mean temperatures show different trends for the five levels in the graph; in particular, there is an increase in the mean pathway temperature as the pathway becomes larger. This implies that, regardless of the input, larger pathways will generally show warmer temperatures and the results will be biased towards them. Likewise, the standard deviations of the null temperatures show level-specific changes (Fig 19b), with the compounds being the most affected entities – the higher the degree of the compound, the lower its standard deviation.

The usage of z-scores instead of raw temperatures has consequences in the highlighted nodes. Reporting the nodes with the top 250 raw temperatures does not reveal any pathway (Fig 19c), whereas five pathways lay among the top 250 z-scores (Fig 19d). Likewise, if only pathway nodes are considered, their ranking using raw temperatures is closely related to the ranking using

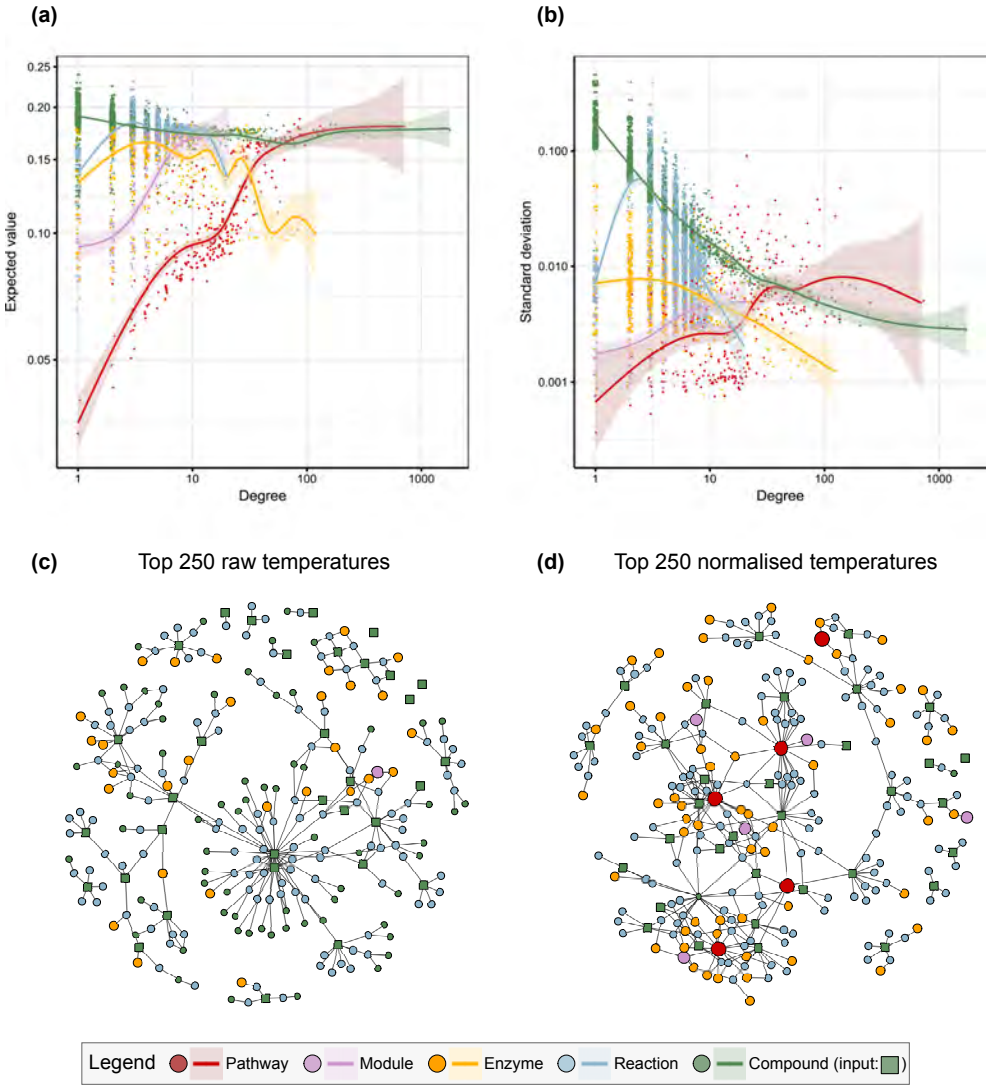


Figure 19: Expected value **(a)** and standard deviation **(b)** of the null temperatures, stratified by level – jitter applied for visual purposes and 0.95 confidence intervals computed by the default GAM models in ggplot2 R library (Wickham, 2009). Clear biases arise due to the node degree, a topological property of the nodes: the larger the pathway, the higher its mean value, and the more connected a compound is, the smaller its variance. If pathways are ranked by raw temperatures, a large pathway will have an undesired, consistent advantage over small ones and will be reported too often. The usage of z-scores **(d)** instead of raw temperatures **(c)** to select the top 250 nodes addresses these biases and highlights pathway and module nodes that were eclipsed by other compounds and reactions with higher mean null temperatures.

the mean temperatures from the null model (Fig 20a), which is a property of the graph but not of the experimental data; using z-scores instead corrects this bias (Fig 20b). If the top 20 pathways are selected through their raw temperature, some of them are even below their mean null temperature (Fig 20c), whereas keeping the top 20 z-scores removes the bias towards larger pathways and suggests otherwise overlooked pathways (Fig 20d).

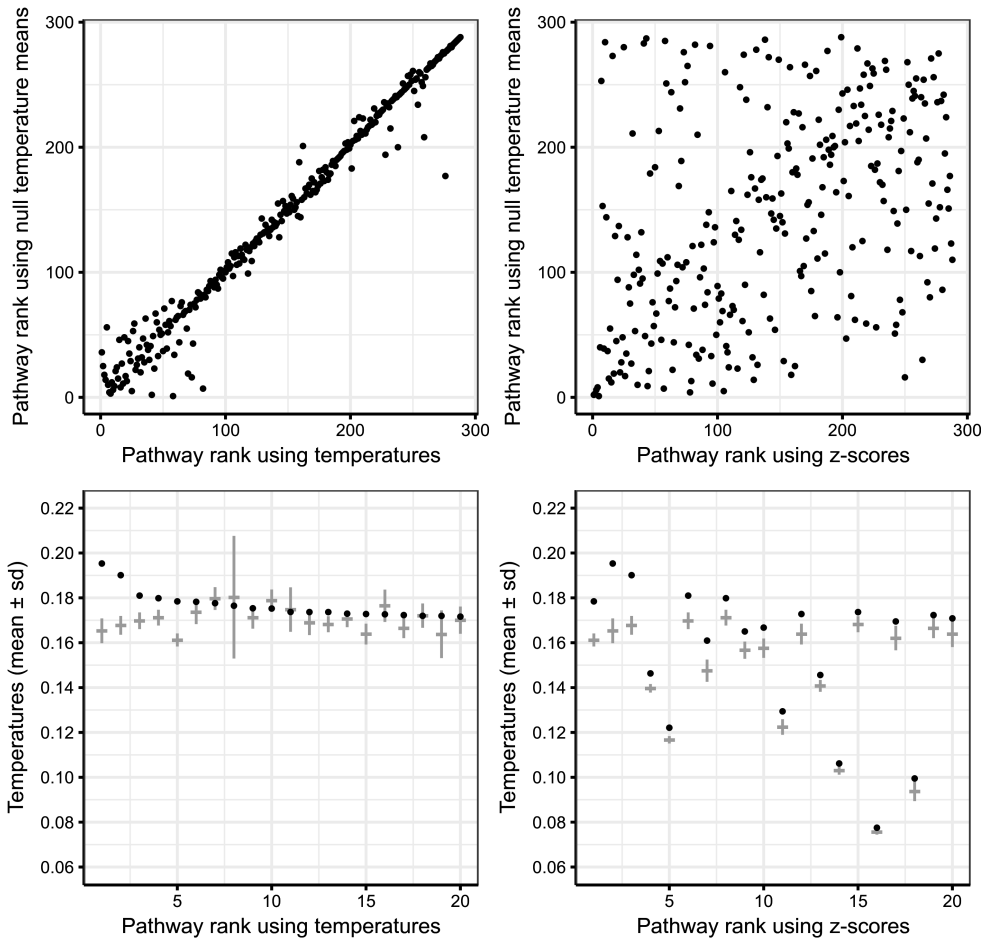


Figure 20: Ranking the 288 KEGG pathways – lower is best– using raw temperatures **(a)** biases the ranks towards pathways with higher mean null temperature, which in turn tend to be large pathways. Using the z-scores instead **(b)** breaks this clear dependence and avoids reporting pathways just because of their size. The top 20 pathways through raw temperatures **(c)**, depicted as black dots, include pathways that are even below their mean value, while the top 20 z-scores **(d)** suggest smaller pathways that were penalised by the aforementioned bias.

6.3.3 Subgraph extraction

Four subgraphs have been extracted using the MS-derived compounds. The desired number of nodes k for each approach, together with the actual number of reported nodes and the number of KEGG pathways, are shown in Table 6. A connected component (CC) of an undirected graph is a maximal connected subgraph so that any two nodes in the subgraph are connected by a path. For the directed graphs, the weak CC definition is used, in which directed edges are considered as undirected when computing the CC. The number of nodes belonging to each solution subgraph, along with its largest CC and the number of CCs, are also reported. Additional details regarding the largest CC and number of CCs for other values of k can be found in S5 Appendix.

Defining the overlap coefficient between two solutions G_1 and G_2 as

Table 6: Summary of the outputs

Name	k	Pathways	Nodes	#CC	Largest CC
HD norm	250	hsa00250, hsa00270, hsa00480, hsa05230, hsa05231	250	8	206
HD sim	250	hsa00250, hsa00270, hsa00330, hsa00480, hsa05230, hsa05231	261	8	221
PR norm	250	hsa00250, hsa00270, hsa00480, hsa05231	250	9	187
PR sim	250	hsa00250, hsa00270, hsa00480, hsa05231	279	10	152

Summary of the outputs, using diffusion (HD) as well as PageRank (PR), and normalising the scores with Monte Carlo simulations (sim) or z-scores (norm). Monte Carlo simulations have been run 10,000 times per solution, and 9 solutions have been computed to build a consensus solution. Note that the desired number of nodes k is slightly different to the number of nodes actually reported in the Monte Carlo simulations. The last two columns contain the number of connected components (CC) and the number of nodes in the largest CC.

$$\text{overlap}(G_1, G_2) = \frac{|G_1 \cap G_2|}{\min(|G_1|, |G_2|)}, \tag{41}$$

solutions tend to overlap despite their differences (Table 7). Regarding the stratification of the subgraphs in terms of KEGG categories, they follow a trend similar to the KEGG graph (S5 Appendix).

Table 7: Solutions overlap

	HD norm	HD sim	PR norm	PR sim
HD norm	1.00	0.82	0.88	0.82
HD sim	0.82	1.00	0.77	0.83
PR norm	0.88	0.77	1.00	0.84
PR sim	0.82	0.83	0.84	1.00

Overlap coefficient statistics for HD and PR. The overlapping nature of solutions is a sign of consistency among approaches.

6.3.4 Pathway analysis

Our methods are compared to IMPaLA and MetaboAnalyst to verify the concordance in terms of metabolic pathways. All the approaches have been compared using the example data from IMPaLA (S2 Table) and MetaboAnalyst (S3 Table), and they show consistent and compatible reports.

The results for our dataset are summarised in Table 8 and described in S1 Table, together with further details about the reports of the alternative tools. The metabolic pathways Alanine, aspartate and glutamate metabolism (hsa00250), Cysteine and methionine metabolism (hsa00270) and especially the Glutathione metabolism (hsa00480) recur in all of the approaches. Some of our solutions are more specific, suggesting the module Glutathione Biosynthesis (M00118) as well. Our null model takes pathway overlap and crosstalk into account and allows a visualisation of the pathway structure through the null diffusion correlation matrix (S4 Appendix).

The subgraph resulting from applying HD sim (Fig 21) inherits the scale-free structure from the whole graph and enrols the three recurrently reported pathways in the same connected component: hsa00250, hsa00270 and hsa00480. The biological perturbation stemming from the MS-derived com-

Table 8: Reported pathways

KEGG id	Pathway name	HD norm	HD sim	PR norm	PR sim	MA FCS	MA ORA	IMPaLA ORA
hsa00250	Alanine, aspartate and glutamate metabolism	+	+	+	+	+	+	-
hsa00270	Cysteine and methionine metabolism	+	+	+	+	+	+	+
hsa00480	Glutathione metabolism	+	+	+	+	+	+	+
hsa05230 (hsa00970)	Central carbon metabolism in cancer	+	+	-	-	*	-	+
hsa05231 (hsa00564)	Choline metabolism in cancer	+	+	+	+	*	-	-
hsa00260 (M00020)	Glycine, serine and threonine metabolism	*	*	-	-	+	-	-
hsa00330 (M00133)	Arginine and proline metabolism	*	+	-	-	+	-	+
hsa00510 (M00073)	N-Glycan biosynthesis	-	-	*	*	-	-	-

Pathways reported by our methods. '+' means a hit for the term reported in the KEGG id column, '*' stands for a hit of the closely related term in parenthesis in the same column and '-' states no hit. Our 4 solutions are compared to MetaboAnalyst (MA), using ORA and FCS, and IMPaLA using ORA. Pathways hsa00250, hsa00270 and hsa00480 are repeatedly reported by all the methodologies. Pathways hsa05230 and hsa05231 are reported by some of our methods, while alternative approaches find some close (*) and exact (+) matches. In some cases, instead of reporting a whole pathway, only specific modules within it are reported as relevant; this is the case of M00133 and M00073. Furthermore, module M00073 does not contain any compounds, being out of the scope of MetaboAnalyst and IMPaLA, but is reported by one of our methods due to the presence of other indirect relationships through enzymes in the graph.

pounds can be tracked in terms of reactions, enzymes and modules, up to the relevant pathways.

On the other hand, results on the recovery of synthetic signals can be found in Fig 22. In (a) absence of signal, HD ranks pathways with a mean rank close to 0.5, and only a few are biased to the top or the bottom of the list. Mean ranks in Fisher's exact test and PR are also centered around 0.5, but have more dispersion. In (b) the presence of a target pathway, three sampling schemes have been explored. In (1) the signal is actually noise and the target pathway is a decoy. The rank of the target pathway for HD and PR is uniformly spread in $[0, 1]$, whereas Fisher's exact test shows some asymmetry in the rank distribution. In (2), the sampling probability depends on the presence or absence of the metabolite in the pathway. Fisher's exact test outperforms HD and PR as the median rank of the target pathway is closer to 0, as expected by its optimality. However, in (3), the sampling probability is network-based and HD outperforms PR, which in turn outperforms Fisher's exact test. Differences between sim (Monte Carlo trials) and norm (parametric approach) are subtle.

6.3.5 NMR analysis

NMR carbon tracking revealed 13 isotopically enriched metabolites from ^{13}C-glucose, showing differential fractional enrichment between case-control, of which 5 had already been found through MS; some of these metabolites can be seen in Fig 23 in the context of the Glutathione metabolism. Our solutions are assessed in terms of the resistance distance from the reported reactions to the remaining 8 metabolites. The smaller the overall distance of a solution, the more related its nodes are to the 8 metabolites proven affected by NMR. The resistance distances have been computed on the reaction-compound graph, which is the largest CC of the subgraph that contains all the reactions and compounds in the KEGG graph.

The reactions suggested in our subgraphs show lower resistance distances to the 8 NMR-derived metabolites than the totality of reactions in the reaction-compound graph (Table 9). Furthermore, they are also lower than the resis-

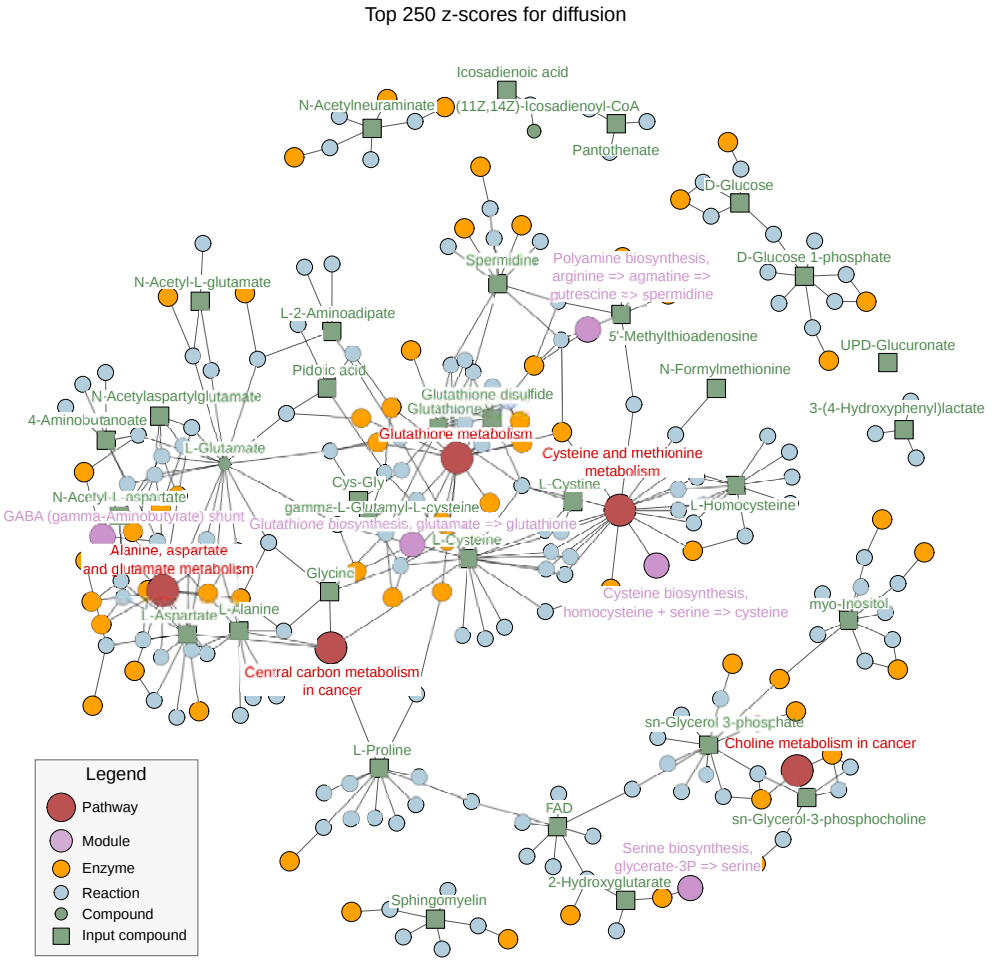


Figure 21: Subgraph reported through HD norm, the names of reactions and enzymes have been omitted for clarity. Compounds are green, reactions are blue, enzymes are orange, modules are purple and pathways are red. The compounds in the input are highlighted as green squares to ease the tracing of the biological perturbation up to the pathways. The presence of reactions and enzymes that link pathways in this subgraph might suggest relevant entities by which affected pathways crosstalk. All the reported pathways and modules lie in a large CC, as well as a newly proposed metabolite (L-Glutamate).

tance distances from the neighbouring reactions of the MS-derived metabolites to the 8 NMR metabolites (FDR < 0.01).

6.4 DISCUSSION

Our approach for enriching summary metabolomics data, Fig 16, is based on diffusion processes over a graph drawn from several KEGG categories (Fig 17). KEGG is the database of choice due to its level of curation and structure, which eases the graph representation. Specifically, the definition of KEGG categories naturally allows a hierarchical arrangement of levels. Our graph design is enhanced by the compound-reaction-enzyme-gene networks built by MetScape (S1 Appendix), and the inclusion of modules and

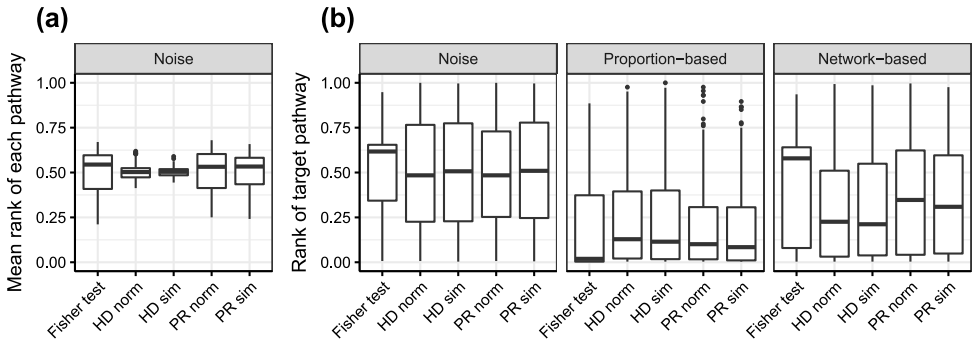


Figure 22: Synthetic signals evaluation using the pathway rank as a metric to assess orderings. Lowest ranks correspond to best ranked pathways. The proposed methodology is compared to ORA, represented by Fisher's exact test. **(a)** 288 noisy signals have been generated, and every pathway has been ranked in each of the 288 runs. Data points for a given methodology are the mean rank of each pathway, giving 288 data points per box. **(b)** 288 signals with a target pathway have been generated, in three scenarios: pure noise, proportion-based sampling and network-based sampling. Each box contains the rank of the target pathway, leading to 288 data points per box.

Table 9: Distance to NMR metabolites

Method	Graph order	C00299	C00122	C00116	C00105	C00020	C00581	C00300	C00025
Reaction-compound graph	4539[8008]	0.56(0.62)	0.56(0.62)	0.57(0.62)	0.54(0.62)	0.47(0.62)	0.93(0.62)	0.82(0.62)	0.47(0.62)
First neighbours	414[447]	0.42(0.12)	0.43(0.12)	0.44(0.12)	0.40(0.12)	0.33(0.12)	0.79(0.12)	0.68(0.12)	0.33(0.12)
HD norm	147[250]	0.39(0.10)	0.39(0.10)	0.40(0.10)	0.37(0.10)	0.30(0.10)	0.76(0.10)	0.65(0.10)	0.30(0.10)
HD sim	148[261]	0.39(0.09)	0.39(0.09)	0.40(0.10)	0.37(0.09)	0.30(0.09)	0.76(0.09)	0.65(0.09)	0.30(0.09)
PR norm	143[250]	0.39(0.10)	0.39(0.10)	0.40(0.10)	0.37(0.10)	0.30(0.10)	0.75(0.10)	0.65(0.10)	0.30(0.10)
PR sim	172[279]	0.40(0.12)	0.41(0.12)	0.42(0.12)	0.38(0.12)	0.31(0.12)	0.77(0.12)	0.66(0.12)	0.31(0.12)

Mean resistance distance between the reactions reported in our solutions and each compound reported using NMR, with their standard deviations in parentheses. For each subgraph of KEGG graph, the number of reactions and the total number of nodes (in square brackets) are displayed. The reaction-compound subgraph contains the largest connected component having all the reactions and compounds in the KEGG graph. The first neighbours subgraph contains the MS-derived metabolites and all the reactions in which they participate. Resistance distances are computed on the reaction-compound graph. For every NMR-derived metabolite, there is a significant difference in resistance distances between the reactions proposed in our solutions and the reactions involving any of the MS-derived metabolite (one-sided Wilcoxon test, FDR < 0.01 for the 32 possible comparisons: 8 NMR metabolites, tests of 4 solutions against the first neighbours reactions). This implies that the reported reactions are closer to the NMR-derived compounds than the bulk of neighbouring reactions.

pathways in our arrangement allows a comprehensive picture of the affected biology.

The graph contains all the KEGG compounds and the subset of affected metabolites forced to diffuse inside it (Fig 17). The closer a node is to the affected compounds, the higher its score becomes. Likewise, the top scoring candidates naturally involve higher flow and become relevant in the flow discharge from the graph. Because our KEGG graph is conceived and curated in a bottom-up manner, diffusion is expected to follow that trend too: the perturbation in the lowest level will diffuse to the upper levels to exit the graph. Ideally, a relevant subgraph found through this diffusion (Fig 21) would inherit the stratification of the KEGG graph, thus allowing the extrapolation of knowledge in terms of compounds to the rest of categories. This allows holistic picturing of pathways of interest, such as Glutathione

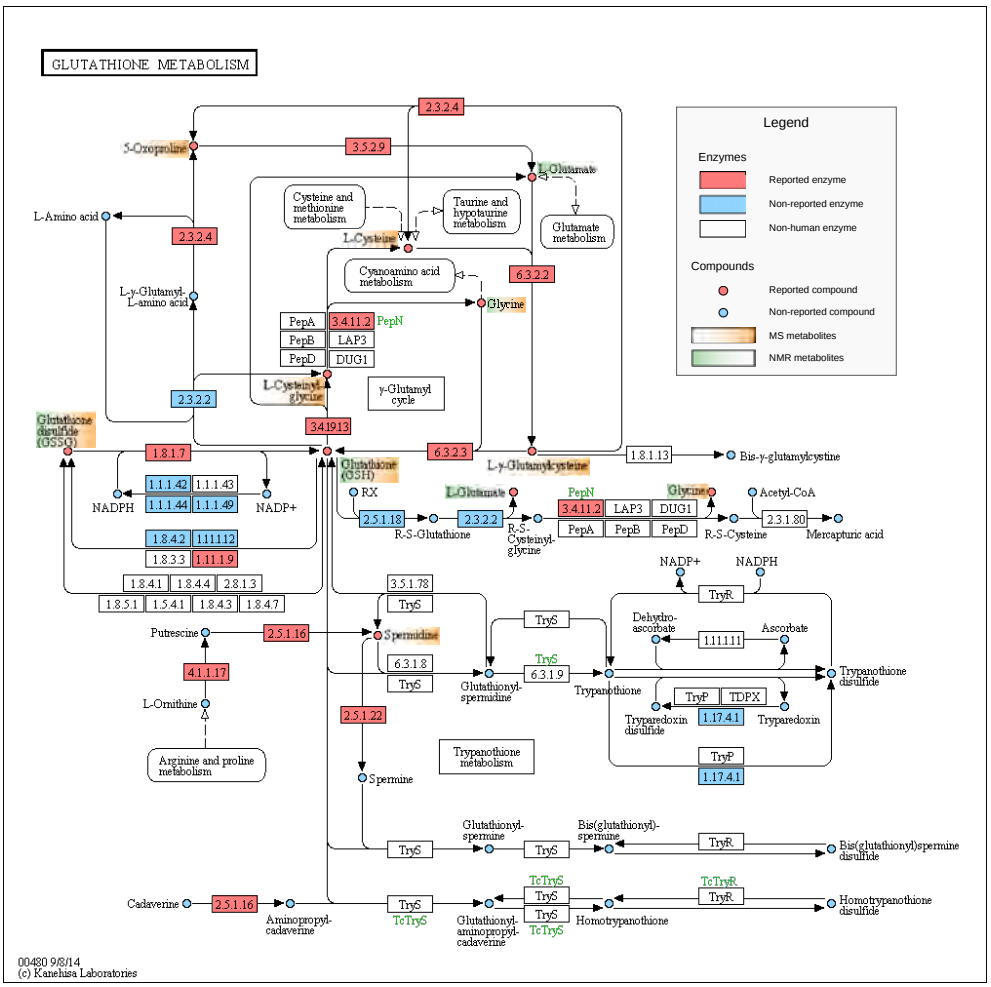


Figure 23: KEGG representation of the Glutathione metabolism (hsa00480). KEGG compounds found affected through MS (orange) and NMR (blue) are pinpointed in the figure. Additionally, enzymes and compounds reported by HD norm are depicted in red. Our approach provides a criterion for highlighting a pathway together with the entities it contains, for example its reported enzymes, to build a sub-pathway representation richer than the classical methods that rely solely on pathways and compounds. Reprinted from www.genome.jp under a CC BY license, with permission from Kanehisha Laboratories, original copyright 2014.

metabolism (Fig 23) and importantly, it relates affected pathways through reactions, enzymes and compounds.

The mathematical formulation of the heat diffusion stationary temperatures is equivalent to the scores in HotNet and TieDIE, with ad-hoc boundary conditions (Fig 17). Conversely, our settings for PageRank force upwards diffusion and allow exit from every node through the damping factor. Node selection for HotNet follows a combinatorial model, whereas TieDIE applies a unique threshold for all the scores, which in turn come from two diffusive processes. In our case, selection is achieved through a unique diffusion followed by a null model that normalises the scores. Comparing raw scores between nodes can lead to biases related to the node level and topology (Fig 19ab), pathway nodes clearly being affected by their degree and, in addition, overshadowed by other compounds and reactions with higher mean null

temperatures. Without further action, the temperatures of larger pathways are systematically warmer regardless of the input, thus biasing all the results and any biological interpretation. Instead, our concept of a high score for a given node relies on comparing its score to its null distribution, treating each node according to its own topological features (Fig 16).

This is consistent with the pathway over-representation analysis, as the latter can be posed as a very simple diffusion problem that needs the null model to translate the observed statistics into p-values that are comparable across pathways (Fig 18). Ranking pathways by the number of hits and ignoring the null model would bias the results towards larger pathways, which is also what happens in our diffusion approach if raw temperatures are used (Fig 20ab).

Finally, we extract four subgraphs by considering the top k scores for HD norm, HD sim, PR norm and PR sim. Spurious highlighted nodes are expected to appear as isolated or having very small CCs, similar to random selection of nodes in a sparse graph, whereas strong biological perturbations yield larger CCs. Therefore, the large CCs reported in the four subgraphs (Table 6) are natural goodness-of-solution indicators.

Analysing the two statistical approaches, we suggest both deterministic parametric techniques and stochastic non-parametric ones. Computing a z-score is simple and fast, giving insights into how high a score is in terms of standard deviations from the mean value. On the other hand, Monte Carlo trials can show some variability between solutions, so an ensemble approach can address this, while providing confidence measures for each reported node. Conversely, several quantiles can be estimated and stored if the graph is unchanged for further analyses, which is reasonable for a given KEGG database release.

Regarding time and memory complexity, the complete analysis of the database requires a one-off computation the inverse of the conductance matrix of the graph, which is feasible in our scenario and already pre-computed for our public package. The cost of the Monte Carlo trials is benchmarked in S5 Appendix. Comparing both random walk approaches, we observe a tendency to report larger CCs through heat diffusion (Table 6), because it can propose new compounds in the solution that connect otherwise disjoint CCs. This is not the case for PageRank, as forcing the diffusion upwards excludes other compounds from being visited by the random walks. As expected, all the approaches tend to report the metabolites that were specified in the input, although the z-scores can be more restrictive when suggesting new compounds in heat diffusion, possibly due to their high variance. Despite the differences between scoring methods and statistical approximations, solutions show a consistency because of their high overlap (Table 7). Furthermore, reporting subgraphs with a stratification similar to the KEGG graph (S5 Appendix) indicates perturbation traceability and allows inference on various KEGG categories by measuring only compounds.

As a pathway enrichment method, our procedure shows results consistent with the state of the art. Artificial signals have been generated to discover biases in particular pathways and assess the goodness of the rankings produced by the methods. In (a) the absence of signal, the mean rank of a pathway is expected to be uniform on $[0, 1]$ and have a mean value of 0.5. If

the mean value is closer to 0, the pathway might be systematically favoured in any analysis and could become a recurrent false positive. HD shows small deviations from 0.5 in the mean rank of the 288 pathways in the KEGG graph while PR and Fisher's exact test show more dispersion. This may be due to the discrete nature of Fisher's exact test, which is partly inherited by PR as it only allows upwards propagation. In (b) the presence of signal, a target pathway generates the signal and is ranked in the prioritisation of each method. In the first sampling scheme, the target pathway is actually a decoy and is expected to be ranked uniformly on $[0, 1]$. This is the case for HD and PR, but Fisher's exact test shows an asymmetrical distribution, probably a consequence of pathways tied at 0 hits. If the sampling strategy is affine to Fisher's exact test alternative hypothesis, this test has an edge over HD and PR in terms of discovering the true positive. Conversely, if the sampling is network-based, HD and PR perform better, as the binary nature of Fisher's exact test cannot account for metabolites close to, but not inside of, a target pathway. This sampling generates signals that are harder to recover because of the network topology: crosstalk effects are present and unspecific metabolites divide their contribution over all the pathways to which they belong. This implies that, focusing on the pathway ranking problem, the optimal choice between Fisher's exact test and HR or PR depends on the network influence in the generative model of the data.

An added value of our approach is in providing further details about the reported pathways, together with more specificity due to the presence of KEGG modules. Our results offer sub-pathway resolution and, unlike other sub-pathway focused tools, details at several molecular levels between the metabolites and the pathways. Entities like enzymes or metabolites that appear relevant and shared among pathways can give insights of pathway overlap and crosstalk that is specific to the condition under study. Our pathway hits are consistent with the current techniques, both using list format and abundance data (Table 8). The same tendency is observed when benchmarking with IMPaLA and MetaboAnalyst example data, details in Tables S2 and S3. However, the nature of our scores takes into account pathway overlap, which is not the case for IMPaLA (ORA) and MetaboAnalyst (ORA and MSEA).

Our prior studies (Aivio and Stracker, 2014) suggest that the Glutathione metabolism (Fig 23) is of particular interest and it is consistently pinpointed by the enrichment methods. Its study is illustrative of the workings of our methodology: nodes surrounding the input metabolites support warmer temperatures and hence the proposed enzymes within the pathway are close to the MS-derived metabolites. The suggestion of these enzymes gives a richer view within the pathway and can help generate new biological hypotheses. This context also depicts L-glutamate, an extra metabolite suggested by the method, which is surrounded by MS-derived metabolites and also found through NMR.

The lack of a gold standard procedure and a reference benchmark dataset with known biology for pathway enrichment (Huang et al., 2009; Khatri et al., 2012) encouraged the analysis of metabolic changes using isotopic labelling and NMR. The novelty of our tool includes the generation of a comprehensive subgraph that contains more than pathways and compounds – conse-

quently we also partially validate the reactions that appear in the subgraph. The definition of performance is not straightforward, given the lack of means to prove that a node (compound, reaction) is not affected, so the usual quality measures (false positives, true negatives) are not applicable. Results show that our reported reactions have lower resistance distances to the 8 metabolites found by NMR than all the reactions involving any of the MS-derived metabolites (Table 9). The choice of resistance distance as a validation metric is motivated by the presence of hubs in the metabolic network that affect the usual shortest paths metrics, meaning that connections through very specific metabolic reactions are masked by very general reactions involving hubs like adenosine triphosphate (ATP). As resistance distance takes into account the whole graph structure, and specifically the presence of multiple shortest paths, it is more informative than shortest paths distance.

6.5 CONCLUSIONS

We propose a secondary analysis methodology for summary metabolomics data that combines pathway enrichment and sub-network analysis. Instead of reporting a list of pathways, we build meaningful sub-pathway representations of the biology at several molecular levels, derived through a null diffusive process on a curated graph object built from the KEGG database. This approach accounts for pathway over-representation, topology and crosstalk. Nodes reported as relevant are drawn in a comprehensive heterogeneous network that contains not only pathways and compounds, but also enzymes, reactions and KEGG modules. This richer biological context adds value to the top pathway hits by suggesting possible paths through which affected compounds translate into dysregulated pathways.

The proposed methodology has been tested and assessed in a case-control study, where the suggested pathways are consistent with alternative pathway enrichment techniques and the reported reactions have been partially validated through NMR-based tracking of glucose carbon. Our analysis suggests that the Glutathione metabolism is one of the most affected pathways. Glutathione is critical for the suppression of reactive oxygen species and this result is consistent with our preliminary observations that these cells exhibit higher levels of mitochondrial reactive oxygen species. Tests on simulated data suggest that our methodology can benefit from pathway signals whose generative model is network-based. These results support the potential of our novel methods for aiding in the interpretation of complex metabolomics datasets.

ACKNOWLEDGEMENTS

This work was supported by the Spanish Ministry of Economy and Competitiveness (www.mineco.gob.es) [BFU2012-39521 and BFU2015-68354 to T.S., TEC2014-60337-R to A.P., SAF2011-30578 and BFU2014-57466 to O.Y.].

O.Y., A.P. and S.P. thank for funding the Spanish Biomedical Research Centre in Diabetes and Associated Metabolic Disorders (CIBERDEM) and the

Networking Biomedical Research Centre in the subject area of Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), both initiatives of Instituto de Investigación Carlos III (ISCIII). S.A. was supported by a Finnish Cultural Society Fellowship. S.P. thanks the AGAUR FI-scholarship programme. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Takeda Cambridge Ltd provided support in the form of salaries for authors [FF], but did not have any additional role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript. The specific roles of these authors are articulated in the 'author contributions' section.

Competing interests

We have the following interests: Francesc Fernández-Albert has been employed by Takeda Cambridge Ltd. There are no patents, products in development or marketed products to declare. This does not alter our adherence to all the PLOS ONE policies on sharing data and materials, as detailed online in the guide for authors.

SUPPORTING INFORMATION

S1 TABLE. **E**xperimental data results. Reported subgraphs and pathway analysis using IMPaLA and MetaboAnalyst on the experimental dataset.

S2 TABLE. **I**MPaLA example data. Reported pathways for the IMPaLA example data using top 250 z-scores in heat diffusion, IMPaLA and MetaboAnalyst.

S3 TABLE. **M**etaboAnalyst example data. Reported pathways for the MetaboAnalyst example data using top 250 z-scores in heat diffusion, IMPaLA and MetaboAnalyst.

S1 APPENDIX. **G**raph structure and curation. Details on how to generate and curate the KEGG graph.

S2 APPENDIX. **H**eat diffusion process. Formulation of the heat diffusion scoring method.

S3 APPENDIX. **P**ageRank. Formulation of the PageRank web ranking algorithm.

S4 APPENDIX. **N**ull models. Definition of the null models and visualisation of the pathway correlation matrix.

S5 APPENDIX. Details on reported solutions. Solution stratification, CC evolution, computational cost of Monte Carlo permutations and damping factor influence.

REFERENCES

Aivio, Suvi Marjaana and Travis H. Stracker

2014 *The Role of EXD2 in the maintenance of mithocondrial homeostasis*, Doctoral Thesis, Universitat Pompeu Fabra, Departament de Ciències Experimentals i de la Salut.

Alonso, Arnald, Sara Marsal, and Antonio Julià

2015 "Analytical methods in untargeted metabolomics: state of the art in 2015", *Front. Bioeng. Biotechnol.*, 3, 23, ISSN: 2296-4185.

Bapat, RB

2004 "Resistance matrix of a weighted graph", *MATCH-COMMUN MATH CO*, 50, pp. 73-82.

Bonals, Lluís Albert

2005 *Transferència de calor: apunts de classe*.

Chagoyen, Monica and Florencio Pazos

2011 "MBRole: enrichment analysis of metabolomic data", *Bioinformatics*, 27, 5, pp. 730-731.

2013 "Tools for the functional interpretation of metabolomic experiments", *Brief. Bioinform.*, 14, 6, pp. 737-744.

Croft, David, Antonio Fabregat Mundo, Robin Haw, Marija Milacic, Joel Weiser, Guanming Wu, Michael Caudy, Phani Garapati, Marc Gillespie, Maulik R. Kamdar, Bijay Jassal, Steven Jupe, Lisa Matthews, Bruce May, Stanislav Palatnik, Karen Rothfels, Veronica Shamovsky, Heeyeon Song, Mark Williams, Ewan Birney, Henning Hermjakob, Lincoln Stein, and Peter D'Eustachio

2014 "The Reactome pathway knowledgebase", *Nucleic Acids Res.*, 42, Database issue, pp. D472-D477.

Csardi, Gabor and Tamas Nepusz

2006 "The igraph software package for complex network research", *InterJournal*, Complex Systems, p. 1695, http://igraph.org.

DeBerardinis, Ralph J, Anthony Mancuso, Evgueni Daikhin, Ilana Nissim, Marc Yudkoff, Suzanne Wehrli, and Craig B Thompson

2007 "Beyond aerobic glycolysis: transformed cells can engage in glutamine metabolism that exceeds the requirement for protein and nucleotide synthesis", *Proc. Natl. Acad. Sci. U.S.A.*, 104, 49, pp. 19345-19350.

Donato, Michele, Zhonghui Xu, Alin Tomoiaga, James G Granneman, Robert G MacKenzie, Riyue Bao, Nandor Gabor Than, Peter H Westfall, Roberto Romero, and Sorin Draghici

2013 "Analysis and correction of crosstalk effects in pathway analysis", *Genome Res.*, 23, 11, pp. 1885-1893.

Draghici, Sorin, Purvesh Khatri, Adi Laurentiu Tarca, Kashyap Amin, Arina Done, Calin Voichita, Constantin Georgescu, and Roberto Romero

2007 "A systems biology approach for pathway level analysis", *Genome Res.*, 17, 10, pp. 1537-1545.

Faust, Karoline, Pierre Dupont, Jérôme Callut, and Jacques van Helden

2010 "Pathway discovery in metabolic networks by subgraph extraction", *Bioinformatics*, 26, 9, pp. 1211-1218, ISSN: 13674803.

Fernández-Albert, Francesc, Rafael Llorach, Cristina Andrés-Lacueva, and Alexandre Perera

2014 "An R package to analyse LC/MS metabolomic data: MAIT (Metabolite Automatic Identification Toolkit)", *Bioinformatics*, 30, 13, pp. 1937-1939.

Haynes, Winston A, Roger Higdon, Larissa Stanberry, Dwayne Collins, and Eugene Kolker

2013 "Differential expression analysis for pathways", *PLOS Comput. Biol.*, 9, 3, e1002967.

Huang, Da Wei, Brad T Sherman, and Richard A Lempicki

2009 "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists", *Nucleic Acids Res.*, 37, 1, pp. 1-13.

Ideker, Trey, Owen Ozier, Benno Schwikowski, and Andrew F Siegel

2002 "Discovering regulatory and signalling circuits in molecular interaction networks", *Bioinformatics*, 18, suppl 1, S233-S240.

Kamburov, Atanas, Rachel Cavill, Timothy M. D. Ebbels, Ralf Herwig, and Hector C. Keun

2011 "Integrated pathway-level analysis of transcriptomics and metabolomics data with IMPaLA", *Bioinformatics*, 27, 20, pp. 2917-2918.

Kanehisa, Minoru, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshiaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, and Yoshihiro Yamanishi

2008 "KEGG for linking genomes to life and the environment", *Nucleic Acids Res.*, 36, Database issue, pp. D480-D484.

Kankainen, Matti, Peddinti Gopalacharyulu, Liisa Holm, and Matej Orešič

2011 "MPEA – metabolite pathway enrichment analysis", *Bioinformatics*, 27, 13, pp. 1878-1879.

Karnovsky, Alla, Terry E. Weymouth, Tim Hull, V. Glenn Tarcea, Giovanni Scardoni, Carlo Laudanna, Maureen A. Sartor, Kathleen A. Stringer, H. V. Jagadish, Charles F. Burant, Brian D. Athey, and Gilbert S. Omenn

2012 "Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data", *Bioinformatics*, 28, 3, pp. 373-380.

Kelder, Thomas, Martijn P van Iersel, Kristina Hanspers, Martina Kutmon, Bruce R Conklin, Chris T Evelo, and Alexander R Pico

2012 "WikiPathways: building research communities on biological pathways", *Nucleic Acids Res.*, 40, Database issue, pp. D1301-D1307.

Kessler, Nikolas, Heiko Neuweger, Anja Bonte, Georg Langenkämper, Karsten Niehaus, Tim W. Nattkemper, and Alexander Goesmann

2013 "MeltDB 2.0-advances of the metabolomics software system", *Bioinformatics*, 29, 19, pp. 2452-2459.

Khatri, Purvesh, Marina Sirota, and Atul J. Butte

2012 "Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges", *PLOS Comput. Biol.*, 8, 2.

Li, Xianbin, Liangzhong Shen, Xuequn Shang, and Wenbin Liu

2015 "Subpathway analysis based on signaling-pathway impact analysis of signaling pathway", *PLOS ONE*, 10, 7, e0132813.

Mitra, Koyel, Anne-Ruxandra Carvunis, Sanath Kumar Ramesh, and Trey Ideker

2013 "Integrative approaches for finding modular structure in biological networks", *Nat. Rev. Genet.*, 14, 10, pp. 719-732.

Nicholson, Jeremy K., John Connelly, John C. Lindon, and Elaine Holmes

2002 "Metabonomics: a platform for studying drug toxicity and gene function", *Nat. Rev. Drug Discov.*, 1, 2, pp. 153-161.

North, Bernard V, David Curtis, and Pak C Sham

2002 "A note on the calculation of empirical P values from Monte Carlo procedures", *Am. J. Hum. Genet.*, 71, 2, p. 439.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd

1999 *The PageRank citation ranking: bringing order to the Web*, tech. rep., Stanford InfoLab.

Paull, Evan O, Daniel E Carlin, Mario Niepel, Peter K Sorger, David Haussler, and Joshua M Stuart

2013 "Discovering causal pathways linking genomic events to transcriptional states using Tied Diffusion Through Interacting Events (TieDIE)", *Bioinformatics*, 29, 21, pp. 2757-2764.

R Core Team

2015 *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Rahnenführer, Jörg, Francisco S Domingues, Jochen Maydt, and Thomas Lengauer

2004 "Calculating the statistical significance of changes in pathway activity from gene expression data", *Stat. Appl. Genet. Mol.*, 3, 1.

Reddy, Junuthula Narasimha and David K Gartling

2010 *The finite element method in heat transfer and fluid dynamics*.

Smoot, Michael E., Keiichiro Ono, Johannes Ruscheinski, Peng-Liang Wang, and Trey Ideker

2011 "Cytoscape 2.8: new features for data integration and network visualization", *Bioinformatics*, 27, 3, pp. 431-432.

Subramanian, Aravind, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S, Lander, and Jill P. Mesirov

2005 "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles", *Proc. Natl. Acad. Sci. U.S.A.*, 102, 43, pp. 15545-15550.

Tarca, Adi Laurentiu, Sorin Draghici, Gaurav Bhatti, and Roberto Romero

2012 "Down-weighting overlapping genes improves gene set analysis", *BMC Bioinform.*, 13, 1, p. 136.

Vandin, Fabio, Eli Upfal, and Benjamin J Raphael

2011 "Algorithms for detecting significantly mutated pathways in cancer", *J. Comput. Biol.*, 18, 3, pp. 507-522.

Vinaixa, Maria, Emma L Schymanski, Steffen Neumann, Miriam Navarro, Reza M Salek, and Oscar Yanes

2015 "Mass spectral databases for LC/MS and GC/MS-based metabolomics: state of the field and future prospects", *TrAC-Trend. Anal. Chem.*, 78, pp. 23-25.

Weckwerth, Wolfram

2003 "Metabolomics in Systems Biology", *Annu. Rev. Plant Biol.*, 54, 1, pp. 669-689.

Wickham, Hadley

2009 *ggplot2: Elegant Graphics for Data Analysis*, Springer-Verlag New York, ISBN: 978-0-387-98140-6, http://ggplot2.org.

Wishart, David S., Timothy Jewison, Anchi Guo, Michael Wilson, Craig Knox, Yifeng Liu, Yannick Djoumbou, Rupasri Mandal, Farid Aziat, Edison Dong, et al.

2013 "HMDB 3.0 - The Human Metabolome Database in 2013", *Nucleic Acids Res.*, 41, Database issue, pp. D801-D807.

Xia, Jianguo, Igor V Sinelnikov, Beomsoo Han, and David S Wishart

2015 "MetaboAnalyst 3.0 – making metabolomics more meaningful", *Nucleic Acids Res.*, 43, Web Server issue, W251-W257.

Xia, Jianguo and David S. Wishart

2010 "MSEA: a web-based tool to identify biologically meaningful patterns in quantitative metabolomic data", *Nucleic Acids Res.*, 38, Web Server issue, W71-W77.

7 THE R PACKAGE FELLA

FELLA: AN R PACKAGE TO ENRICH METABOLOMICS DATA

Pathway enrichment techniques are useful for understanding experimental metabolomics data. Their purpose is to give context to the affected metabolites in terms of the prior knowledge contained in metabolic pathways. However, the interpretation of a prioritized pathway list is still challenging, as pathways show overlap and cross talk effects.

We introduce FELLA, an R package to perform a network-based enrichment of a list of affected metabolites. FELLA builds a hierarchical representation of an organism biochemistry from the Kyoto Encyclopedia of Genes and Genomes (KEGG), containing pathways, modules, enzymes, reactions and metabolites. In addition to providing a list of pathways, FELLA reports intermediate entities (modules, enzymes, reactions) that link the input metabolites to them. This sheds light on pathway cross talk and potential enzymes or metabolites as targets for the condition under study. FELLA has been applied to six public datasets –three from *Homo sapiens*, two from *Danio rerio* and one from *Mus musculus*– and has reproduced findings from the original studies and from independent literature.

The R package FELLA offers an innovative enrichment concept starting from a list of metabolites, based on a knowledge graph representation of the KEGG database that focuses on interpretability. Besides reporting a list of pathways, FELLA suggests intermediate entities that are of interest per se. Its usefulness has been shown at several molecular levels on six public datasets, including human and animal models. The user can run the enrichment analysis through a simple interactive graphical interface or programmatically. FELLA is publicly available in Bioconductor under the GPL-3 license.

7.1 BACKGROUND

Metabolomics is the science that measures lightweight molecules in living organisms and stands as a valuable source of biomarkers and biological knowledge (Madsen et al., 2010). The preprocessing of such data can be achieved through pipelines like MeltDB (Kessler et al., 2013) or MAIT (Fernández-Albert et al., 2014). Once metabolite abundances are available, pathway analysis tools ease data interpretation (Khatri et al., 2012) by framing the affected metabolites in terms of contextual knowledge. Databases

This chapter is a postprint of the following journal article: Picart-Armada, Sergio, Francesc Fernández-Albert, Maria Vinaixa, Oscar Yanes, and Alexandre Perera-Lluna. "FELLA: an R package to enrich metabolomics data". *BMC bioinformatics 19, no. 1 (2018): 538*.

like the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al., 2011) are sources of curated pathway data. The classification of enrichment techniques used here follows the review in (Khatri et al., 2012).

Over representation analysis (ORA) approaches are based on testing the proportion of a list of affected metabolites inside a pathway. ORA is available in tools like the web server MetaboAnalyst (Xia et al., 2015) and the R package *clusterProfiler* (Yu et al., 2012). Functional class scoring (FCS) approaches use quantitative data instead and seek subtle but coordinated changes in the metabolites belonging to a pathway. MSEA in MetaboAnalyst and IMPaLA (Kamburov et al., 2011) contain implementations of FCS for metabolomics. Pathway topology-based (PT) approaches further include topological measures of the metabolites in the statistic, accounting for their inequivalence within the pathway. PT analyses can be performed using MetaboAnalyst.

Here, we introduce the R package *FELLA*, available in Bioconductor (Huber et al., 2015), for metabolomics data interpretation that combines pathway enrichment with network analysis. The list of affected metabolites and the reported pathways are connected through intermediate entities -reactions, enzymes, modules- in a heterogeneous network layout. This suggests how the perturbation spreads at the pathway level and how pathways cross talk, enhancing the interpretability of the output.

7.2 IMPLEMENTATION

FELLA is an R package that performs metabolomics data enrichment starting from (I) a network derived from KEGG and (II) a list of KEGG compounds (Fig. 25). A sub-network relevant to the input is extracted from (I) using network propagation algorithms that start from the labels in (II), providing a data enrichment that goes beyond a pathway list. The purpose of *FELLA* is to elaborate a biological explanation that justifies how the input metabolites can reach the reported pathways, as well as perspective on pathway cross talk. Two user guides illustrate the principles and the usage of *FELLA*: a quickstart (additional file 3) and an in-depth vignette with implementations details and three real examples (additional file 1). Two additional vignettes (additional files 4 and 6) serve as case studies for non-human organisms.

7.2.1 Methodology

The cornerstone of *FELLA* is its knowledge graph representation of the biochemistry in KEGG at several molecular levels. The network is hierarchical and connects KEGG compounds (metabolites) to KEGG pathways through intermediate entities, namely reactions, enzymes and KEGG modules, see figure 24. Such connections (edges) are obtained directly from KEGG annotations. The presence of intermediate levels allows inference at their level, meaning that relevant reactions, enzymes and KEGG modules can be suggested just by starting from a list of affected metabolites. This feature is

evaluated in several case studies, by linking the suggested enzymatic families and reactions to literature and to original findings within the studies.

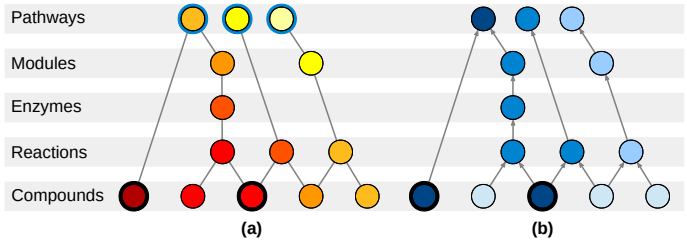


Figure 24: Node arrangement for the knowledge model used by *FELLA*. Entities are organised in a hierarchical manner, from bottom to top: KEGG compounds or metabolites, reactions, enzymes, KEGG modules and pathways. Binary labels at the level of metabolites are propagated to the rest of the network and a relevant, small sub-network is automatically reported. Nodes are ranked using the network propagation algorithms (a) heat diffusion and (b) PageRank. The affected metabolites are highlighted with a black ring. For heat diffusion (a), affected metabolites are forced to generate unitary flow. Every pathway is highlighted with a blue ring, representing its connection to a cool boundary node. In equilibrium, the highest temperature pathways (and nodes) will have the greatest heat flow, suggesting a relevant role in the experiment. For PageRank (b), affected metabolites are the start of random walks. PageRank scores, represented by the intensity of the blue colour, will attain higher values in the frequently reached random walk nodes. Figure extracted from (Picart-Armada et al., 2017).

In order to report a sub-network, nodes are ranked according to a scoring function –based on network propagation– and only the top scoring nodes are returned. Two algorithms are supported for propagating the labels from the affected metabolites: a classical heat diffusion approach (Vandin et al., 2011) and the PageRank web ranking algorithm (Page et al., 1999). Further details on the network propagation settings can be found in (Picart-Armada et al., 2017) and in additional file 1. The main difference between both algorithms is that heat diffusion is undirected whereas PageRank is directed upwards. In practice, contrary to PageRank, heat diffusion will frequently report new metabolites because heat is allowed to propagate back to compounds from the upper levels (Picart-Armada et al., 2017). This behaviour can ease the discovery of intermediate metabolites that lay close to the input metabolites and tend to connect them. An example of its usefulness can be found in the gilt-head bream study.

As exposed in (Picart-Armada et al., 2017), ranking nodes according to their raw diffusion scores suffers from a strong bias, related to the node level and topological features. This is addressed by normalising the diffusion score of every node using its background distribution under input permutations. Permutations can be simulated through Monte Carlo trials to obtain an empirical p-value, labelled as `p-score`. Alternatively, a parametric z-score can be obtained without requiring Monte Carlo trials. The `p-score` is obtained by transforming the z-score to lie in the $[0, 1]$ interval through the cumulative distribution function of a standard normal distribution. Under both statistical approximations, nodes with the lowest p-scores are reported

as the suggested sub-network. Note that p-scores are used as a ranker rather than for testing hypotheses.

An optional filter allows the removal of small connected components from the reported sub-network. When building the database, a number of random sub-networks are sampled to characterise how infrequent a connected component of order at least r is when k nodes are uniformly sampled. The assumption behind this filter is that meaningful inputs encompass metabolites relatively close to each other within the knowledge graph, prone to be reported in large connected components involving most of them.

7.2.2 Classes

FELLA relies on two classes: *FELLA.DATA* for the internal knowledge representation, based on the *igraph* R package (Csardi and Nepusz, 2006), and *FELLA.USER* for the user analysis, see figure 25. These classes contain subclasses, invisible to the user and described in the additional file 1. The functions to manipulate both classes are described below, following the three blocks from figure 25.

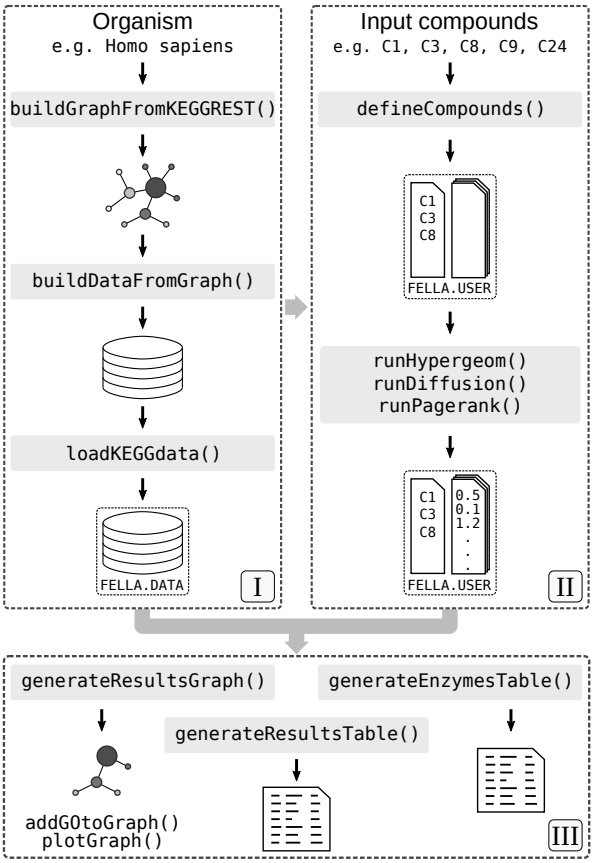


Figure 25: Design of the R package *FELLA*. **(I)** creation of a graph object from an organism code and its database, **(II)** ID mapping and propagation algorithms (diffusion, PageRank) to score all the nodes, **(III)** node prioritisation and results exporting.

Block I: local database

The function `buildGraphFromKEGGREST()` retrieves the tabular KEGG data for the desired organism and builds the knowledge graph as described in (Picart-Armada et al., 2017). Then, a database can be built from the graph and stored in a local folder using `buildDataFromGraph()`. Databases are needed for the enrichment and should be loaded through the function `loadKEGGdata()`.

Block II: enrichment analysis

Once the database is loaded, i.e. the *FELLA.DATA* object is in memory, `defineCompounds()` maps the list of input metabolites, in the form of KEGG identifiers, to the internal representation, providing a *FELLA.USER* object. Then, the propagation algorithms in (Picart-Armada et al., 2017) are run to score the graph nodes. `runDiffusion()` uses the undirected heat diffusion model (Vandin et al., 2011) whereas `runPagerank()` runs the directed PageRank algorithm (Page et al., 1999). Both approaches are automatically followed by the statistical normalisation, either as a parametric z-score (`approx = "normality"`) or as a simulated permutation analysis (`approx = "simulation"`), see table 10. The wrapper `enrich()` performs the metabolite mapping and the desired propagation algorithm (argument `method`) and statistical normalisation with a single call.

Table 10: Scoring methods offered in *FELLA*, chosen by the `enrich` function arguments `method` and `approx`. Each row corresponds to a method mentioned in the original publication (Picart-Armada et al., 2017). The method `hypergeom` is Fisher's exact test, included for reference. Method `diffusion` scores the nodes using the heat diffusion model to score the nodes. Method `pagerank` uses the PageRank algorithm on an upwards-directed version of the network. Both scores undergo a statistical normalisation to remove structural biases, controlled through the `approx` argument. The user can choose the fast, parametric z-scores (`normality`) or the slower, non-parametric permutation analysis (`simulation`). N/A: non-applicable.

Method	Approx	Notation in (Picart-Armada et al., 2017)	Comment
hypergeom	N/A	hypergeometric test	Included for reference
diffusion	normality	HD norm	Heat diffusion scores followed by z-scores
diffusion	simulation	HD sim	Heat diffusion scores followed by permutations
pagerank	normality	PR norm	PageRank scores followed by z-scores
pagerank	simulation	PR sim	PageRank scores followed by permutations

Block III: exporting results

Finally, the best scoring KEGG entries can be visualised through `plot()`, exported as a sub-network with `generateResultsGraph()`, or in tabular format with `generateResultsTable()`. A dedicated table with the reported enzymes and its associated genes can be obtained with `generateEnzymesTable()`. Alternatively, `exportResults()` allows writing such objects directly to files.

7.2.3 User interface

FELLA includes an interactive graphical interface, based on the R package *shiny* (Chang et al., 2018) and deployable through `launchApp()`. The interface is divided with four tabs that encompass most options from *FELLA* (figure 26). Currently, the database needs to be built outside the graphical interface and prior to its usage.

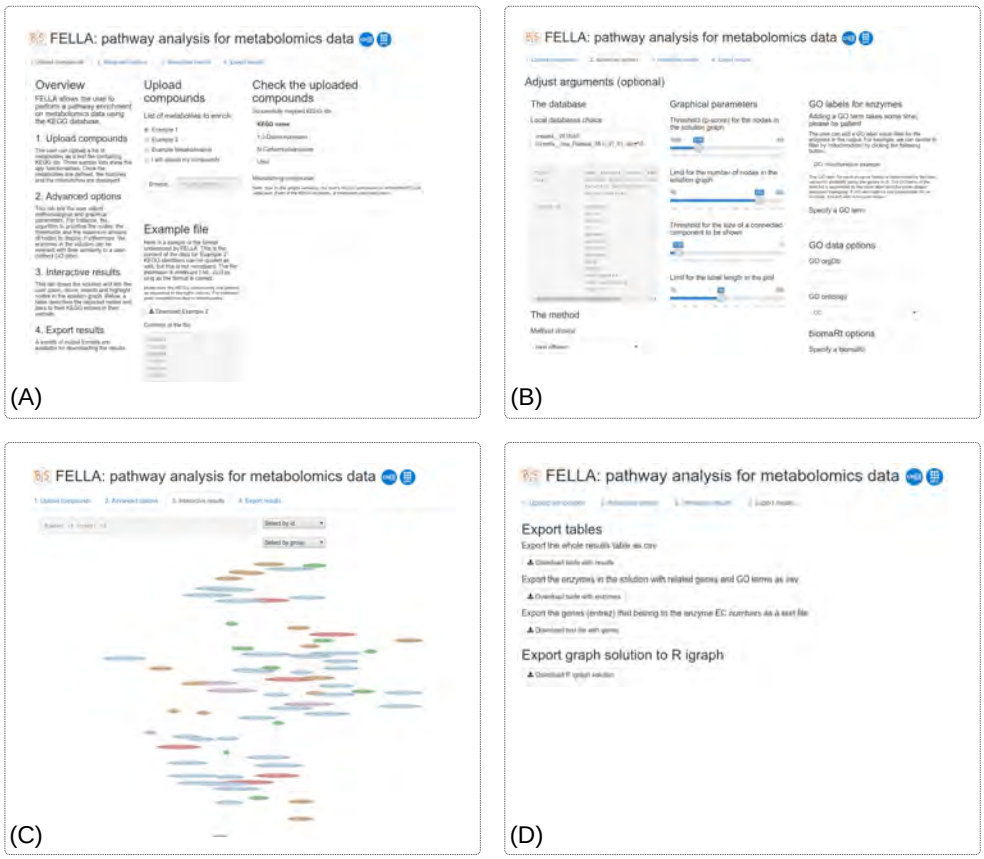


Figure 26: Perspective of the interactive app within *FELLA*. The app is composed by four tabs: (A) compounds upload, (B) advanced options, (C) results and (D) export. The lay user can rapidly explore his or her data without knowing the details about the syntax in *FELLA*.

Compounds upload

This tab contains a general description of the interface and a handle to submit the input metabolite list as a text file. Examples are provided as well. The right panel shows the mapped and the mismatching compounds with regard to the current database.

Advanced options

Widgets from this tab adjust the main function arguments for customising the enrichment procedure. They ease database choice from the internal package directory, method and approximation definition and parameter tweaking. It also allows the semantic similarity analysis on the reported en-

zymes, using the R package *GOSemSim* (Yu, F. Li, et al., 2010) with the Gene Ontology annotations (Consortium, 2015).

Results and discussion

The results section mainly consists of an interactive network plot with the top k KEGG entries. Nodes can be moved, selected, queried and hovered to reveal the original KEGG entry. An interactive table lies below the plot and expands the data on the nodes.

Export

The last tab offers several options to download the reported sub-network (tabular format or R object) and enzymes (tabular format).

7.3 RESULTS

The algorithmic part of *FELLA* has already been discussed and validated in (Picart-Armada et al., 2017). The usage of *FELLA* is hereby demonstrated on three public human studies on epithelial cells (Chen et al., 2015), ovarian cancer cells (Yu et al., 2014) and febrile illnesses (Decuypere et al., 2016). The examples guide the user on how to build the database, format the input data, complete the enrichment and export its results (see additional file 1). *FELLA* reproduces findings from the original publications, not only in the form of pathway hits but also as newly suggested enzymes and metabolites. The additional file 2 shows further details on the metabolites in each input and the reported sub-networks.

To demonstrate its usefulness outside human studies, *FELLA* is applied to two datasets from a gilt-head bream study (Ziarrusta et al., 2018) and a mouse model of non-alcoholic fatty liver disease (Gogiashvili et al., 2017). The complete analyses can be respectively found in additional files 4 and 6, whereas their respective R workspaces are saved in files 5 and 7. Table 11 summarises the knowledge graphs in the *FELLA.DATA* object for each organism.

Table 11: Summary of the *FELLA.DATA* objects used for the three human and the three non-human datasets. Generalist and overview pathways are excluded from the models, see additional files 1, 4 and 6 for further details on each organism.

Organism	KEGG release	Nodes	Pathways	Modules	Enzymes	Reactions	Compounds
Homo sapiens	85.0+/02-16	9899	314	182	1110	4829	3464
Danio rerio	87.0+/09-14	9637	162	179	995	4843	3458
Mus musculus	87.0+/09-14	9909	316	185	1107	4843	3458

7.3.1 Epithelial cells dataset

The epithelial cancer cells study (Chen et al., 2015) runs an in vitro model of dry eye in which the human epithelial cells IOBA-NHC are put under hy-

perosmotic stress. The list of 9 metabolites hereby used reflects metabolic changes in "Treatment 1" (24 hours in serum-free media at 380 mOsm) against control (24 hours at 280 mOsm). The metabolites have been extracted from "Table 1" in the original manuscript and mapped to 9 KEGG ids, from which 8 map to the *FELLA.DATA* object. The enrichment (sub-network in figure 27) is obtained by leaving the default parameters in *FELLA*: `method = "diffusion"`, `approx = "normality"` and `threshold = 0.05`. The amount of nodes has been limited to `nlimit = 150`.

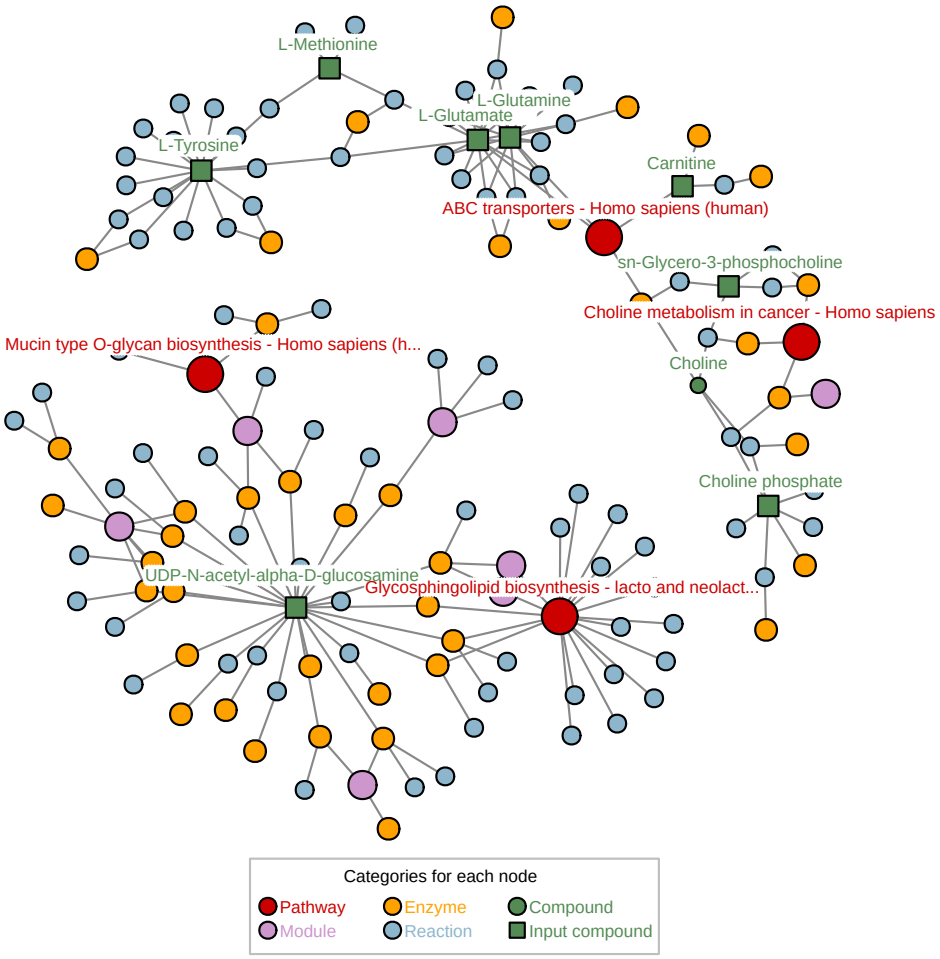


Figure 27: Results of the node prioritisation by *FELLA* in the epithelial cells dataset. The user is given a list of positive entities, after a score threshold described in (Picart-Armada et al., 2017), with information on how the input metabolites reach the suggested pathways and on how these pathways cross talk. Plots of the ovarian and malaria datasets can be found in the additional file 2.

The activation of the "glycerophosphocholine synthesis" rather than the "carnitine" response is a main result in the original work (Chen et al., 2015). *FELLA* highlights the related pathway "choline metabolism in cancer" and the "choline" metabolite as well. Another key process is the "O-linked glycosilation", which is close to the KEGG module "O-glycan biosynthesis, mucin type core" and to the KEGG pathway "Mucin type O-glycan biosynthesis". Finally, *FELLA* reproduces the finding of "UAP1" by report-

ing the enzyme "2.7.7.23", named "UDP-N-acetylglucosamine diphosphorylase". "UAP1" is a key protein in the study, pinpointed by iTRAQ (Isobaric Tags for Relative and Absolute Quantitation) and validated via western blot.

7.3.2 Ovarian cancer cells dataset

The second dataset has been extracted from the study on metabolic responses of ovarian cancer cells (Yu et al., 2014). OCSCs are isogenic ovarian cancer stem cells derived from the OVCAR-3 ovarian cancer cells. The abundances of 6 metabolites are affected by the exposure to several environmental conditions: glucose deprivation, hypoxia and ischemia. From those, 5 metabolites map to the *FELLA.DATA* object. The sub-network is obtained by leaving the default parameters and setting a limit of `nlimit = 150` nodes.

Several "TCA cycle"-related entities are highlighted, also found by the authors and by previous work (Pollard et al., 2003). It also mentions "sphingosine degradation", closely related to the reported "sphingosine metabolism" in the original work. Enzymes that have been formerly related to cancer are suggested within the TCA cycle, like "fumarate hydratase" (Lehtonen et al., 2007; Pithukpakorn et al., 2006; Pollard et al., 2003), "succinate dehydrogenase" (Ni et al., 2008; Pollard et al., 2003) and "aconitase" (Singh et al., 2006). Another suggestion is "lysosome"(s), known to suffer changes in cancer cells and directly affect apoptosis (Kirkegaard and Jäättelä, 2009). Finally, the graph contains several "hexokinases", potential targets to disrupt glycolysis, a fundamental need in cancer cells (Kaelin and Thompson, 2010).

7.3.3 Malaria dataset

The metabolites in this example are related to the distinction between malaria and other febrile illnesses (Decuypere et al., 2016). Specifically, the list of 11 KEGG identifiers (9 in the *FELLA.DATA* object) has been extracted from the original supplementary data spreadsheet, using all the possible KEGG matches for the "non malaria" patient group. The sub-network is obtained by leaving the default parameters and setting a limit of `nlimit = 50` nodes.

In this case, the depicted subnetwork contains the modules "C21-Steroid hormone biosynthesis, progesterone =>corticosterone/aldosterone" and "C21-Steroid hormone biosynthesis, progesterone =>cortisol/cortisone", related to the "corticosteroids" as a main pathway reported in the original text. This is part of the also reported "Aldosterone synthesis and secretion"; aldosterone is known to show changes related to fever as a metabolic response to infection (Beisel, 1975). Another plausible hit in the sub-network is "linoleic acid metabolism", as erythrocytes infected by various malaria parasytes can be enriched in linoleic acid (Fitch et al., 2000). In addition, the pathway "sphingolipid metabolism" can play a role in the immune response (Maceyka and Spiegel, 2014; Seo et al., 2011). As for the enzymes, "3alpha-hydroxysteroid 3-dehydrogenase (Si-specific)" and "Delta4-3-oxosteroid 5beta-reductase" are related to three input metabolites each and might be candidates for further examination.

7.3.4 Oxybenzone exposition on gilt-head bream datasets

A study of the consequences of the oxybenzone contaminant on gilt-head bream (Ziarrusta et al., 2018) found five dysregulated KEGG metabolites in their liver and eleven in their plasma. The study justified its findings through literature and complemented them with insights provided by *FELLA*. Here, both metabolite lists are used to build suggested sub-networks with the default parameters and fixing `nlimit = 250`. The *FELLA.DATA* object is built for the *Danio Rerio* organism, a common approximation when annotations specific to gilt-head bream are not available. Further details can be found in the vignette (additional file 4) and its workspace (additional file 5).

The enrichment on the liver-derived metabolites links all of them within a connected component of roughly 100 nodes. It points to "Phenylalanine metabolism" as one of the key metabolic pathways, in accordance with the main results from the article. Among the suggested metabolites, "Tyrosine" is of particular help to explain the connection between the affected metabolites (see *Fig. 2* from (Ziarrusta et al., 2018)).

Plasma metabolites involve a more complex scenario. *FELLA* reports ten out of the eleven metabolites in a connected component involving around 120 nodes. Seven pathways are suggested, from which "Linoleic acid metabolism", "Biosynthesis of unsaturated fatty acids", "alpha-Linolenic acid metabolism", "Glycerophospholipid metabolism" and "Glycine, serine and threonine metabolism" were used to build a comprehensive picture of the metabolic changes in the original manuscript (*Fig. 3* from (Ziarrusta et al., 2018)). Such figure brings a structured overview that narrows down the core processes, also backed up by prior publications. Likewise, by drawing intermediate metabolites found through *FELLA*, like "Linoleic acid" and "Phosphatidylcholine", it achieves a cohesive representation of the input metabolites.

7.3.5 Non-alcoholic fatty liver disease mouse model

This dataset exemplifies how *FELLA* can also be applied on an animal disease model. Metabolites in liver tissue from leptin-deficient *ob/ob* mice and wild-type were compared using Nuclear Magnetic Resonance, whereas several candidate genes were further investigated for differences in expression (Gogiashvili et al., 2017). Six affected metabolites are introduced in *FELLA*, leaving the default parameters and `nlimit = 250`. The *FELLA.DATA* object is built for the *Mus musculus* organism. The vignette with the whole analysis is provided provided as additional file 6, whereas its R workspace can be found in additional file 7.

The sub-network found by *FELLA* involves "N,N-Dimethylglycine", a marginally significant metabolite in the experimental data but with a relevant role within the findings from the study. Regarding the genes, *FELLA* is able to find the enzyme associated to *Bhmt*, validated and discussed in the study. The enzyme associated to *Cbs*, another central hit, is not directly found. However, its ranking (top 17% among enzymes) and especially that of its reaction (top 3% among reactions) are highly suggestive. We also show how other (1) related metabolites, found by leveraging the expression

data, and (2) differentially expressed genes, taken from an external study (Godoy et al., 2016), tend to have top p-scores in the prioritisation provided by *FELLA*.

7.4 CONCLUSIONS

We present *FELLA*, an R package for enriching metabolomics data, focused on interpretability. It can be used either programmatically or through a simple user interface. *FELLA* offers a comprehensive enrichment by depicting the intermediate reactions, enzymes and modules that link the input metabolites to the relevant pathways. This layout gives a biological picture with information of the pathway overlap and the connections between the entities of interest, while suggesting enzymes and possibly other metabolites for further study. The utility of *FELLA* has been demonstrated on six public datasets, both with human and non-human organisms, where reported entities include several original findings in addition to results from third studies. *FELLA* is publicly available in the Bioconductor public repository under the GPL-3 license.

AVAILABILITY AND REQUIREMENTS

- **Project name**: FELLA
- **Project home page**: `https://doi.org/doi:10.18129/B9.bioc.FELLA`, `https://github.com/b2slab/FELLA`
- **Operating system(s)**: platform independent
- **Programming language**: R
- **Other requirements**: none
- **License**: GPL-3
- **Restrictions to use by non-academics**: those derived by the GPL-3 license

ABBREVIATIONS

GPL-3: General Public License version 3; KEGG: Kyoto Encyclopedia of Genes and Genomes; ORA: Over Representation Analysis; FCS: Functional Class Scoring; PT: Pathway Topology-based; TCA: TriCarboxylic Acid; iTRAQ: Isobaric Tags for Relative and Absolute Quantitation; UDP: Uridine DiPhosphate; N/A: Non-Applicable

DECLARATIONS

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

CONSENT TO PUBLISH

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

All data generated or analysed during this study are included in this published article (additional files 2, 5 and 7).

COMPETING INTERESTS

The authors declare that they have no competing interests.

FUNDING

This work was supported by the Spanish Ministry of Economy and Competitiveness (MINECO) [BFU 2014-57466-P to OY, TEC 2014-60337-R and DPI 2017-89827-R to AP]. OY, AP and SP thank for funding CIBERDEM and CIBER-BBN, both initiatives of Instituto de Investigación Carlos III (ISCIII). SP thanks the AGAUR FI-scholarship programme. The funding body had no role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

AUTHORS' CONTRIBUTIONS

SP, FF, MV, OY and AP conceived the software. SP implemented the software and analysed the data. SP wrote the original manuscript. FF, MV, OY and AP critically revised the original manuscript. OY and AP supervised the project. All authors read and approved the final manuscript.

ACKNOWLEDGEMENTS

We would like to thank Haizea Ziarrusta and our collaboration with the Department of Analytical Chemistry, University of the Basque Country (UPV/EHU), Leioa, for using, discussing and helping improve our software.

We would also like to thank the anonymous reviewers for their valuable comments.

ADDITIONAL FILES

Additional file 1 — FELLA.pdf

User guide within the R package *FELLA* with background, implementation details and three real examples on its usage.

Additional file 2 — datasets.zip

Descriptive files on the three human datasets: a summary of the inputs (descriptive_input.csv), input and reported subgraph in each dataset (dataset_input.csv, dataset_subgraph.csv and dataset_subgraph.pdf), hits discussed in the results section (descriptive_hits.csv). Also contains the database object (fella_data.RData) and metadata about the database (info_fella_data.txt), the KEGG version (info_kegg.txt) and the R session (info_session.txt).

Additional file 3 — quickstart.html

User guide within *FELLA* showing fast and concise toy examples of its application.

Additional file 4 — zebrafish.pdf

Case study with *FELLA*: two datasets on the effect of oxybenzone exposition on gilt-head bream.

Additional file 5 — zebrafish.zip

R workspace from the gilt-head bream datasets.

Additional file 6 — musmusculus.pdf

Case study with *FELLA*: a multi-omic mouse model of non-alcoholic fatty liver disease.

Additional file 7 — musmusculus.zip

R workspace from the mouse model study.

REFERENCES

Beisel, William R

1975 "Metabolic response to infection", *Annu. Rev. Med.*, 26, 1, pp. 9-20.

Chang, Winston, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson

2018 *shiny: Web Application Framework for R*, R package version 1.1.0, https://CRAN.R-project.org/package=shiny.

Chen, Liyan, Jing Li, Tiannan Guo, Sujoy Ghosh, Siew Kwan Koh, Dechao Tian, Liang Zhang, Deyong Jia, Roger W Beuerman, Ruedi Aebersold, et al.

2015 "Global metabonomic and proteomic analysis of human conjunctival epithelial cells (IOBA-NHC) in response to hyperosmotic stress", *J. Proteome Res.*, 14, 9, pp. 3982-3995.

Consortium, Gene Ontology et al.

2015 "Gene ontology consortium: going forward", *Nucleic Acids Res.*, 43, D1, pp. D1049-D1056.

Csardi, Gabor and Tamas Nepusz

2006 "The igraph software package for complex network research", *InterJournal*, Complex Systems, p. 1695.

Decuypere, Saskia, Jessica Maltha, Stijn Deborggraeve, Nicholas JW Rattray, Guiraud Issa, Kaboré Bérenger, Palpouguini Lompo, Marc C Tahita, Thusitha Ruspasinghe, Malcolm McConville, et al.

2016 "Towards Improving Point-of-Care Diagnosis of Non-malaria Febrile Illness: A Metabolomics Approach", *PLoS Negl Trop Dis*, 10, 3, e0004480.

Fernández-Albert, Francesc, Rafael Llorach, Cristina Andrés-Lacueva, and Alexandre Perera

2014 "An R package to analyse LC/MS metabolomic data: MAIT (Metabolite Automatic Identification Toolkit)", *Bioinformatics*, 30, 13, pp. 1937-1939.

Fitch, Coy D, Guang-zuan Cai, and James D Shoemaker

2000 "A role for linoleic acid in erythrocytes infected with Plasmodium berghei", *Biochim. Biophys. Acta-Mol. Basis Dis.*, 1535, 1, pp. 45-49.

Godoy, Patricio, Agata Widera, Wolfgang Schmidt-Heck, Gisela Campos, Christoph Meyer, Cristina Cadenas, Raymond Reif, Regina Stöber, Seddik Hammad, Larissa Pütter, et al.

2016 "Gene network activity in cultivated primary hepatocytes is highly similar to diseased mammalian liver tissue", *Arch. Toxicol.*, 90, 10, pp. 2513-2529.

Gogiashvili, Mikheil, Karolina Edlund, Kathrin Gianmoena, Rosemarie Marchan, Alexander Brik, Jan T Andersson, Jörg Lambert, Katrin Madjar, Birte Hellwig, Jörg Rahnenführer, et al.

2017 "Metabolic profiling of ob/ob mouse fatty liver using HR-MAS 1 H-NMR combined with gene expression analysis reveals alterations in betaine metabolism and the transsulfuration pathway", *Anal. Bioanal. Chem.*, 409, 6, pp. 1591-1606.

Huber, W., V. J. Carey, R. Gentleman, S. Anders, M. Carlson, B. S. Carvalho, H. C. Bravo, S. Davis, L. Gatto, T. Girke, R. Gottardo, F. Hahne, K. D. Hansen, R. A. Irizarry, M. Lawrence, M. I. Love, J. MacDonald, V. Obenchain, A. K. Ole's, H. Pagès, A. Reyes, P. Shannon, G. K. Smyth, D. Tenenbaum, L. Waldron, and M. Morgan

2015 "Orchestrating high-throughput genomic analysis with Bioconductor", *Nature Methods*, 12, 2, pp. 115-121.

Kaelin, William G and Craig B Thompson

2010 "Q&A: Cancer: clues from cell metabolism." *Nature*, 465, 7298, pp. 562-564.

Kamburov, Atanas, Rachel Cavill, Timothy MD Ebbels, Ralf Herwig, and Hector C Keun

2011 "Integrated pathway-level analysis of transcriptomics and metabolomics data with IMPaLA", *Bioinformatics*, 27, 20, pp. 2917-2918.

Kanehisa, Minoru, Susumu Goto, Yoko Sato, Miho Furumichi, and Mao Tanabe

2011 "KEGG for integration and interpretation of large-scale molecular data sets", *Nucleic Acids Res.*, 40, D1, pp. D109-D114.

Kessler, Nikolas, Heiko Neuweger, Anja Bonte, Georg Langenkämper, Karsten Niehaus, Tim W Nattkemper, and Alexander Goesmann

2013 "MeltDB 2.0–advances of the metabolomics software system", *Bioinformatics*, 29, 19, pp. 2452-2459.

Khatri, Purvesh, Marina Sirota, and Atul J. Butte

2012 "Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges", *PLoS Comput. Biol.*, 8, 2.

Kirkegaard, Thomas and Marja Jäättelä

2009 "Lysosomal involvement in cell death and cancer", *Biochim. Biophys. Acta-Mol. Cell Res.*, 1793, 4, pp. 746-754.

Lehtonen, Heli J., Ignacio Blanco, Jose M. Piulats, Riitta Herva, Virpi Launonen, and Lauri A. Aaltonen

2007 "Conventional renal cancer in a patient with fumarate hydratase mutation", *Hum. Pathol.*, 38, 5, pp. 793-796.

Maceyka, Michael and Sarah Spiegel

2014 "Sphingolipid metabolites in inflammatory disease", *Nature*, 510, 7503, p. 58.

Madsen, Rasmus, Torbjörn Lundstedt, and Johan Trygg

2010 "Chemometrics in metabolomics – a review in human disease diagnosis", *Anal. Chim. Acta*, 659, 1, pp. 23-33.

Ni, Ying, Kevin M. Zbuk, Tammy Sadler, Attila Patocs, Glenn Lobo, Emily Edelman, Petra Platzer, Mohammed S. Orloff, Kristin A. Waite, and Charis Eng

2008 "Germline Mutations and Variants in the Succinate Dehydrogenase Genes in Cowden and Cowden-like Syndromes", *Am. J. Hum. Genet.*, 83, 2, pp. 261-268.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd

1999 *The PageRank citation ranking: Bringing order to the web.* Tech. rep., Stanford InfoLab.

Picart-Armada, Sergio, Francesc Fernández-Albert, Maria Vinaixa, Miguel Angel Rodriguez, Suvi Aivio, Travis H. Stracker, Oscar Yanes, and Alexandre Perera-Lluna

2017 "Null diffusion-based enrichment for metabolomics data", *PloS one*, 12, 12, e0189012.

Pithukpakorn, M, M-H Wei, O Toure, P J Steinbach, G M Glenn, B Zbar, W M Linehan, and J R Toro

2006 "Fumarate hydratase enzyme activity in lymphoblastoid cells and fibroblasts of individuals in families with hereditary leiomyomatosis and renal cell cancer." *J. Med. Genet.*, 43, 9, pp. 755-62.

Pollard, Patrick, Noel Wortham, and Ian Tomlinson

2003 "The TCA cycle and tumorigenesis: the examples of fumarate hydratase and succinate dehydrogenase", *Ann. Med.*, 35, 8, pp. 634-639.

Seo, Young-Jin, Stephen Alexander, and Bumsuk Hahm

2011 "Does cytokine signaling link sphingolipid metabolism to host defense and immunity against virus infections?", *Cytokine Growth Factor Rev.*, 22, 1, pp. 55-61.

Singh, Keshav K, Mohamed M Desouki, Renty B Franklin, and Leslie C Costello

2006 "Mitochondrial aconitase and citrate metabolism in malignant and nonmalignant human prostate tissues." *Mol. cancer*, 5, p. 14.

Vandin, Fabio, Eli Upfal, and Benjamin J Raphael

2011 "Algorithms for detecting significantly mutated pathways in cancer", *J. Comput. Biol.*, 18, 3, pp. 507-522.

Xia, Jianguo, Igor V Sinelnikov, Beomsoo Han, and David S Wishart

2015 "MetaboAnalyst 3.0 – making metabolomics more meaningful", *Nucleic Acids Res.*, 43, Web Server issue, W251-W257.

Yu, Guangchuang, Fei Li, Yide Qin, Xiaochen Bo, Yibo Wu, and Shengqi Wang

2010 "GOSemSim: an R package for measuring semantic similarity among GO terms and gene products", *Bioinformatics*, 26, 7, pp. 976-978.

Yu, Guangchuang, Li-Gen Wang, Yanyan Han, and Qing-Yu He

2012 "clusterProfiler: an R package for comparing biological themes among gene clusters", *OMICS*, 16, 5, pp. 284-287.

2014 "Distinct metabolic responses of an ovarian cancer stem cell line", *BMC Syst Biol*, 8, 1, p. 134.

Ziarrusta, Haizea, Leire Mijangos, Sergio Picart-Armada, Mireia Irazola, Alexandre Perera-Lluna, Aresatz Usobiaga, Ailette Prieto, Nestor Etxebarria, Maitane Olivares, and Olatz Zuloaga

2018 "Non-targeted metabolomics reveals alterations in liver and plasma of gilt-head bream exposed to oxybenzone", *Chemosphere*, 211, pp. 624-631.

8 DISEASE GENE IDENTIFICATION

BENCHMARKING NETWORK PROPAGATION METHODS FOR DISEASE GENE IDENTIFICATION

In-silico identification of potential target genes for disease is an essential aspect of drug target discovery. Recent studies suggest that successful targets can be found through by leveraging genetic, genomic and protein interaction information.

Here, we systematically tested the ability of 12 varied algorithms, based on network propagation, to identify genes that have been targeted by any drug, on gene-disease data from 22 common non-cancerous diseases in OpenTargets. We considered two biological networks, six performance metrics and compared two types of input gene-disease association scores. The impact of the design factors in performance was quantified through additive explanatory models. Standard cross-validation led to over-optimistic performance estimates due to the presence of protein complexes. In order to obtain realistic estimates, we introduced two novel protein complex-aware cross-validation schemes. When seeding biological networks with known drug targets, machine learning and diffusion-based methods found around 2-4 true targets within the top 20 suggestions. Seeding the networks with genes associated to disease by genetics decreased performance below 1 true hit on average. The use of a larger network, although noisier, improved overall performance.

We conclude that diffusion-based prioritisers and machine learning applied to diffusion-based features are suited for drug discovery in practice and improve over simpler neighbour-voting methods. We also demonstrate the large impact of choosing an adequate validation strategy and the definition of seed disease genes.

8.1 AUTHOR SUMMARY

The use of biological network data has proven its effectiveness in many areas from computational biology. Networks consist of nodes, usually genes or proteins, and edges that connect pairs of nodes, representing information such as physical interactions, regulatory roles or co-occurrence. In order to find new candidate nodes for a given biological property, the so-called network propagation algorithms start from the set of known nodes with that

This chapter is a postprint of the following journal article: Picart-Armada, Sergio, Steven J. Barrett, David R. Willé, Alexandre Perera-Lluna, Alex Gutteridge, and Benoit H. Dessailly. "Benchmarking network propagation methods for disease gene identification". *PLoS computational biology 15, no. 9 (2019): e1007276.*

property and leverage the connections from the biological network to make predictions. Here, we assess the performance of several network propagation algorithms to find sensible gene targets for 22 common non-cancerous diseases, i.e. those that have been found promising enough to start the clinical trials with any compound. We focus on obtaining performance metrics that reflect a practical scenario in drug development where only a small set of genes can be essayed. We found that the presence of protein complexes biased the performance estimates, leading to over-optimistic conclusions, and introduced two novel strategies to address it. Our results support that network propagation is still a viable approach to find drug targets, but that special care needs to be put on the validation strategy. Algorithms benefitted from the use of a larger -although noisier- network and of direct evidence data, rather than indirect genetic associations to disease.

8.2 INTRODUCTION

The pharmaceutical industry faces considerable challenges in the efficiency of commercial drug research and development (Scannell et al., 2012) and in particular in improving its ability to identify future successful drug targets.

It has been suggested that using genetic association information is one of the best ways to identify such drug targets (Nelson et al., 2015). In recent years, a large number of highly powered GWAS studies have been published for numerous common traits (for example, (Schizophrenia Working Group of the Psychiatric Genomics Consortium, 2014; Verstockt et al., 2018)) and have yielded many candidate genes. Further potential targets can be identified by adding contextual data to the genetic associations, such as genes involved in similar biological processes (Boyle et al., 2017; Jia and Zhao, 2013). Biological networks and biological pathways can be used as a source of contextual data.

Biological networks are widely used in bioinformatics and can be constructed from multiple data sources, ranging from macromolecular interaction data collected from the literature (Orchard et al., 2012) to correlation of expression in transcriptomics or proteomics samples of interest (Langfelder and Horvath, 2008). A large number of interaction network resources have been made available over the years, many of which are now in the public domain, combining thousands of interactions in a single location (Razick et al., 2008; Szklarczyk, Morris, et al., 2016). They are based on three different fundamental types of data: (1) data-driven networks such as those built by WGCNA (Langfelder and Horvath, 2008) for co-expression; (2) interactions extracted from the literature using a human curation process as exemplified by IntAct (Kerrien et al., 2011) or BioGRID (Chatr-Aryamontri et al., 2017); and (3) interactions extracted from the literature using text mining approaches (Al-Aamri et al., 2017).

On the other hand, a plethora of network analysis algorithms are available for extracting useful information from such large biological networks in a variety of contexts. Algorithms range in complexity from simple first-neighbour approaches, where the direct neighbours of a gene of interest are assumed to be implicated in similar processes (Piovesan et al., 2015), to ma-

chine learning (ML) algorithms designed to learn from the features of the network to make more useful biological predictions (Re et al., 2012).

One broad family of network analysis algorithms are the so-called Network Propagation approaches (Cowen et al., 2017), used in contexts such as protein function prediction (Sharan et al., 2007), disease gene identification (Cowen et al., 2017) and cancer gene mutation identification (Leiserson et al., 2014). In this paper, we perform a systematic review of the usefulness of network analysis methods for the purpose of identification of disease genes. As further explained in Methods, we define our test set of disease genes as genes for which the relationship with a disease was sufficiently clear to justify the start of a drug development programme. Claims that such methods are helpful in that context have been made on numerous occasions but a comprehensive validation study is lacking. One major challenge in doing such a study is to define a list of true disease genes for this purpose.

To address this, the Open Targets collaboration between pharmaceutical companies and public institutions collects information on known drug targets to help identify new ones (Koscielny et al., 2016). A dedicated internet platform provides a free-to-use accessible resource summarising known data on gene-disease relationships from a number of data sources, like known released drugs and genetic associations from GWAS (Koscielny et al., 2016).

The purpose of this work is to quantify the performance of network propagation methods to prioritise novel drug targets, using various networks and validation schemes, and aiming at a faithful reflection of a realistic drug development scenario. We are not predicting gene targets for specific drugs, but rather sensible genes to target for a specific disease. Data on actual compounds targeting a gene is ignored: as long as the gene has been targeted by one or more compounds reaching the clinical trials, it is considered a sensible drug target. We select a number of network propagation approaches that are representative of several classes of algorithms, and test their ability to recover known target genes for several non-cancerous diseases by cross-validation.

We benchmark multiple definitions of disease genes as input for the prioritisers, computational methods, biological networks, validation schemes and performance metrics. We account for all possible combinations of such factors and derive guidelines for future disease target identification studies. The code and data that support our conclusions can be found in `https://github.com/b2slab/genedise`.

8.3 RESULTS

8.3.1 Benchmark framework

Our general approach, summarised in Fig 28, consisted in using a biological network and a list of genes with prior disease-association scores as input to a network propagation approach. We tested some variations of classical network propagation `-ppr`, `raw`, `gm`, `mc` and `z-` which differ on the directedness of the propagation, the input weights and the presence of a statistical normalisation of the scores. Semi-supervised methods included,

under the positive-unlabelled learning framework: `knn` and `wsld`. Both work directly on a graph kernel, closely related to network propagation. Supervised methods were also considered: `COSNet`, which regards the network as an artificial neural network, `bagsvm`, a bagging Support Vector Machine on a graph kernel, and `rf` and `svm`, which apply either Random Forest or a Support Vector Machine to network-based features that encode propagation states in a lower dimensionality. The `EGAD` method, based on neighbour voting, served as a baseline prioritiser. Three input-naïve baselines were included: `pr` and `randomraw`, both biased by the network topology, and `random`, a purely random prioritiser.

We used three cross-validation schemes -two take into account protein complexes- in which some of the prior disease-association scores are hidden. The desired output was a new ranking of genes in terms of their association scores to the disease. Such ranking was compared to the known target genes in the validation fold using several performance metrics. Given the amount of design factors and comparisons, the metrics were analysed through explanatory additive models (see Methods). Specifically, regression models explained the performance metrics (dependent variable) as a function of the prediction method, the cross-validation scheme, the network and the disease (regressors). This enabled a formal analysis of the impact of each factor on overall performance while correcting for the others. Alternatively, we provide plots on the raw metrics in S1 Appendix, stratified by method in Figures J and K or by disease in Figures L and M.

We considered 2 metrics (AUROC and top 20 hits) and 2 input types (known drug target genes and genetically associated genes), resulting in a total of 4 combinations, each described through an additive main effect model. Another 4 metrics were explored and can be found in Figure Q and Tables F and G in S1 Appendix.

Interaction terms within the explanatory models were explored, but they did not provide any added value for the extra complexity, see Figure S in S1 Appendix.

8.3.2 Performance using known drug targets as input

Fig 29 describes the additive models for AUROC and top 20 hits, and using known drug targets as input. Note that the disease was included as a regressor in the explanatory models for further discussion. This was possible given our definition of drug targets: methods had to predict whether a gene has been targeted by any drug for a particular disease, implying that metrics were available at the disease level.

Fig 30 contains their predictions for each method, network and cross-validation scheme with 95% confidence intervals, averaged over diseases. The models are complex and we therefore review each main effect separately.

For interpretability within real scenarios, the top 20 hits is regarded as the reference metric in the main body. The standard AUROC (quasi-binomial) clearly led to different conclusions and is kept throughout the results section for comparison. The remaining metrics (AUPRC, pAUROC 5%, pAUROC

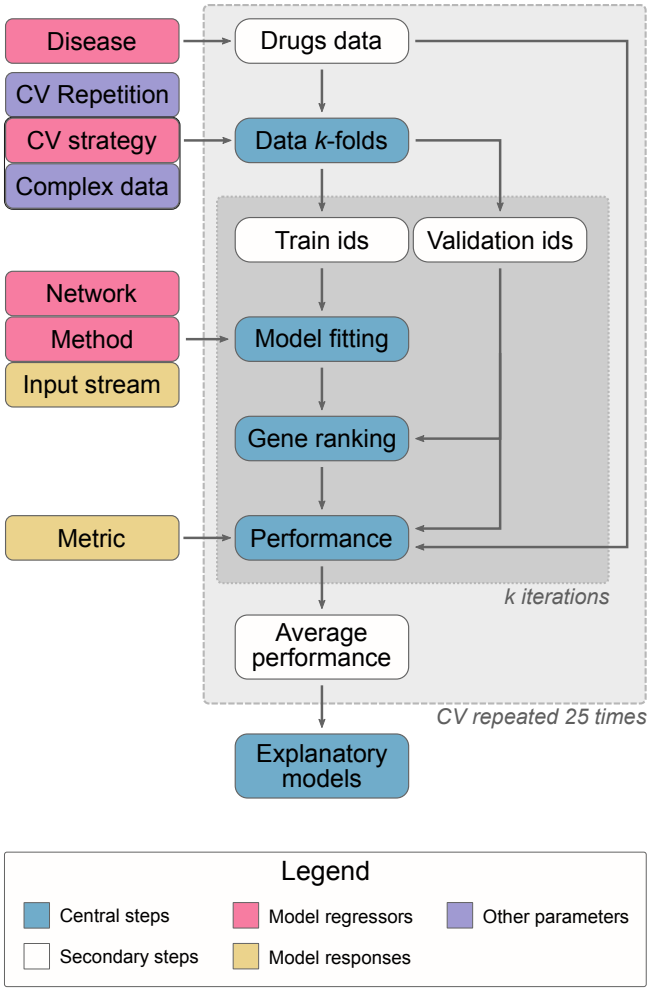


Figure 28: Benchmark overview. This work describes six performance metrics using two input streams (genetic association and drug-based genes) to predict drug target-based genes for 22 common diseases. 3-fold cross-validation (CV), repeated 25 times, was run under three CV strategies. The gene identifiers in each fold are determined using only the drugs data, regardless of the input. Two validation strategies are complex-aware and therefore needed this data to define the splits. 15 methods based on network propagation (including 4 baselines) were evaluated, using two networks with different properties, by modelling their performance -averaged on every CV round- with explanatory models. After obtaining the performance metrics, the explanatory models allowed hypothesis testing and a direct performance comparison between diseases, CV strategies, networks and methods, by setting them as the independent variables of the models. The latter is depicted by pink (independent variables) and yellow (dependent variable) blocks, and should not be confused with the "model fitting" block, which refers to the network propagation prioritisers.

10% and top 100 hits) result in similar method prioritisations as top 20 hits, see Figure Q in S1 Appendix. Detailed models can be found in S1 Appendix, indexed by Tables F and G.

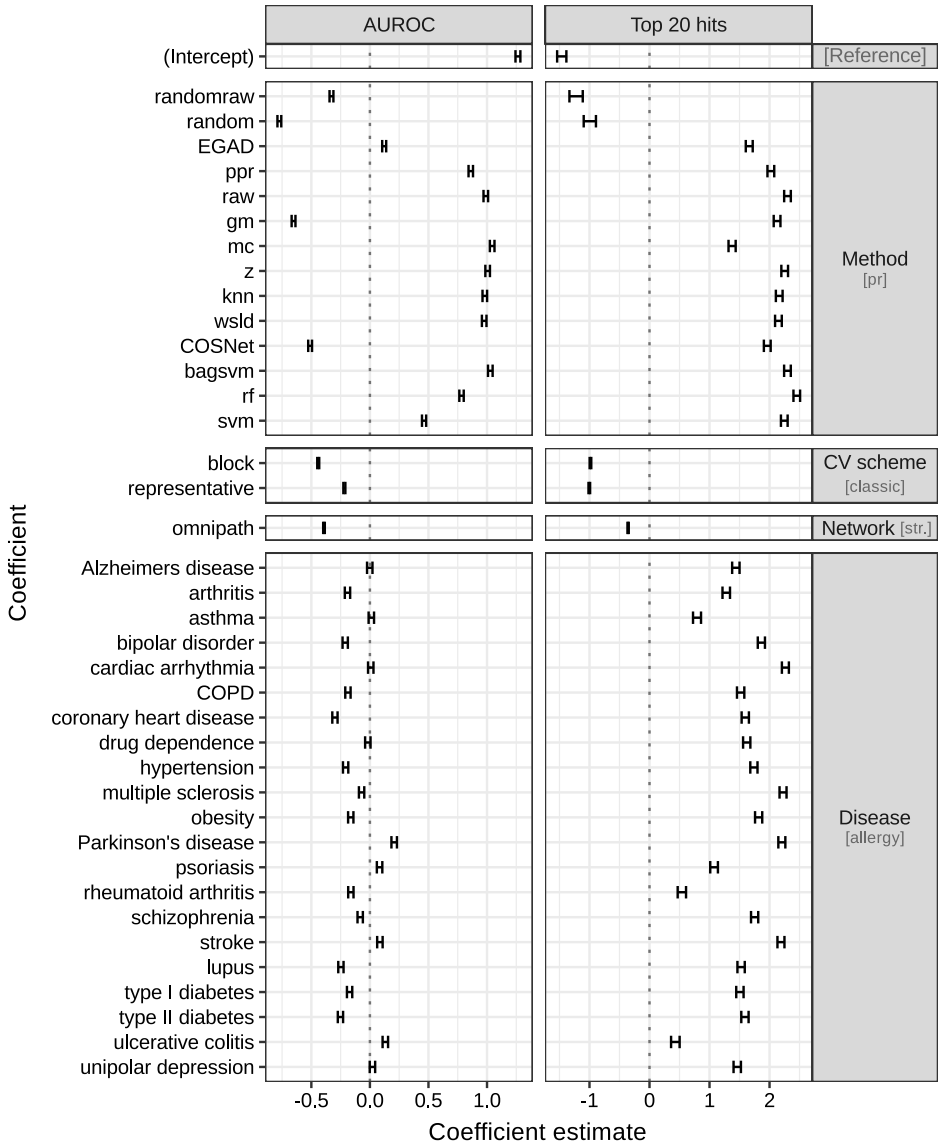


Figure 29: Additive explanatory models for AUROC and top 20 hits. Each column corresponds to a different model, whereas each row depicts the 95% confidence interval for each model coefficient. Rows are grouped by the categorical variable they belong to: method, cv scheme, network and disease. Each variable has a **reference level**, implicit in the intercept and specified in brackets: `pr` method, **classic** validation scheme, **STRING** network and **allergy**. Positive estimates improve performance over the reference levels, whereas negative ones reduce it. For example, the data suggest that method `rf` performs better than the baseline using both metrics, and is the preferred method using the top 20 hits. Switching from STRING to the OmniPath network, or from classic to block or representative cross-validation, has a negative effect on both performance metrics. Specific model estimates and confidence intervals can be found in Tables H and I in S1 Appendix.

Comparing cross-validation schemes

Whether protein complexes were properly taken into account when performing the cross-validation (see Methods) stood out as a key influence on

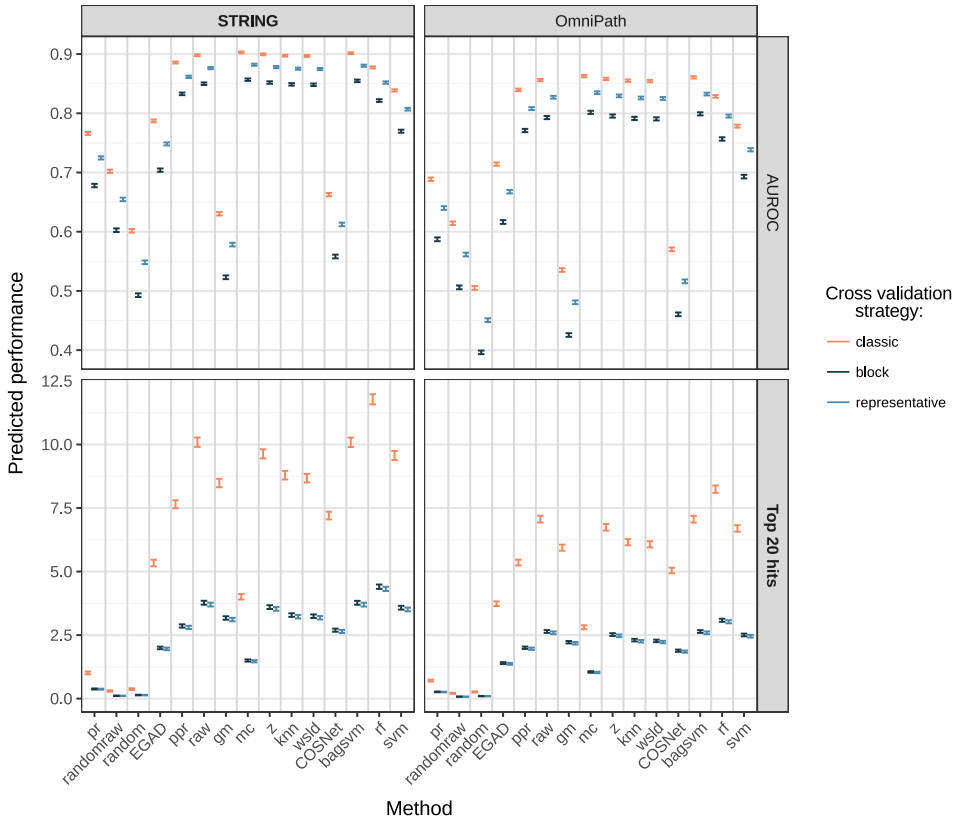


Figure 30: Performance predicted for AUROC and top 20 hits through the additive explanatory models. Each row corresponds to a different model and error bars depicts the 95% confidence interval of the additive model prediction, averaging over diseases. In bold, the main network (STRING) and metrics (AUPRC, top 20 hits). The exact values can be found in Table I in S1 Appendix.

the quality of predictions: there was a dramatic reduction in performance for most methods when using a complex-aware cross-validation strategy. For instance, method `rf` applied on the STRING network dropped from almost 12 correct hits in the top 20 predicted disease genes when using our *classic* cross-validation scheme down to fewer than 4.5 when using either of our complex-aware cross-validation schemes. Likewise, Table E in S1 Appendix ratifies that only the *classic* cross-validation splits complexes. A recent study raised analogous concerns on estimating the performance of supervised methods when learning gene regulatory networks (Tabe-Bordbar et al., 2018). Random cross-validation would lead to overly optimistic performances when predicting new regulatory contexts, requiring to control for the distinctness between the training and the testing data. This confirms that other areas in computational biology may benefit from adjusted cross-validation strategies.

Our data suggests that the performance drop when choosing the appropriate validation strategy is comparable to the performance gap of competitive methods versus a simple neighbour-voting baseline EGAD (see Fig 29). This highlights the importance of carefully controlling for this bias when estimating the performance of target gene prediction using network propagation. Overall, the *classic* cross-validation scheme gave biased estimates

in our dataset, whereas our *block* and *representative* cross-validation schemes had similar effects on the prediction performance. Method ranking was independent of the cross-validation choice thanks to the use of an additive model. Since both the *block* and *representative* schemes led to the same conclusions, we chose to focus on results from the block scheme in the rest of this study.

Comparing networks

We found that using STRING as opposite to OmniPath improved overall performance of disease gene prediction methods. Our models for top 20 hits quantified this effect as noticeable although less important than that of the cross-validation strategy. For reference, method `rf` obtains about 3 true hits under both complex-aware strategies in OmniPath. It has been previously shown that the positive effect on predictive power of having more interactions and coverage in a network can outweigh the negative effect of increased number of false positive interactions (Huang et al., 2018), which is in line with our findings. The authors also report STRING among the best resources to discover disease genes, which is analogous to our findings on the drug targets.

We focus on the STRING results in the rest of the text.

Comparing methods

Having identified the optimal cross-validation scheme and network for our benchmark in the previous sections, we quantitatively compared the performance of the different methods.

First, network topology alone had a slight predictive power, as method `pr` (PageRank approach that ignores the input gene scores) showed better performance than the `random` baseline under all the metrics. The randomised diffusion `randomraw` lied between `random` and `pr` in performance, depending on the metric. Both facts support the existence of an inherent network topology-related bias among target genes that benefits diffusion-based methods. This finding is compatible with the existence of a reduced set of critical edges that account for most of the predictive power in GBA methods (Gillis and Pavlidis, 2012), as highly connected genes are more likely to be involved in those.

Second, the basic GBA approach from `EGAD` had an advantage over the input-naïve baselines `pr`, `randomraw` and `random`. It also outperformed prioritising genes using other Open Targets data stream scores such as genes associated to disease from pathways or from the literature (see Table S in S1 Appendix).

Most diffusion-based and ML-based methods outperformed `EGAD`. To formally test the differences between methods, we carried a Tukey's multiple comparison test on the model coefficients (Fig 31) as implemented in the R package multcomp (Hothorn et al., 2008). Although such differences were in most cases statistically significant after multiplicity adjustment, their actual effect size or magnitude can be modest in practice, see Figs 30 and 32. Results from top 20 hits suggest using `rf` for the best performance followed by, in order: `raw` and `bagsvm`, `z` and `svm` (main models panel in Fig 32).

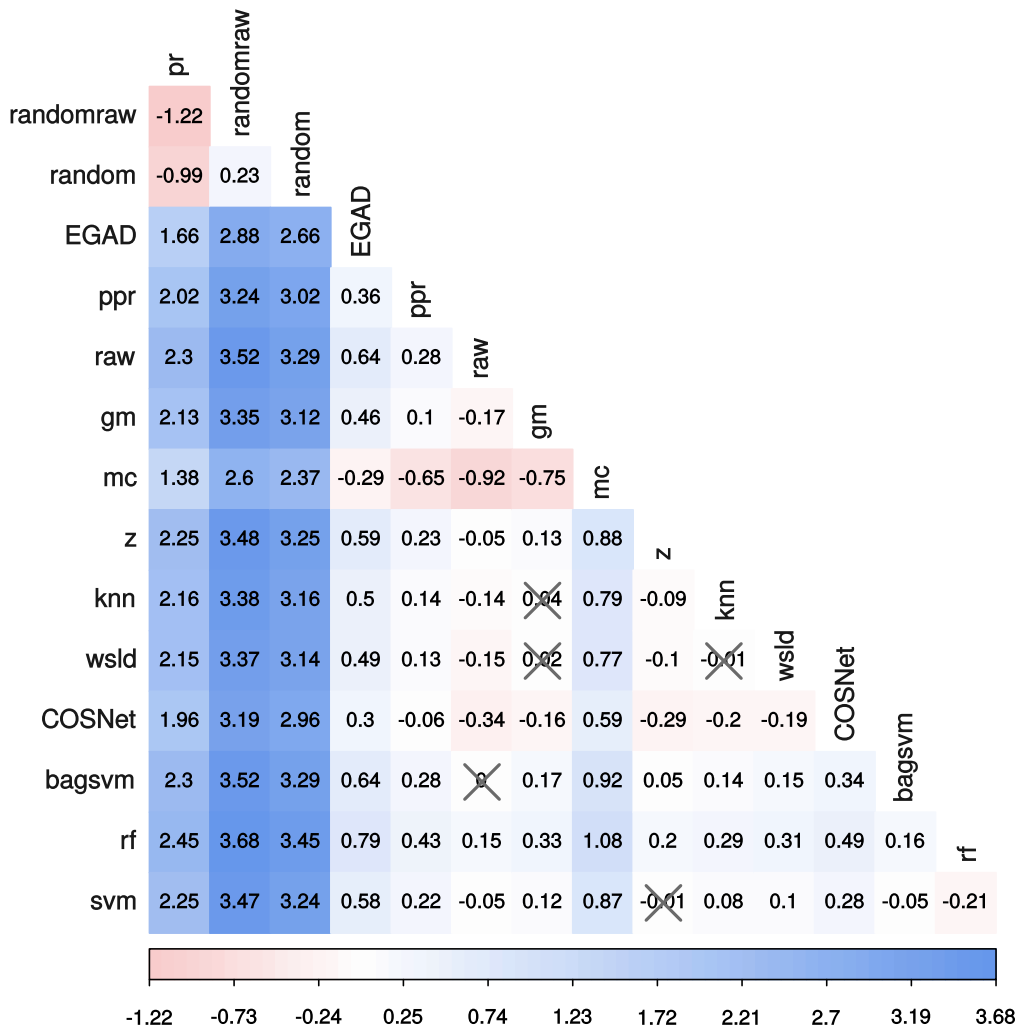


Figure 31: **Pairwise contrasts on top 20 hits predicted by the quasipoisson explanatory model**. Differences are expressed in the model space. Most of the pairwise differences are significant (Tukey's test, p <0.05) – non-significant differences have been crossed out.

The ranking of methods was similar when using the metrics AUPRC, pAUROC and top k hits (see Figure Q in S1 Appendix) and is only intended to be a general reference, given the impact of the problem definition, cross-validation scheme and the network choice.

With AUROC on the other hand, `rf` lost its edge whilst most diffusion-based and ML-based methods appeared technically tied. Despite its theoretical basis, interpretability and widespread use in similar benchmarks, these results support the assertion that AUROC is a sub-optimal choice in drug discovery practical scenarios.

Fig 33 further shows how the different methods compare with one another. Distances between each pair of method in terms of their top 100 novel predictions were represented graphically. We observe that the supervised bagged Support Vector Machine approach (`bagsvm`) behaves similarly to the simple diffusion approach (`raw`), reflecting the fact that they use the same kernel. We also observe that diffusion approaches do not necessarily produce sim-

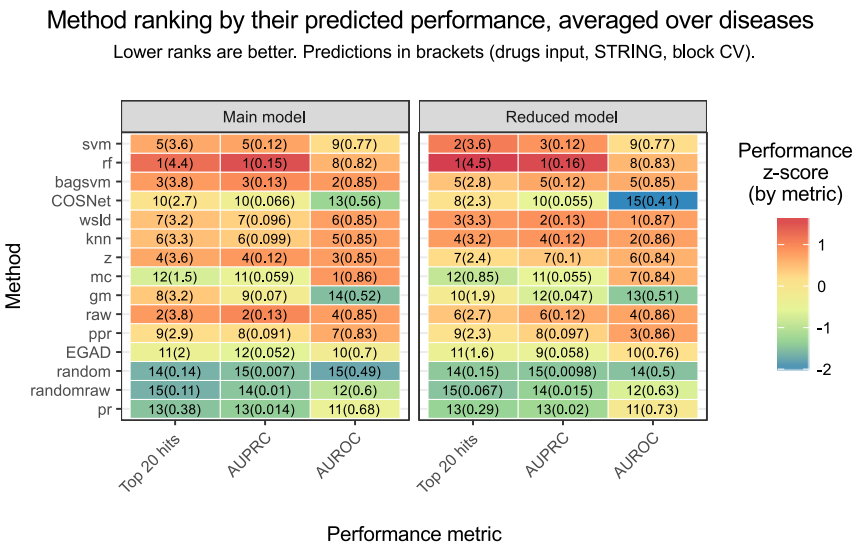


Figure 32: Ranking of all the methods. Ranking according to the predictions of the main explanatory models (left) and the reduced explanatory models within the STRING network and block cross-validation (right), in both cases on the drugs input and averaging over diseases. The main models serve as a global description of the metrics, whereas the reduced models are specific to the scenario of most interest. A column-wise z-score on the predicted mean is depicted, in order to illustrate the magnitude of the difference. Note how the top 20 hits and the AUPRC metrics lead to similar conclusions, as opposed to AUROC.

ilar results; for instance, `raw` and `z`. Besides, methods `EGAD` (arguably one of the simplest) and `COSNet` (arguably one of the most complex) seemed to result in similar predictions. Fully supervised and semi-supervised approaches largely group in the top right hand quadrant of the STRING plot away from diffusion methods, possibly showing better learning capability with the larger network.

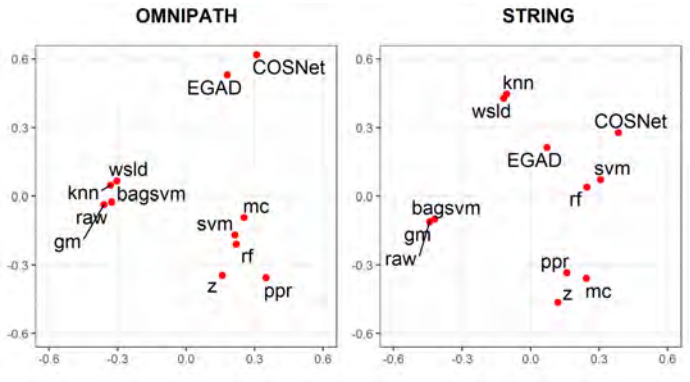


Figure 33: Multi-view MDS plot displaying the preserved Spearman's footrule distances between methods. The differential ranking of their top 100 novel predictions using known drug target inputs are taken into account across all 22 diseases. Results are shown separately for the 2 networks considered in this study. Seed genes are excluded from the distance calculations.

When comparing overall performances shown in Fig 32 with the prediction differences from the MDS plot (Fig 33), the best methods owed their performance to different reasons as they do not occur within the same region of the plot (e.g. `rf` and `raw`). MDS plots on the eight possible combinations of network, input type and inclusion of seed genes are displayed in Figures O and P in S1 Appendix.

Focusing only on the STRING network and the block validation scheme, we fitted six additive explanatory models, called the reduced models, to model the six metrics for the drugs data input as a function of the method and the disease (see Table G in S1 Appendix). Methods were prioritised according to their main effects (Fig 32). The reduced models better described this particular scenario, as they were not forced to fit the trends in all networks and validation schemes in an additive way. Considering the top 20 hits, `rf` and `svm` were the optimal choices, followed by `wsld` and `knn`.

Comparing diseases

The top 20 hits model in Fig 29 shows that allergy (the figure's baseline reference), ulcerative colitis and rheumatoid arthritis (group I) are the diseases for which prediction of target genes was worst, whereas cardiac arrhythmia, Parkinson's disease, stroke and multiple sclerosis (group II) are those for which it was best. As shown in Fig 34, group I diseases had fewer known target genes and lower modularity compared to group II diseases.

Prediction methods worked better when more known target genes were available as input in the network, with two possible underlying reasons: the greater data availability to train the methods, and the natural bias of top 20 hits towards datasets with more positives. Likewise, a stronger modularity within target genes justifies the guilt-by-association principle and led to better performances. In turn, the number of genes and the modularity were positively correlated, see Figure N in S1 Appendix.

8.3.3 Performance using genetic associations as input

Using genetically associated genes as input to a prediction approach to find known drug targets mimicked a realistic scenario where novel genetic associations are screened as potential targets. However, inferring known drug targets through the indirect genetic evidence posed problems to prediction strategies, especially those based on machine learning. Learning is done using one class of genes in order to predict genes that belong to another class, and the learning space suffers from intrinsic uncertainties in the genetic associations to disease. Both classes are inherently different: certain genes can be difficult to target, and a gene does not require to have been formally associated genetically to a disease to become a valid target.

Consequently, we observed a major performance drop on all the prioritisation methods: using any network and cross-validation scheme, the predicted top 20 hits were practically bounded by 1. This was more pronounced on supervised machine learning-focused strategies, as `rf` and `svm` lost their edge on diffusion-based strategies. The fact that the genetic associations of the

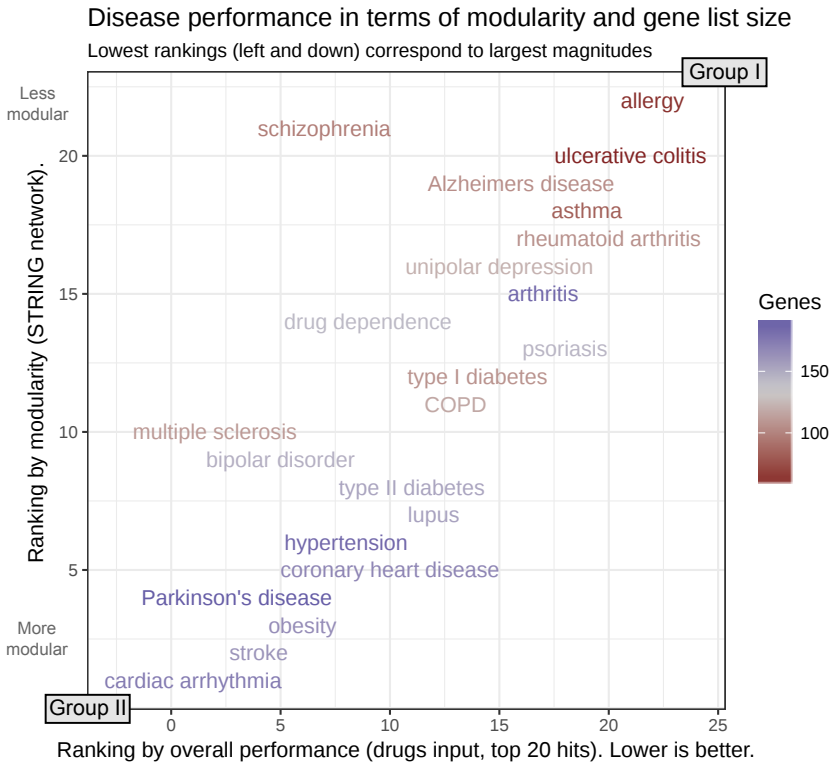


Figure 34: Disease performance in terms of input size and modularity. Disease performance ranked by the number of known target genes and their modularity (obtained using the igraph package, see Figure F in S1 Appendix). Modularity is a measure of the tendency of known target genes to form modules or clusters in the network. Diseases have been ranked using their explanatory model coefficient from the top 20 hits metric with known drug targets as input (x axis) and their modularity (y axis). As discussed in the text, best predicted diseases tend to have longer gene lists and be highly modular.

validation fold were hidden further hindered the predictions and can be a cause of our pessimistic performance estimates.

Comparing cross-validation schemes

For reference, we also ran all three cross-validation schemes on the genetic data to quantify and account for complex-related bias. The models confirm that, contrary to the drugs-related input, the differences between the results for the different cross-validation schemes were rather modest. For example, method `raw` with the STRING network attains 0.59-0.64, 0.50-0.54 and 0.37-0.40 hits in the top 20 under the classical, block and representative cross-validation strategies. The slightly larger negative effect on top 20 hits observed with the representative scheme is expected because the number of positives that act as validation decreased and this metric is biased by the class imbalance. The agreement between method ranking using AUPRC and top 20 hits was less consistent, possibly due to the performance drop, whilst AUROC yielded a noticeably different ranking again. Further data can be found in Tables O and P in S1 Appendix.

Comparing networks

The change in performance for using the OmniPath network instead of the filtered STRING network was also limited. For AUROC the effect was negative, whereas for the top 20 hits metric the performance improved. Method `raw` changed from 0.50-0.54 top 20 hits in STRING to 0.61-0.66 in OmniPath under the block validation strategy.

Comparing methods

To be consistent with the drugs section, we take as reference the block cross-validation strategy and the STRING network.

The baseline approach `pr` that effectively makes use of the network topology alone proved difficult to improve upon, with 0.43-0.47 expected true hits in the top 20. Methods `raw` and `rf` respectively achieved 0.50-0.54 and 0.23-0.26 – although significant, the difference in practice would be minimal. The best performing method was `mc` with 0.65-0.7 hits. All the performance estimates can be found in Table P in S1 Appendix. To give an idea of the effort that would be required in a realistic setting to find novel targets, the number of correct hits in the top 100 hits was 3.29-3.45 with the best performing method (in this case, `ppr`), against 2.25-2.38 of `pr`.

Two main conclusions can be drawn from these results. First, the network topology baseline retained some predictive power upon which most diffusion-based methods, as well as machine-learning approaches `COSNet` and `bagsvm`, only managed to add minor improvements, if any. Second, drug targets could still be found by combining network analysis and genes with genetic associations to disease, but with a substantially lower performance and with a marginal gain compared to a baseline approach that would only use the network topology to find targets (e.g. by screening the most connected genes in the network).

It is worth noting that gene-disease genetic association scores themselves have drawbacks and that better prediction accuracy could result as genetic association data improves.

8.4 DISCUSSION

We performed an extensive analysis of the ability of several approaches based on network propagation to identify novel non-cancerous disease target genes. We explored the effect of various choices in factors including the biological network, the definition of disease genes acting as seeds, and the statistical framework being used to evaluate methods performance. We show that carefully choosing an appropriate cross-validation framework and suitable performance metric has an important effect in evaluating the utility of these methods.

Our main conclusion is that network propagation seems effective for drug target discovery, reflecting the fact that drug targets tend to cluster within the network. This may be due to the fact that the scientific community has so

far been focusing on testing the same proven mechanisms, which can induce some ascertainment bias

In a strict cross-validation setting, we found that even the most basic guilt-by-association method was useful, with ~2 correct hits in its top 20 predictions, compared to ~0.1 when using a random ranking. The best diffusion based algorithm improved that figure to ~3.75, and the best overall performing method was a random forest classifier on network-based features (~4.4 hits). Leading approaches can be notably different in terms of their top predictions, suggesting potential complementarity. We found a better performance when using a network with more coverage at the expense of more false positive interactions. In a more conservative network, random forest performance dropped to ~3.1 hits. Comparing performance on different diseases shows that the more known target genes, and the more clustered these are in the network, the better the performance of network propagation approaches for finding novel targets for it.

We also explored the prediction of known drug target genes by seeding the network with an indirect data stream, in particular, genetic association data. Here, the best performing methods were diffusion-based and presented a statistically significant, but marginal, improvement over approaches that only look at network centrality.

We conclude that network propagation methods can help identify novel targets for disease, but that the choice of the input network and the seed scores on the genes needs careful consideration. Based on our approach and endorsed benchmarks, we recommend the use of methods employing representations of diffusion-based information (the MashUp network-based features and the diffusion kernels), namely random forest, the support vector machine variants, and raw diffusion algorithms for optimal results.

8.5 MATERIALS AND METHODS

8.5.1 Selection of methods for investigation

Network propagation algorithms were selected for validation based on the following criteria:

1. Published in a peer-reviewed journal, with evidence of improved performance in gene disease prediction relative to contenders.
2. Implemented as a well documented open source package, that is efficient, robust and usable within a batch testing framework.
3. Directly applicable for gene disease identification from a single gene or protein interaction network, without requiring fundamental changes to the approach or additional annotation information.
4. Capable of outputting a ranked list of individual genes (as opposed to gene modules, for example).

In addition, we selected methods that were representative of a diverse panel of algorithms, including diffusion variants, supervised learning on

features derived from network propagation, and a number of baseline approaches (see Table 12).

8.5.2 Testing framework, algorithms and parameterisation

All tests and batch runs were set-up and conducted using the R statistical programming language (R Core Team, 2016). When no R package was available, the methodology was re-implemented, building upon existing R packages whenever possible. Standard R machine learning libraries were used to train the support vector machine and random forest classifiers. Only the MashUp algorithm (Cho et al., 2016) required feature generation outside of the R environment, using the Matlab code from their publication. Further details on the methods implementation can be found in S1 Appendix, section "Method details".

EGAD (Ballouz et al., 2017), a pure neighbour-voting approach, was used here as a baseline comparator.

Diffusion (propagation) methods are central in this study. We used the random walk-based personalised PageRank (Page et al., 1999), previously used in similar tasks (Jiang et al., 2017), as implemented in igraph (Csardi and Nepusz, 2006). The remaining diffusion-based methods were run on top of the regularised Laplacian kernel (Smola and Kondor, 2003), computed through diffuStats (Picart-Armada, Thompson, et al., 2017). We included the classical diffusion `raw`, a weighted approach version `gm` that assigns a bias term to the unlabelled nodes, and two statistically normalised scores (`mc` and `z`), as implemented in diffuStats. The normalised scores adjust for systematic biases in the diffusion scores that relate to the graph topology, in order to provide a more uniform ranking. In the scope of positive-unlabelled learning (Elkan and Noto, 2008; Yang et al., 2012), we included the kernelised scores `knn` and the linear decayed `wsld` from RANKS (Valentini, Paccanaro, et al., 2014). `knn` computes each gene score based on the k-nearest positive examples, using the graph kernel to compute the distances. Conversely, `wsld` uses all the kernel similarities to the positive examples, but applies a decaying factor to downweight the furthest positives. Closing this category, we implemented the bagging Support Vector Machine approach from ProDiGe1 (Mordelet and Vert, 2011), here `bagsvm`, which trains directly on the graph kernel to find the optimal hyperplane separating positive and negative genes.

Purer ML-based methods were also included. On one hand, network-based features were generated using MashUp (Cho et al., 2016) and two classical classifiers were fitted to them, based on caret (Kunn, 2008) and mlr (Bischl et al., 2016). These are `svm`, the Support Vector Machine as implemented in kernlab (Karatzoglou et al., 2004), and `rf`, the Random Forest found in the randomForest package (Liaw and Wiener, 2002). On the other hand, we tried the parametric Hopfield recurrent neural network classifier in the COSNet R package (Bertoni et al., 2011; Frasca et al., 2013). `COSNet` estimates network parameters on the sub-network containing the labelled nodes, extends them to the sub-network containing the unlabelled ones and then predicts the labels.

Table 12: List of methods included in this benchmark. Method identifiers are shortened method names used throughout the text. Other columns are self-explanatory.

Method Identifier	Method Name	Method Class	Implementation	Reference
`pr`	PageRank with a uniform prior	Baseline	igraph (Bioconductor (Gentleman et al., 2004; Huber et al., 2015) package)	(Page et al., 1999)
`random`	Random	Baseline	R	(see text)
`randomraw`	Random Raw	Baseline	R	(see text)
`EGAD`	Extending Guilt by Association' by Degree	Baseline	EGAD (Bioconductor package)	(Ballouz et al., 2017)
`ppr`	Personalized PageRank	Diffusion	igraph (R package)	(Jiang et al., 2017)
`raw`	Raw Diffusion	Diffusion	diffuStats (Bioconductor package)	(Vandin et al., 2011)
`gm`	GeneMania-based weights	Diffusion	diffuStats (Bioconductor package)	(Mostafavi et al., 2008)
`mc`	Monte Carlo normalised scores	Diffusion	diffuStats (Bioconductor package)	(Picart-Armada, Fernández-Albert, et al., 2017)
`z`	Z-scores	Diffusion	diffuStats (Bioconductor package)	(Picart-Armada, Fernández-Albert, et al., 2017)
`knn`	K nearest neighbours	Semi-supervised learning	RANKS (R package)	(Valentini, Armano, et al., 2016)
`wsld`	Weighted Sum with Linear Decay	Semi-supervised learning	RANKS (R package)	(Valentini, Armano, et al., 2016)
`COSNet`	COst Sensitive neural Network	Supervised learning	COSNet (R package)	(Frasca et al., 2013)
`bagsvm`	Bagging SVM (based on ProDiGe1)	Supervised learning	kernlab (R package)	(Mordelet and Vert, 2011)
`rf`	Random Forest	Supervised learning	randomForest (R package) + Matlab (features)	(Cho et al., 2016)
`svm`	Support Vector Machine	Supervised learning	kernlab (R package) + Matlab (features)	(Cho et al., 2016)

Finally, we defined three naive baseline methods: (1) `pr`, a PageRank with a uniform prior, where input scores on the genes are ignored; (2) `randomraw`, which applies the `raw` diffusion approach to randomly permuted input scores on the genes; and (3) `random`, a uniform re-ranking of input genes without any network propagation. The inclusion of `pr` and `randomraw` allowed us to quantify the predictive power of the network topology alone, without any consideration for the input scores on the genes.

8.5.3 Biological networks

The biological network used in the validation is of critical importance as current network resources contain both false positive and false negative interactions, possibly affecting subsequent predictions (Huang et al., 2018).

Here, we used two human networks with different general properties, one more likely to contain false positive interactions (STRING (Szklarczyk, Franceschini, et al., 2014)), and another more conservative (OmniPath (Türei et al., 2016)), to test the effect of the network itself on network propagation performance. We further filtered STRING (Szklarczyk, Franceschini, et al., 2014) to retain only a subset of interactions. Having tested several filters, we settled upon high-confidence interactions (combined score > 700) with some evidence from the "Experiments" or "Databases" data sources (see Table B in S1 Appendix). Applying these filters and taking the largest connected component resulted in a connected network of 11,748 nodes and 236,963 edges. Edges were assigned weights between 0 and 1 by rescaling the STRING combined score.

We did not filter the OmniPath network (Türei et al., 2016). After removing duplicated edges and taking the largest connected component, the OmniPath network contained 8,580 nodes and 42,145 unweighted edges.

8.5.4 Disease gene data

We used the Open Targets platform (Koscielny et al., 2016) to select known disease-related genes. In this analysis we defined positive genes as those reported in Open Targets as being the target of any known drug against the disease of interest, from which all the metrics were computed. We decided to use drug targets, including unsuccessful ones, as proxies for disease genes on the basis that genes for which a drug programme has been started, generally with significant investment, are most likely to have strong evidence linking them to the disease. We therefore regard them as a set of high-confidence true positive disease genes. This choice means we potentially miss genes that have strong genetic associations to the disease but are not druggable. In other words, we focus on limiting false positives in our reference set of positives, at the expense of having more false negatives in our set of negatives. Alternatively, genes with a genetic association of sufficient confidence with the disease were also used as an input data stream, in order to assess the predictive power of an indirect source of evidence. Associations were binarised: any non-zero drugs-related association was considered positive, implying that the methods would predict genes on which a drug has been essayed,

regardless of whether the drug was eventually approved. Likewise, only genetic associations with an Open Targets score above 0.16 (see Figure A in S1 Appendix) were considered positive. We considered exclusively common diseases with at least 1,000 Open Targets associations, of which a minimum of 50 could be based on known drugs and 50 on genetic associations, in order to avoid empty folds in the nested cross-validations. By applying these filters, we generated a list of phenotypes and diseases which we then manually curated to remove, non-disease phenotype terms (e.g. "body weight and measures") as well as vague or broad terms (e.g."cerebrovascular disorder" or "head disease") and infectious diseases. We also decided to exclude cancers from this analysis. Cancer is a complex process starting from the driver mutation(s) causing disruptive processes involving clonal expansions, which are known to carry their own specific and resultant (non-causal) passenger mutations. Also, the fundamental genetic and biological mechanisms underlying cancers (Hanahan and Weinberg, 2011) are generally very distinct from other diseases. We considered this might affect the reliability of the seed genes and cancers would therefore deserve a separate benchmark. This left 22 diseases considered in this study (Table 13). Further descriptive material on the role of genes associated with disease within the STRING network can be found in the section "Descriptive disease statistics in the STRING network" from S1 Appendix.

Table 13: List of diseases included in this study.

Disease	N(genetic)	N(drugs)	Overlap	P-value	FDR
allergy	112	57	1	4.22e-01	4.42e-01
Alzheimers disease	208	103	4	1.10e-01	1.42e-01
arthritis	174	188	6	6.08e-02	1.03e-01
asthma	105	80	6	7.77e-05	5.70e-04
bipolar disorder	117	148	3	1.83e-01	2.12e-01
cardiac arrhythmia	75	177	6	9.15e-04	3.36e-03
chronic obstructive pulmonary disease (COPD)	154	116	6	4.18e-03	1.31e-02
coronary heart disease	111	171	4	7.86e-02	1.24e-01
drug dependence	75	143	6	2.96e-04	1.30e-03
hypertension	66	188	2	2.85e-01	3.14e-01
multiple sclerosis	71	167	4	1.83e-02	4.03e-02
obesity	69	194	3	1.06e-01	1.42e-01
Parkinson's disease	55	145	0	1	1
psoriasis	131	105	7	1.68e-04	9.23e-04
rheumatoid arthritis	138	95	5	5.18e-03	1.42e-02
schizophrenia	410	163	17	5.44e-05	5.70e-04
stroke	90	156	3	1.18e-01	1.44e-01
systemic lupus erythematosus (lupus)	126	109	5	6.30e-03	1.54e-02
type I diabetes mellitus	87	106	3	4.39e-02	8.04e-02
type II diabetes mellitus	130	154	4	9.14e-02	1.34e-01
ulcerative colitis	136	51	7	1.81e-06	3.98e-05
unipolar depression	123	121	4	3.81e-02	7.63e-02

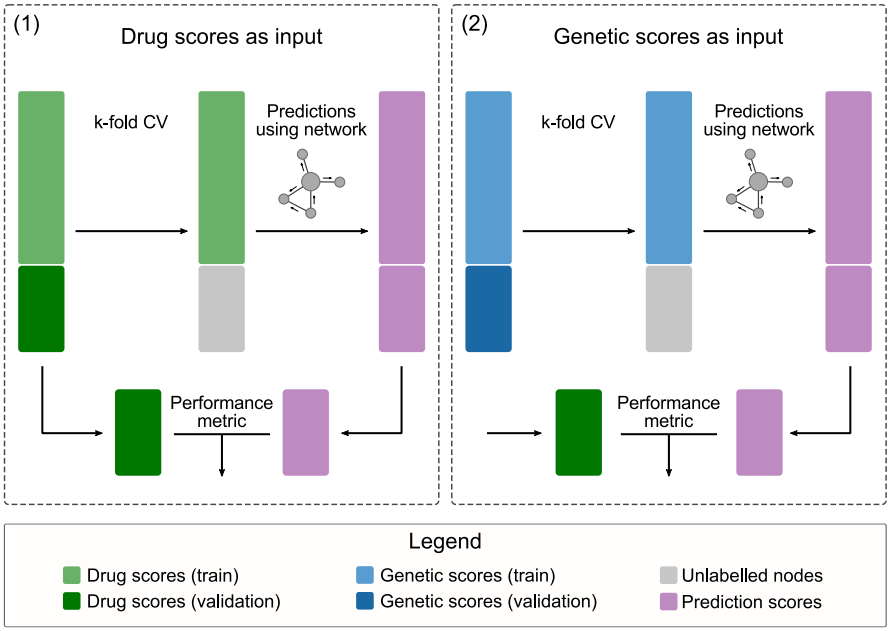
Diseases included in this study, with a minimum of 50 associated genes both in the known drug targets and the genetic categories (see text). The overlap between these two lists of genes showed a degree of dependence between these two Open Targets data streams for some of the diseases. P-values were calculated using Fisher's exact test and are reported without and with correction for false discovery rate (Benjamini and Hochberg, 1995).

8.5.5 Validation strategies

Input gene scores

We used the binarised drug association scores and genetic association scores from Open Targets as input gene-level scores to seed the network propagation analyses (Fig 35) and test their ability to recover known drug targets. With the first approach (panel (A) in Fig 35), we tested the predictive power of current network propagation methods for drug target identification using a direct source of evidence (known drug targets). In the second approach (panel (B) in Fig 35), we assessed the ability of a reasonable but indirect source of evidence – genetic associations to disease – in combination with network propagation to recover known drug targets.

Figure 35: Input gene scores. Two input types were used to feed the prioritisation algorithms: the binary drug scores in panel (A) and the binary genetic scores in panel (B). In both cases, the validation genes were deemed unlabelled in the input to the prioritisers. Cross-validation folds were always calculated taking into account the drugs input and reused on the genetic input.



Metrics

Methods were systematically compared using standard performance metrics. The Area under the Receiver Operating Characteristic curve (AUROC) is extensively used in the literature for binary classification of disease genes (Lee et al., 2011), but can be misleading in this context given the extent of the class imbalance between target and non-target genes (Saito and Rehmsmeier, 2015). We however included it in our benchmark for comparison with previous literature. More suitable measures of success in this case are Area under the Precision-Recall curve (AUPRC) (Saito and Rehmsmeier, 2015) and partial AUROC (pAUROC) (McClish, 1989).

Based on the notation in (Boyd et al., 2013; Dodd and Pepe, 2003; McClish, 1989), let Z be a real-valued random variable corresponding to the output of a given prioritiser, so that largest values correspond to top ranked genes. Let X and Y be the outputs for negative and positive genes, i.e. Z is a mixture of X and Y, representing by D the indicator variable ($D = 0$ for negatives and $D = 1$ for positives). For an arbitrary threshold c, the following metrics can be defined: true positive rate $TPR(c) = P(Y > c) = P(Z > c \mid D = 1)$, false positive rate $FPR(c) = P(X > c) = P(Z > c \mid D = 0)$, precision $Prec(c) = P(D = 1 \mid Z > c)$ and recall $Recall(c) = P(Y > c)$. Then:

$$AUROC = \int_{c=\infty}^{-\infty} TPR(c)\, dFPR(c) \tag{42}$$

$$pAUROC(p) = \frac{1}{p} \int_{c=\infty}^{c_p} TPR(c)\, dFPR(c) \qquad \text{where } FPR(c_p) = p \in (0, 1) \tag{43}$$

$$AUPRC = \int_{c=\infty}^{-\infty} Prec(c)\, dRecall(c) \tag{44}$$

Note that pAUROC contains a normalising constant $\frac{1}{p}$ because the partial area is bounded between 0 and p; the constant allows the metric to lie in $[0, 1]$ again. AUROC, AUPRC and pAUROC were computed with the precrec R package (Takaya Saito and Marc Rehmsmeier, 2017). We also included top k hits, defined as the number of true positives in the top k predicted genes (proportional to precision at k). Given the output of a prioritiser on n genes, $z_1 \geqslant z_2 \geqslant z_3 \geqslant \ldots \geqslant z_n$:

$$top(k) = \sum_{i=z_1}^{z_k} D_i \tag{45}$$

It is straightforward, intuitive and most likely to be useful in practice, such as a screening experiment where only a small number of predicted hits can be assayed.

The main body focuses on AUROC, AUPRC and top 20 hits. We considered another 3 metrics, reported only in S1 Appendix: partial AUROC up to 5% FPR, partial AUROC up to 10% FPR, and number of hits within the top 100 genes.

Cross-validation schemes and protein complexes

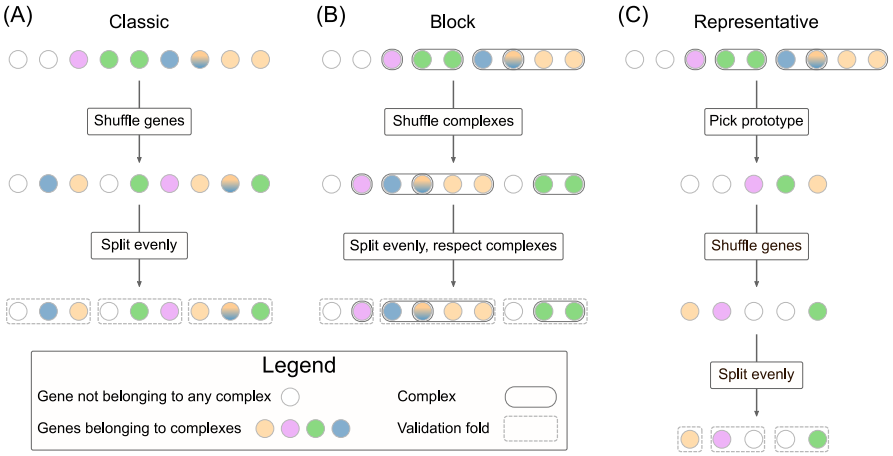
Standard (stratified) and modified k-fold cross-validation were used to estimate the performance of the methods. Folds were based upon known drugs-related genes, regardless of which type of input was used (see Fig 35). Genes in the training fold were negatively or positively labelled according to the input type, whereas genes in the validation fold were left unlabelled.

The direct application of cross-validation to this problem posed a challenge: known drug targets often consist of protein complexes, e.g. multi-protein receptors. Drug-target associations typically have complex-level resolution. The drug target data from Open Targets comes from ChEmbl (Bento

et al., 2014), in which all the proteins in the targeted complex are labelled as targets.

If left uncorrected, this could bias cross-validation results: networks densely connect proteins within a complex, random folds would frequently split positively labelled complexes between train and validation, and therefore network propagation methods would have an unfair advantage at finding positives in the training folds. In view of this, we benchmarked the methods under three cross-validation strategies: a standard cross-validation (A) in line with usual practice and two (B, C) complex-aware schemes (Fig 36) addressing non-independence between folds when the known drug targets act as input.

Figure 36: Cross-validation schemes. Three cross-validation schemes were tested. **(A)**: standard k-fold stratified cross-validation that ignored the complex structure. **(B)**: block k-fold cross-validation. Overlapping complexes were merged and the resulting complexes were shuffled. The folds were computed as evenly as possible without breaking any complex. **(C)**: representative k-fold cross-validation. Overlapping complexes were merged and the resulting complexes from which unique representatives were chosen uniformly at random. Then a standard k-fold cross-validation was run on the representatives, but excluding the non-representatives from train and validation.



Strategy (A), called `classic`, was a regular stratified k-fold repeated cross-validation. We used $k = 3$ folds, averaging metrics over each set of folds, repeated 25 times (see also Fig 28).

Strategy (B), named `block`, performed a repeated cross-validation while explicitly preventing any complexes that contain disease genes to be split across folds. The key point is that, where involved, shuffling was performed at the complex level instead of the gene level – overlapping complexes that shared at least one known drug target were merged into a larger pseudo-complex before shuffling. Fold boundaries were chosen so that no complex was divided into two folds, while keeping them as close as possible to those that would give a balanced partition, see Fig 36. Nevertheless, a limitation of this scheme is that it can fail to balance fold sizes in the presence of large complexes (see Figure I in S1 Appendix). For example, chronic obstructive pulmonary disease exhibited imbalanced folds, as 50 of the proteins involved belong to the Mitochondrial Complex I

Strategy (C), referred to as `representative`, selected only a single representative or prototype gene for each complex to ensure that gene information in a complex was not mixed between training and validation folds. In each repetition of cross-validation, after merging the overlapping complexes, a single gene from each complex was chosen uniformly at random and kept as positive. The remaining genes from the complexes involved in the disease were set aside from the training and validation sets, in order (1) not to mislead methods into assuming their labels were negative in the training phase, and (2) not to overestimate (if set as positives) or penalise (if set as negatives) methods that ranked them highly, as they were expected to do so. This strategy kept the folds balanced, but at the expense of a possible loss of information by summarising each complex by a single gene at a time, reducing the number of positives for training and validation.

8.5.6 Additive performance models

For a systematic comparison between diseases, methods, cross-validation schemes and input types, we fitted an additive, explanatory regression model to the performance metrics of each (averaged) fold from the cross-validation. The use of main effect models eased the evaluation of each individual factor while correcting for the other covariates. We modelled each metric f separately for each input type, not to mix problems of different nature:

$$f \sim \text{cv\_scheme} + \text{network} + \text{method} + \text{disease} \tag{46}$$

We fitted dispersion-adjusted logistic-like *quasibinomial* variance models for the metrics AUROC, pAUROC and AUPRC and *quasipoisson* for top k hits. The quasi-likelihood formalism protected against over and under-dispersion issues, in which the observed variance is either higher or lower than that of the theorical fitted distribution (Hardin et al., 2007), affecting subsequent statistical tests. *The effect of changing any of the four main effects is discussed in separate sub-sections in Results, following the order from the formula above.* After a data-driven choice of cross-validation scheme and network, we fitted reduced explanatory models within them for a more accurate description:

$$f \sim \text{method} + \text{disease} \tag{47}$$

8.5.7 Qualitative methods comparison

The rankings produced by the different algorithms were qualitatively compared using Spearman's footrule (Spearman, 1906). Distances were computed between all method ranking pairs for each individual combination of disease, input type, network and for the top N predicted genes, excluding the original seed genes. This part does not involve cross-validation – all known disease-associated genes were used for gene prioritisations. Pairs of rankings could include genes uniquely ranked highly by a single algorithm from the comparison, so mismatch counts (i.e. percentage mismatches) between these rankings were also taken into account. Mismatches occur when

a gene features in the top N predictions of one algorithm and is missing from the corresponding ranking by another algorithm. A compact visualisation of distance matrices was obtained using a multi-view extension of MDS (Gower, 1966; Kanaan-Izquierdo, Ziyatdinov, and Perera-Lluna, 2018; Mardia, 1978). For this we used the R package *multiview* (Kanaan-Izquierdo, Ziyatdinov, Burgueño, et al., 2018) that generates a single, low-dimensional projection of combined inputs (disease, input and network).

ACKNOWLEDGMENTS

AG would like to acknowledge Philippe Sanseau, Matt Nelson and John Whittaker for critical feedback on the applicability of this research to drug discovery.

SUPPORTING INFORMATION

S1 APPENDIX. **Supplement.** This document contains complementary material that supports our claims in the main body. It includes topics such as descriptive statistics, topological properties of disease-associated genes, raw metrics plots, method details, MDS plots, alternative performance metrics and further explanatory models.

S1 FILE. **MDS plots.** Complementary single-disease MDS plots and distance matrices.

S2 FILE. **Interactions HTML viewer.** Stand-alone viewer to explore models with interaction terms.

REFERENCES

Al-Aamri, Amira, Kamal Taha, Yousof Al-Hammadi, Maher Maalouf, and Dirar Homouz

2017 "Constructing Genetic Networks using Biomedical Literature and Rare Event Classification", *Sci. Rep.*, 7, 1, p. 15784.

Ballouz, Sara, Melanie Weber, Paul Pavlidis, and Jesse Gillis

2017 "EGAD: ultra-fast functional analysis of gene networks", *Bioinformatics*, 33, 4, pp. 612-614.

Benjamini, Yoav and Yosef Hochberg

1995 "Controlling the false discovery rate: a practical and powerful approach to multiple testing", *J. Royal Stat. Soc. Series B (Methodological)*, pp. 289-300.

Bento, A Patrícia, Anna Gaulton, Anne Hersey, Louisa J Bellis, Jon Chambers, Mark Davies, Felix A Krüger, Yvonne Light, Lora Mak, Shaun McGlinchey, et al.

2014 "The ChEMBL bioactivity database: an update", *Nucleic Acids Res.*, 42, D1, pp. D1083-D1090.

Bertoni, Alberto, Marco Frasca, and Giorgio Valentini

2011 "COSNet: a Cost Sensitive Neural Network for Semi-supervised Learning in Graphs", *Lect. Notes Comput. Sc.*, 6911, 4, pp. 219-234.

Bischl, Bernd, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, and Zachary M. Jones

2016 "mlr: Machine Learning in R", *J. Mach. Learn. Res.*, 17, 170, pp. 1-5.

Boyd, Kendrick, Kevin H Eng, and C David Page

2013 "Area under the precision-recall curve: point estimates and confidence intervals", in *Joint European conference on machine learning and knowledge discovery in databases*, Springer, pp. 451-466.

Boyle, Evan A., Yang I. Li, and Jonathan K. Pritchard

2017 "An Expanded View of Complex Traits: From Polygenic to Omnigenic", *Cell*, 169, 7 (June 2017), pp. 1177-1186.

Chatr-Aryamontri, Andrew, Rose Oughtred, Lorrie Boucher, Jennifer Rust, Christie Chang, Nadine K Kolas, Lara O'Donnell, Sara Oster, Chandra Theesfeld, Adnane Sellam, et al.

2017 "The BioGRID interaction database: 2017 update", *Nucleic Acids Res.*, 45, D1, pp. D369-D379.

Cho, Hyunghoon, Bonnie Berger, and Jian Peng

2016 "Compact integration of multi-network topology for functional analysis of genes", *Cell Syst.*, 3, 6, pp. 540-548.

Cowen, Lenore, Trey Ideker, Benjamin J. Raphael, and Roded Sharan

2017 "Network propagation: a universal amplifier of genetic associations", *Nat. Rev. Genet.*, 18, 9 (June 2017), pp. 551-562.

Csardi, Gabor and Tamas Nepusz
2006 "The igraph software package for complex network research", *InterJournal*, Complex Systems, p. 1695, http://igraph.org.

Dodd, Lori E and Margaret S Pepe
2003 "Partial AUC estimation and regression", *Biometrics*, 59, 3, pp. 614-623.

Elkan, Charles and Keith Noto
2008 "Learning classifiers from only positive and unlabeled data", in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 213-220.

Frasca, Marco, Alberto Bertoni, Matteo Re, and Giorgio Valentini
2013 "A neural network algorithm for semi-supervised node label learning from unbalanced data", *Bioinformatics*, 43, C, pp. 84-98.

Gentleman, R.C., V.J. Carey, D.M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A.J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J.Y. Yang, and J. Zhang
2004 "Bioconductor: open software development for computational biology and bioinformatics", *Genome Biol.*, 5, R80.

Gillis, Jesse and Paul Pavlidis
2012 ""Guilt by association" is the exception rather than the rule in gene networks", *PLoS Comput. Biol.*, 8, 3, e1002444.

Gower, John C
1966 "Some distance properties of latent root and vector methods used in multivariate analysis", *Biometrika*, 53, 3-4, pp. 325-338.

Hanahan, Douglas and Robert A Weinberg
2011 "Hallmarks of cancer: the next generation", *Cell*, 144, 5, pp. 646-674.

Hardin, James W, James William Hardin, Joseph M Hilbe, and Joseph Hilbe
2007 "Generalized linear models and extensions", in Stata press, chap. 17.

Hothorn, Torsten, Frank Bretz, and Peter Westfall
2008 "Simultaneous Inference in General Parametric Models", *Biom. J.*, 50, 3, pp. 346-363.

Huang, Justin K, Daniel E Carlin, Michael Ku Yu, Wei Zhang, Jason F Kreisberg, Pablo Tamayo, and Trey Ideker
2018 "Systematic Evaluation of Molecular Networks for Discovery of Disease Genes", *Cell Syst.*, 6, 4, pp. 484-495.

Huber, W., V. J. Carey, R. Gentleman, S. Anders, M. Carlson, B. S. Carvalho, H. C. Bravo, S. Davis, L. Gatto, T. Girke, R. Gottardo, F. Hahne, K. D. Hansen, R. A. Irizarry, M. Lawrence, M. I. Love, J. MacDonald, V. Obenchain, A. K. Ole's, H. Pag'es, A. Reyes, P. Shannon, G. K. Smyth, D. Tenenbaum, L. Waldron, and M. Morgan
2015 "Orchestrating high-throughput genomic analysis with Bioconductor", *Nat. Methods*, 12, 2, pp. 115-121.

Jia, Peilin and Zhongming Zhao

2013 "Network-assisted analysis to prioritize GWAS results: principles, methods and perspectives", *Hum. Genet.*, 133, 2 (Oct. 2013), pp. 125-138.

Jiang, Biaobin, Kyle Kloster, David F Gleich, and Michael Gribskov

2017 "AptRank: an adaptive PageRank model for protein function prediction on bi-relational graphs", *Bioinformatics*, 33, 12, pp. 1829-1836.

Kanaan-Izquierdo, Samir, Andrey Ziyatdinov, Maria Araceli Burgueño, and Alexandre Perera-Lluna

2018 "multiview: a software package for multiview pattern recognition methods", *Bioinformatics*, bty1039.

Kanaan-Izquierdo, Samir, Andrey Ziyatdinov, and Alexandre Perera-Lluna

2018 "Multiview and multifeature spectral clustering using common eigenvectors", *Pattern Recognit. Lett.*, 102, pp. 30-36.

Karatzoglou, Alexandros, Alex Smola, Kurt Hornik, and Achim Zeileis

2004 "kernlab – An S4 Package for Kernel Methods in R", *J. Stat. Softw.*, 11, 9, pp. 1-20.

Kerrien, Samuel, Bruno Aranda, Lionel Breuza, Alan Bridge, Fiona Broackes-Carter, Carol Chen, Margaret Duesbury, Marine Dumousseau, Marc Feuermann, Ursula Hinz, et al.

2011 "The IntAct molecular interaction database in 2012", *Nucleic Acids Res.*, 40, D1, pp. D841-D846.

Koscielny, Gautier, Peter An, Denise Carvalho-Silva, Jennifer A Cham, Luca Fumis, Rippa Gasparyan, Samiul Hasan, Nikiforos Karamanis, Michael Maguire, Eliseo Papa, et al.

2016 "Open Targets: a platform for therapeutic target identification and validation", *Nucleic Acids Res.*, 45, D1, pp. D985-D994.

Kunn, Max

2008 "Building Predictive Models in R Using the caret Package", *J. Stat. Softwe.*, 28, 5, pp. 1-26.

Langfelder, Peter and Steve Horvath

2008 "WGCNA: an R package for weighted correlation network analysis", *BMC Bioinformatics*, 9, 1, p. 559.

Lee, I., U. M. Blom, P. I. Wang, J. E. Shim, and E. M. Marcotte

2011 "Prioritizing candidate disease genes by network-based boosting of genome-wide association data", *Genome Res.*, 21, 7 (May 2011), pp. 1109-1121.

Leiserson, Mark D M, Fabio Vandin, Hsin-Ta Wu, Jason R Dobson, Jonathan V Eldridge, Jacob L Thomas, Alexandra Papoutsaki, Younhun Kim, Beifang Niu, Michael McLellan, Michael S Lawrence, Abel Gonzalez-Perez, David Tamborero, Yuwei Cheng, Gregory A Ryslik, Nuria Lopez-Bigas, Gad Getz, Li Ding, and Benjamin J Raphael

2014 "Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes", *Nature Genet.*, 47, 2 (Dec. 2014), pp. 106-114.

Liaw, Andy and Matthew Wiener

2002 "Classification and Regression by randomForest", *R News*, 2, 3, pp. 18-22, http://CRAN.R-project.org/doc/Rnews/.

Mardia, Kanti V

1978 "Some properties of classical multi-dimensional scaling", *Commun. Stat. Theory Methods*, 7, 13, pp. 1233-1241.

McClish, Donna Katzman

1989 "Analyzing a portion of the ROC curve", *Med. Decis. Mak.*, 9, 3, pp. 190-195.

Mordelet, Fantine and Jean-Philippe Vert

2011 "ProDiGe: Prioritization Of Disease Genes with multitask machine learning from positive and unlabeled examples", *BMC Bioinformatics*, 12, 1, p. 389.

Mostafavi, Sara, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris

2008 "Genemania: a real-time multiple association network integration algorithm for predicting gene function", *Genome Biol.*, 9, S4, pp. 1-15.

Nelson, Matthew R, Hannah Tipney, Jeffery L Painter, Judong Shen, Paola Nicoletti, Yufeng Shen, Aris Floratos, Pak Chung Sham, Mulin Jun Li, Junwen Wang, Lon R Cardon, John C Whittaker, and Philippe Sanseau

2015 "The support of human genetic evidence for approved drug indications", *Nature Genet.*, 47, 8 (June 2015), pp. 856-860.

Orchard, S., S. Kerrien, S. Abbani, B. Aranda, J. Bhate, S. Bidwell, A. Bridge, L. Briganti, F. S. Brinkman, F. Brinkman, G. Cesareni, A. Chatraryamontri, E. Chautard, C. Chen, M. Dumousseau, J. Goll, R. E. Hancock, R. Hancock, L. I. Hannick, I. Jurisica, J. Khadake, D. J. Lynn, U. Mahadevan, L. Perfetto, A. Raghunath, S. Ricard-Blum, B. Roechert, L. Salwinski, V. Stumpflen, M. Tyers, P. Uetz, I. Xenarios, and H. Hermjakob

2012 "Protein interaction data curation: the International Molecular Exchange (IMEx) consortium", *Nat. Methods*, 9, 4 (Apr. 2012), pp. 345-350.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd
1999 *The PageRank citation ranking: Bringing order to the web.* Tech. rep., Stanford InfoLab.

Picart-Armada, Sergio, Francesc Fernández-Albert, Maria Vinaixa, Miguel A Rodríguez, Suvi Aivio, Travis H Stracker, Oscar Yanes, and Alexandre Perera-Lluna
2017 "Null diffusion-based enrichment for metabolomics data", *PloS one*, 12, 12, e0189012.

Picart-Armada, Sergio, Wesley K Thompson, Alfonso Buil, and Alexandre Perera-Lluna
2017 "diffuStats: an R package to compute diffusion-based scores on biological networks", *Bioinformatics*, 34, 3, pp. 533-534.

Piovesan, Damiano, Manuel Giollo, Carlo Ferrari, and Silvio C. E. Tosatto
2015 "Protein function prediction using guilty by association from interaction networks", *Amino Acids*, 47, 12 (July 2015), pp. 2583-2592.

R Core Team
2016 *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Razick, Sabry, George Magklaras, and Ian M Donaldson
2008 "iRefIndex: A consolidated protein interaction database with provenance", *BMC Bioinformatics*, 9, 1, p. 405.

Re, Matteo, Marco Mesiti, and Giorgio Valentini
2012 "A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks", *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 9, 6, pp. 1812-1818.

Saito, Takaya and Marc Rehmsmeier
2015 "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets", *PLoS One*, 10, 3 (Mar. 2015), ed. by Guy Brock, e0118432.

Scannell, Jack W, Alex Blanckley, Helen Boldon, and Brian Warrington
2012 "Diagnosing the decline in pharmaceutical R&D efficiency", *Nat. Rev. Drug Discov.*, 11, 3, pp. 191-200.

Schizophrenia Working Group of the Psychiatric Genomics Consortium
2014 "Biological insights from 108 schizophrenia-associated genetic loci", *Nature*, 511, 7510 (July 2014), pp. 421-427.

Sharan, Roded, Igor Ulitsky, and Ron Shamir
2007 "Network-based prediction of protein function", *Mol. Syst. Biol.*, 3 (Mar. 2007).

Smola, Alexander J and Risi Kondor
2003 "Kernels and regularization on graphs", in *Learning theory and kernel machines*, Springer, pp. 144-158.

Spearman, Charles
1906 "'Footrule'for measuring correlation", *Br. J. Psychol.*, 2, 1, pp. 89-108.

Szklarczyk, Damian, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi P Tsafou, et al.
2014 "STRING v10: protein–protein interaction networks, integrated over the tree of life", *Nucleic Acids Res.*, 43, D1, pp. D447-D452.

Szklarczyk, Damian, John H Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda T Doncheva, Alexander Roth, Peer Bork, Lars J. Jensen, and Christian von Mering
2016 "The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible", *Nucleic Acids Res.*, 45, D1 (Oct. 2016), pp. D362-D368.

Tabe-Bordbar, Shayan, Amin Emad, Sihai Dave Zhao, and Saurabh Sinha
2018 "A closer look at cross-validation for assessing the accuracy of gene regulatory networks and models", *Sci. Rep.*, 8.

Takaya Saito and Marc Rehmsmeier
2017 "Precrec: fast and accurate precision-recall and ROC curve calculations in R", *Bioinformatics*, 33 (1), pp. 145-147.

Türei, Dénes, Tamás Korcsmáros, and Julio Saez-Rodriguez
2016 "OmniPath: guidelines and gateway for literature-curated signaling pathway resources", *Nat. Methods*, 13, 12, p. 966.

Valentini, Giorgio, Giuliano Armano, Marco Frasca, Jianyi Lin, Marco Mesiti, and Matteo Re
2016 "RANKS: a flexible tool for node label ranking and classification in biological networks", *Bioinformatics*, 32, 18, pp. 2872-2874.

Valentini, Giorgio, Alberto Paccanaro, Horacio Caniza, Alfonso E. Romero, and Matteo Re
2014 "RANKS: a flexible tool for node label ranking and classification in biological networks", *Artif. Intell. Med.*, 61, 2, pp. 63-78.

Vandin, Fabio, Eli Upfal, and Benjamin J. Raphael
2011 "Algorithms for detecting significantly mutated pathways in cancer", *J. Comput. Biol.*, 18, 3, pp. 507-22.

Verstockt, Bram, Kenneth GC Smith, and James C Lee
2018 "Genome-wide association studies in Crohn's disease: Past, present and future", *Clin. Transl. Immunology*, 7, 1, e1001.

Yang, Peng, Xiao-Li Li, Jian-Ping Mei, Chee-Keong Kwoh, and See-Kiong Ng
2012 "Positive-unlabeled learning for disease gene identification", *Bioinformatics*, 28, 20, pp. 2640-2647.

9 PUBLICATIONS AND DISCUSSION

The results were hereby summarised, by scientific publication. The last section presents a conceptual breakdown of the contributions as a whole.

9.1 CONCEPTION OF THE STATISTICAL NORMALISATION

9.1.1 Characterisation of the statistical normalisation

> Sergio Picart-Armada, Wesley K Thompson, Alfonso Buil, and Alexandre Perera-Lluna (2020), "The effect of statistical normalisation on network propagation scores", *BioRxiv*

The bias of the diffusion scores was characterised by studying the two first statistical moments of their null distribution. Two biases were examined, depending on their main source: expected value and, in its absence, variance-related bias. The so-called reference expected value and reference variance were defined, in order to quantify their presence.

The usage of graph kernels derived from the unnormalised graph Laplacian matrix would generally lead to expected value-related bias in the presence of unlabelled nodes. If all the nodes had a label, the expected value bias would disappear, but the variance-related bias would persist.

Several propositions of the diffusion scores were proven, remarkably:

- Closed expressions of the mean value vector and covariance matrix of the null distributions.
- Some diffusion scores lead to identical node prioritisations under certain conditions, simplifying their choice thereof.
- Parametric and non-parametric normalisations are invariant to changes in the weights of each label.
- The null covariance is directly related to graph spectral properties. In particular, the principal covariance direction is proportional to the Fiedler vector.

A proof of concept, synthetic study was conceived to generate artificial signals with and without expected value-related bias, consisting of a list of true nodes that had to be found starting from a list of seed nodes. Normalised scores were preferrable on unbiased signals, whereas unnormalised scores worked best on biased signals. This provided a first criterion to decide about normalising: if the positives are expected to be unbiased, normalisation is advised; otherwise it is discouraged.

An artificial gene expression array dataset illustrated that the bias can be counterintuitive. Normalisation helped when re-prioritising labelled nodes, while it was detrimental when prioritising unlabelled nodes. This was explained through the expected value-related bias, which had opposite directions within labelled and unlabelled nodes.

A third dataset posed the problem of prospective pathay gene prediction, affected by a variance-related bias. New genes in KEGG pathways from August 2018 were sought by using the same pathways from March 2011 and the BioGRID network from 2011. The topological properties of the new genes were unknown beforehand, but were expected to differ from those of pathway genes in 2011. We hypothesised that the network would not provide a highly consistent knowledge representation on the novel genes from 2018. Normalised scores outperformed their unnormalised counterparts, further supporting this statement.

9.1.2 The diffuStats R package

> Sergio Picart-Armada, Wesley K Thompson, Alfonso Buil, and Alexandre Perera-Lluna (2017), "diffuStats: an R package to compute diffusion-based scores on biological networks", *Bioinformatics*, 34, 3, pp. 533-534

The characterisation of the diffusion scores highlighed the importance of examining the presence of bias, and its alignment with the properties of the positive nodes. These algorithms were implemented in an R package named *diffuStats* to ease their benchmark and adoption by the scientific community. *diffuStats* was also published in Bioconductor[1] and downloaded 1,462 times from 619 unique IP addresses during 2019[2].

diffuStats was based on the graph kernel formalism and implemented seven diffusion scores: `raw`, `ml`, `gm`, `mc`, `z`, `ber_s` and `ber_p`. Difference between scores stemmed from how the positive and negative labels were codified, i.e. which quantities are diffused on positive and negative nodes, and the presence of a statistical normalisation. The following scores were unnormalised:

- `raw`: classical diffusion scores. If the input contained binary classes, the positive class would diffuse 1 positive unit in each node and the negative class would not diffuse anything.
- `ml`: like `raw`, but the negative class would diffuse one negative unit in each negative node.
- `gm`: like `ml`, but the unlabelled nodes would diffuse a quantity based on the balance between positives and negatives.
- `ber_s`: hybrid option that quantifies the relative change of the score of a node, before and after diffusion.

1 *diffuStats* can be found at https://doi.org/doi:10.18129/B9.bioc.diffuStats. Accessed on 31/12/2019.

2 http://bioconductor.org/packages/stats/bioc/diffuStats/. Accessed on 09/02/2020.

On the other hand, the scores below involved a statistical normalisation:

- `mc`: non-parametric normalisation, computationally intensive as it requires permutations
- `z`: parametric normalisation
- `ber_p`: hybrid option that combines `raw` with `mc`

diffuStats was equipped with a quickstart vignette and a main vignette that elaborates on the scores, the kernels, and provides a sample case study. The dataset contained 13 annotations of biological functions in a yeast interactome. For each function, half of the proteins were used as positives in the input, while the other half helped estimate the performance. Trying the seven diffusion scores revealed that `z` was slightly preferrable over the unnormalised alternatives.

9.2 APPLICATION TO METABOLOMICS DATA ENRICHMENT

The diffusion formalism that was explored in the two articles above, especially the implications of the statistical normalisation, had a potential impact on a wide array of computational biology areas. The first choice of application was data interpretation in metabolomics through network-based algorithms. The technical limitations of experimental devices in metabolomics and the lack of a mature, comprehensive, well-established range of tools for understanding this data further encouraged efforts in this direction.

9.2.1 Null diffusion-based enrichment for metabolomics data

> Sergio Picart-Armada, Francesc Fernández-Albert, Maria Vinaixa, Miguel A Rodríguez, Suvi Aivio, Travis H Stracker, Oscar Yanes, and Alexandre Perera-Lluna (2017), "Null diffusion-based enrichment for metabolomics data", *PloS one*, 12, 12, e0189012

This article developed a novel pathway enrichment technique to overcome the low interpretability of standard tools. For this purpose, a knowledge graph was built from the KEGG database, with a hierarchical representation of entities that connect metabolites to biological pathways: reactions, enzymes and KEGG modules. Mining this object enabled a rich interpretation on how affected metabolites might translate into dysregulated pathways.

Given a list of input metabolites, this approach provided a relevant but succint biological explanation, in the form of a subgraph from the knowledge graph. Node prioritisation was achieved through the definition of a diffusion process: the input metabolites would introduce one flow unit each, whereas only nodes corresponding to biological pathways were allowed to dispel it. The sub-network was prioritised according to the best diffusion scores.

The presence of topology-related biases in diffusion scores was examined. Two factors had a noticeable impact: the molecular level of the node (i.e. level within the hierarchy) and its degree. Specifically, prioritising pathways by their unnormalised scores greatly correlated with the expected value of their null distributions, in turn related to their degree. The statistical normalisation alleviated these shortcomings and was therefore included.

This diffusion-based approach was validated on a case-control experiment aimed at characterising a mitochondrial protein. Affected metabolites were derived by two experimental platforms: LC/MS[3] and NMR[4]. The algorithm suggested several sub-networks (one for each unique combination of parameters) starting from the LC/MS metabolites. The reported pathways were consistent with those obtained by state-of-the-art tools. Within the knowledge graph, the reported reactions were closer to the NMR metabolites than the bulk of reactions involving any of the LC/MS metabolites, which proved that the suggestions of entities between metabolites and pathways was meaningful.

9.2.2 The FELLA R package

> Sergio Picart-Armada, Francesc Fernández-Albert, Maria Vinaixa, Oscar Yanes, and Alexandre Perera-Lluna (2018), "FELLA: an R package to enrich metabolomics data", *BMC bioinformatics*, 19, 1, p. 538

After demonstrating that the novel diffusion-based approach could provide valuable biological insights within the knowledge graph, the algorithms were disseminated as an R package: *FELLA*. *FELLA* is part of the Bioconductor repository[5] for bioinformatics tools and was downloaded 2,078 times from 905 unique IP addresses during 2019[6].

FELLA was organised in three blocks: database creation, data enrichment and results exporting. *FELLA* provided an automated way to generate any organism-specific knowledge graph from the latest KEGG release, also allowing to filter out user-defined pathways. The user input, a list of metabolies as KEGG identifiers, is mapped to the knowledge graph and the (raw) diffusion scores are computed using the unnormalised regularised Laplacian kernel or the PageRank algorithm. The statistical normalisation of choice (parametric or non-parametric) is applied and mapped to lie in $[0, 1]$, referred to as the `p-score`. Once a threshold on the `p-score` or on the number of nodes is defined, an optional filter discards small connected components whose order could arise from random selection of graph nodes. Finally, several plotting and exporting options were implemented for tabular and network data. A graphical interface facilitates analytical and exporting routines.

FELLA's usefulness was shown on six case studies from public datasets. For each dataset, a subgraph was generated and connected to the findings

3 Liquid Chromatography followed by Mass Spectrometry
4 Nuclear Magnetic Resonance
5 *FELLA* can be downloaded from https://doi.org/doi:10.18129/B9.bioc.FELLA. Accessed on 31/12/2019.
6 http://bioconductor.org/packages/stats/bioc/FELLA/. Accessed on 09/02/2020.

within the original article and in independent literature. The variety of organisms (human, mouse and zebrafish) and case studies (in silico essays, disease studies, animal models) highlighted the potential utility of *FELLA* to a broad metabolomics community.

FELLA promotes reproducible and accessible research with the inclusion of reproducible and self-explanatory vignettes for every study. Any potential user only needs to replace few lines of code to leverage the knowledge graph on their data.

9.2.3 Gilt-head bream oxybenzone exposition study

> Haizea Ziarrusta, Leire Mijangos, Sergio Picart-Armada, Mireia Irazola, Alexandre Perera-Lluna, Aresatz Usobiaga, Ailette Prieto, Nestor Etxebarria, Maitane Olivares, and Olatz Zuloaga (2018), "Non-targeted metabolomics reveals alterations in liver and plasma of gilt-head bream exposed to oxybenzone", *Chemosphere*, 211, pp. 624-631

FELLA was applied to an ecotoxicological study that studied the effect of oxybenzone on juvenile gilt-head bream[7]. 50 fish shared a water tank and underwent exposures of 0, 2, 4, 7 or 14 days (10 fish were sampled in each timepoint). Likewise, 50 fish were kept in a control tank with the same sampling frequence. Differentially abundant metabolites in liver, brain and plasma were sought through untargeted metabolomics. Liver and plasma showed metabolic alterations, whereas changes in brain could not be proven. After mapping the affected metabolites to the KEGG database, *FELLA* was used to elucidate a biological explanation of the metabolic perturbations.

In liver, 8 metabolic features were affected, from which 4 mapped to KEGG. Pathway enrichment reported amino acid metabolism as a relevant process, specifically phenylalanine metabolism, as well as tyrosine-related metabolites. Oxidative stress was also highlighted as perturbed, found by the alterations on lipidic metabolites, on hippurate levels and on precursors of the carbohydrate metabolism.

An analogous statistical analysis yielded 10 metabolic features in plasma (9 in KEGG). *FELLA* suggested that such metabolites were mainly involved in the alteration of the lipid metabolism, concretely in fatty acid elongation, alpha-linolenic acid metabolism, biosynthesis of unsaturated fatty acids and fatty acid metabolism. The alterations of 8 metabolites in lipid metabolism was linked to oxidative stress since several authors had previously reported that UV filters such as oxybenzone generate oxidative stress in fish. There were also signs of glutathione metabolism perturbation since 5-oxo-L-proline was altered.

In summary, the study showed that despite an absence of mortality or alterations in general physiological parameters (i.e, fish weight and length) and brain metabolome, oxybenzone produced significant metabolic perturbations in both liver and plasma. The alterations on energy metabolism

7 This article was included as an appendix of this thesis

and oxidative stress can lead to important health implications on fish, encouraging more metabolomics studies on non-lethal levels of xenobiotics for environmental risk assessment.

9.3 APPLICATION TO GENE TARGET DISCOVERY

In silico gene target discovery is a classical problem in computational biology with notable implications in the drug discovery cycle. Network-based approaches offer competitive performances, opening the question about the adequateness of the statistical normalisation of diffusion scores.

9.3.1 Benchmark of gene target prioritisation

S Picart-Armada, SJ Barrett, DR Willé, A Perera-Lluna, A Gutteridge, and BH Dessailly (2019), "Benchmarking network propagation methods for disease gene identification", *PLoS Comput Biol*, 15, 9, e1007276

This manuscript describes a benchmark of several network diffusion-based approaches, including naive neighbour-voting, semi-supervised and supervised machine learning approaches. Special attention was payed to the validation scheme and the informativeness of performance metrics in the drug development field.

The Open Targets platform was used to retrieve 22 common diseases where at least 50 genes had been essayed in phase I or beyond. Two networks were considered: STRING with filters on edge types and weights, which was larger but noisier, and OmniPath, more stringent and confident. Provided that drug target data is usually known at the protein complex level, known complexes were retrieved from ChEMBL.

The effect of changing the diffusion approach, the network, the disease and the cross-validation strategy was quantified with explanatory models on the performance estimates. The validation strategy was reported as the most influential factor in the study. Ignoring the protein complex data would lead to circularity in the cross-validation and to performance overestimation. Imposing that no protein complex shall be splitted, performances experienced a pronounced drop, but were still encouraging the adoption of in silico drug discovery. Competitive methods would typically find between 2.5 and 4 true hits within the 20 highest prioritised genes.

Although the non-parametric normalisation `mc` performed poorly, the parametric `z` and the classical diffusion scores `raw` were competitive, `raw` with a narrow lead. On the other hand, low-dimensional projections of the differences between the top 100 predictions by normalised and unnormalised diffusion scores unveiled distinct behaviours. Combining both observations, the normalised and unnormalised counterparts might be successful because of different underlying mechanisms.

9.4 OUTCOME

The scientific output has been so far divided into three main topics based on their domain of application: general purpose, metabolomics and disease gene data. However, such boundaries are not aligned with the conceptual contributions of the articles. To that end, figure 37 draws an alternative classification of the scientific articles.

- The preprint *The effect of statistical normalisation on diffusion scores in computational biology* and the article *Null diffusion-based enrichment for metabolomics data* address fundamental questions on the bias and justify how and why should diffusion scores be normalised. The deterministic (parametric) normalisation provides the benefits of normalising without the drawbacks of an explicit permutation analysis (computationally intensive, stochastic and approximate).
- The articles *diffuStats: an R package to compute diffusion-based scores on biological networks* and *FELLA: an R package to enrich metabolomics data* provide software implementations of the general purpose and metabolomics applications.
- The articles *Benchmarking network propagation methods for disease gene identification* and *Non-targeted metabolomics reveals alterations in liver and plasma of gilt-head bream exposed to oxybenzone* are case studies, where the main question does not involve the statistical normalisation. Instead, the latter is a proxy to achieve another goal; respectively, choosing a validation scheme for network propagation-based disease gene prioritisers and understanding metabolic changes in gilt-head bream when exposed to environmental contamination.

Therefore, this thesis conveys and end-to-end perspective of the normalisation of diffusion scores, from conceptualisation to practical application.

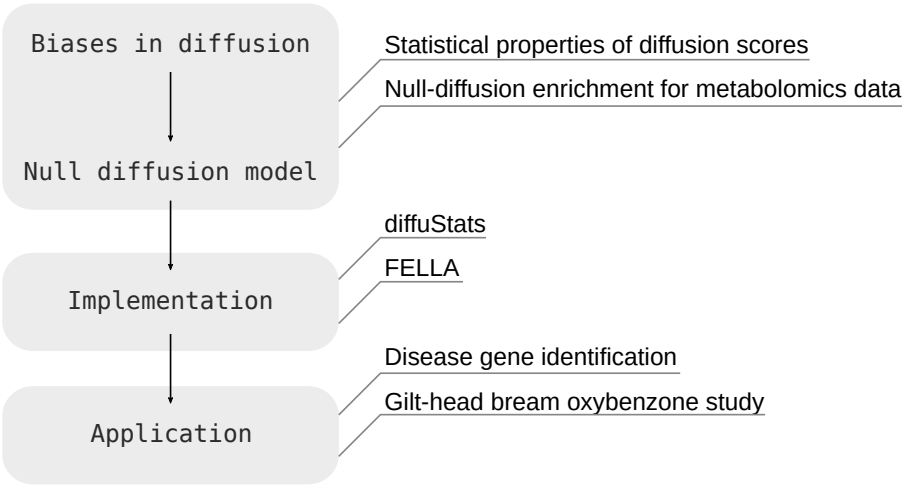


Figure 37: Conceptual map of the thesis. This figure links the fundamental ideas in this thesis to its scientific publications. The algorithmic part is covered by the discovery, characterisation and removal of the bias through null diffusion models. The implementation block distributes the algorithms within open source software tools for the scientific community. The application block involves studies focused on a specific biological question rather than the algorithmic part.

10 CONCLUSIONS

10.1 CONCLUSION

The present doctoral thesis was motivated by the desire to explore the statistical properties of diffusion scores.

- The first findings pointed to the presence of a bias that could, a priori, be due to the network topology and to the user input. In parallel, parametric and non-parametric statistical normalisations were conceived.
- This work showed that the graph kernel, the network topology and the statistical background played key roles in biasing diffusion scores. The covariance of the null distributions was tightly connected with the spectral properties of the graph. Guidelines were derived, suggesting to normalise only if the bias was expected to hinder novel findings.
- The statistical normalisations were implemented and published within the R package *diffuStats* to ease their adoption and benchmark by the scientific community.

Besides, two areas of computational biology were revisited from the statistical normalisation perspective: metabolomics pathway enrichment and target gene discovery. This choice allowed its application to two distinct network types: a knowledge graph, whose purpose is to represent our understanding of biological mechanisms, and protein interaction networks, which depict physical events between molecular entities.

- A knowlege graph was outlined specifically for metabolomics data. Starting from a list of metabolites, not only biological pathways were pointed out, but also other molecular entitites (reactions, enzymes and modules).
- A method based on network diffusion was developed to prioritise entities. The normalisation was found mandatory due to the hierarchical nature of the network. This algorithm was validated on an in vitro study, with measurements from two metabolomics experimental platforms.
- The pathway enrichment algorithm was distributed within the R package *FELLA*. Users can create knowledge graphs for their organism of choice, run the diffusion prioritiser and export the results as networks or tables. Six case studies were bundled, with the code and the discussion of the findings, to facilitate a starting point for new users.

On the other hand, the target gene prediction was benchmarked for an array of network-based algorithms, ranging from simple neighbour voting to supervised learning.

- This work emphasised on the necessity of a proper protein complex-aware validation scheme. Although pure diffusion-based methods were competitive, the best method was a random forest classifier on the top of network-based features.
- The parametric normalisation and the unnormalised scores performed similarly but with diverging behaviours, suggesting some degree of complementarity.
- Explanatory additive models allowed a systematic assessment of the impact of several factors (method, disease, cross-validation strategy and network) in the performance estimates.

To conclude, the statistical normalisation had an impact in all the areas covered in the present thesis. The normalisation did not imply a systematic improvement everywhere, but rather provided a first step to control for unwanted biases. Its adequateness depended on each particular instance and should ideally undergo ad-hoc examination in future cases. This thesis contributed with the basis and the means to that end.

10.2 FUTURE WORK

10.2.1 Statistical normalisation

The formulation of the parametric scores opens several reserach lines, depending on the final purpose.

Binned version

Only one bin is permuted in the current normalisation, but some authors have pointed out benefits of controlling for further confounding factors by binning the input nodes. The parametric approach would be especially beneficial due to its computational advantage in medium networks.

Multivariate version

The parametric normalisation is univariate, but it can be defined in a multivariate sense. The assumptions and implications of the multivariate approach would need careful understanding to identify challenges in computational biology fitting these needs.

Accounting for uncertainty

The statistical model can be further modified to accomodate for unobserved features. Quantifying the impact of uncertainty can bring robustness and a measure of sensitivity to current network analyses.

Characterising other null models

Null models based on edge rewiring have found use in the literature. Their statistical characterisation might shed light on their behaviour, along

with similarities and differences with the input permutation-based null models.

Multilayer networks

Diffusion in multilayer networks is a pioneering approach regarding the advent of omics data integration. Given that diffusion in classical networks suffers from biases, the next logical step is to characterise multilayer diffusion in the same terms.

10.2.2 Pathway analysis

The benefits of mining a knowledge graph for metabolomics data can be translated to other omics data.

Using a harmonised resource

Resources that aggregate and put several comprehensive databases in a common language, like PathMe, have demonstrated an improvement over single databases. The single database approach in this thesis can benefit from mining a richer, broader network.

A STATISTICAL PROPERTIES

A.1 SUPPLEMENT 1: MATHEMATICAL PROPERTIES

A.1.1 Introduction

This document outlines and proves several properties of the diffusion scores discussed in the main body. We mainly derive equivalences between scores and properties of the statistical normalisations and their null distributions.

Notation

Tables 14 and 15 contain an overview of the notation used to formulate and prove the properties.

Table 14: Notation of matrices, vectors and scalars.

Notation	Data type	Description
n	Scalar	Number of nodes in the graph
n_+	Scalar	Number of positive nodes
n_-	Scalar	Number of negative nodes
n_l	Scalar	Number of labelled nodes, $n_l = n_+ + n_-$
n_u	Scalar	Number of unlabelled nodes, $n_u = n - n_l$
L	Matrix	Unnormalised graph Laplacian $n \times n$ matrix
v_i	Column matrix	i-th eigenvector of L
v_i^T	Row matrix	i-th eigenvector of L
λ_i	Scalar	i-th eigenvalue of L
K	Matrix	Graph kernel $n \times n$ matrix from (Smola and Kondor, 2003)
$\mathcal{K}$	Matrix	$n \times n_l$ sub-matrix from K (columns indexed by labelled nodes)
K_{ij}	Scalar	Entry (i, j) from K
K_{i*}	Row matrix	i-th row of kernel matrix K
K_{*j}	Column matrix	j-th column of kernel matrix K
y_{raw}	Column matrix	Vector in $\mathbb{R}^n$ with the input scores to `raw` diffusion scores
$y_{raw}(i)$	Scalar	Input score of i-th node in the `raw` method
$\mathcal{y}_{raw}$	Column matrix	Vector in $\mathbb{R}^{n_l}$, y_{raw} restricted to labelled nodes only
y^+_{raw}	Scalar	Weight of the positive class for the `raw` scores
y^-_{raw}	Scalar	Weight of the negative class for the `raw` scores
y^u_{raw}	Scalar	Weight of the unlabelled class for the `raw` scores
f_{raw}	Column matrix	Vector with the `raw` scores
$f_{raw}(i)$	Scalar	`raw` score of i-th node
$\mathbb{1}_k$	Column matrix	Vector whose k entries are 1
I_k	Matrix	$k \times k$ identity matrix

The graph Laplacian L is a real $n \times n$ matrix defined as $L := D - W$, where $D = D_{ii}$ is the (diagonal) degree matrix and $W = W_{ij}$ the adjacency matrix. This definition assumes an undirected graph, either unweighted or with weights $W_{ij} \in [0, \infty)$ (Smola and Kondor, 2003).

This appendix reproduces the supplementary data (Supplements 1 to 4) of: Picart-Armada, Sergio, Wesley K. Thompson, Alfonso Buil, and Alexandre Perera-Lluna. "The effect of statistical normalisation on network propagation scores". *BioRxiv (2020)*.

Table 15: Notation of functions and operators.

Notation	Function/operator	Notes
$r(\lambda)$	Regularisation function (Smola and Kondor, 2003)	$\lambda \geqslant 0$, $r(\lambda) \geqslant 0$ monotonically increasing
$\frac{1}{r(\lambda)}$	Inverse of $r(\lambda)$	By convention, $0^{-1} \equiv 0$, see (Smola and Kondor, 2003)
$\mathbb{E}(X)$	Expected value	X random vector in $\mathbb{R}^k$, $\mathbb{E}(X) \in \mathbb{R}^k$, both column matrices
$\Sigma(X)$	Covariance	X as above, $\Sigma(X) \in \mathbb{R}^{k \times k}$ symmetric square matrix

Importantly, some of the present proofs use a sub-matrix $\mathcal{K}$ of the whole graph kernel K (Smola and Kondor, 2003) (we focus on the finite dimension case). $\mathcal{K}$ contains the **rows corresponding to all the graph nodes and the columns corresponding to labelled nodes**. In the absence of unlabelled nodes, $\mathcal{K} = K$ will be a square matrix, i.e. the whole kernel matrix – see for instance proposition 1. Otherwise $\mathcal{K}$ will be a rectangular sub-matrix of it, containing all the original rows but only some of the columns; the latter is not a kernel matrix properly speaking. This simplifies the notation because (i) only one score in our study, `gm`, actually places non-null weights on the unlabelled class, and (ii) the normalised scores permute only the labelled nodes.

Likewise, the input vector $\mathcal{y}$ represents y indexed by the labelled entries. For instance, the vector of input labels for the `raw` scores $\mathcal{y}_{raw}$ contains only the nodes under the positive ($y_{raw}(i) = y^{+}_{raw} = 1$) and negative classes ($y_{raw}(i) = y^{-}_{raw} = 0$), but not the unlabelled nodes.

Therefore, the matrix-vector product $f_{raw} = \mathcal{K}\mathcal{y}_{raw}$ is properly defined in terms of dimensionality. If n is the number of nodes in the graph and n_u the number of unlabelled nodes, then $f_{raw} \in \mathbb{R}^n$, $\mathcal{K} \in \mathbb{R}^{n \times (n - n_u)}$ and $\mathcal{y}_{raw} \in \mathbb{R}^{n - n_u}$. This is equivalent to including the unlabelled with a weight of $y^{u}_{raw} = 0$; this is, $f_{raw} = Ky_{raw}$, with $f_{raw} \in \mathbb{R}^n$, $K \in \mathbb{R}^{n \times n}$ and $y_{raw} \in \mathbb{R}^n$.

Definition of the scores

While the input can be quantitative, we focus on the case where the entities can only be positive, negative or unlabelled. The `raw` diffusion scores are defined as $f_{raw} = Ky_{raw} = \mathcal{K}\mathcal{y}_{raw}$, according to (Picart-Armada, Sergio and Thompson, Wesley K and Buil, Alfonso and Perera-Lluna, Alexandre, 2017). y_{raw} takes by default these weights: $y^{+} = 1$ for the positives, $y^{-} = y^{u} = 0$ for the negatives and unlabelled nodes. In general, a diffusion score f can be computed with other weights as $f = Ky$, where K is the graph kernel and y the input coded by another choice of y^{+}, y^{-} and y^{u}.

We use the diffuStats package (Picart-Armada, Sergio and Thompson, Wesley K and Buil, Alfonso and Perera-Lluna, Alexandre, 2017), equipped with the possibilities that are studied in the main body. One of the aims of the following properties is to prove equivalences between label choices in certain conditions are met.

A.1.2 Equivalences between scores

Proposition 1. *Consider* $f_{raw} = Ky_{raw}$ *of finite dimension in a scenario with no unlabelled nodes (*K *square matrix), i.e. the label of the* i*-th node* $y_i = y^+_{raw} = 1$ *for the positives and* $y_i = y^-_{raw} = 0$ *for the negatives, and* $n_u = 0$*. Let* $f = Ky$ *be another score using* $y^+ > y^-$ *as new real numbered weights for positives and negatives. Then, if the kernel* K *is a spectral transformation of the unnormalised graph Laplacian* L*, the result of ranking (prioritising) the nodes using* f_{raw} *and using* f *is identical.*

Proof. Let $f_{raw}(i_1) \geqslant f_{raw}(i_2) \geqslant \ldots \geqslant f_{raw}(i_n)$ be the ranking of the nodes using the f_{raw} scores, i.e. their prioritisation through decreasing f_{raw}, being the top suggestions the highest scores. If we prove that $f_{raw}(i) \geqslant f_{raw}(j) \Leftrightarrow f(i) \geqslant f(j)$, then the ranking using the scores f must be identical. Note that ties in f_{raw} must happen in f and vice versa, because $f_{raw}(i) = f_{raw}(j) \Leftrightarrow f_{raw}(i) \geqslant f_{raw}(j) \wedge f_{raw}(i) \leqslant f_{raw}(j) \Leftrightarrow f(i) \geqslant f(j) \wedge f(i) \leqslant f(j) \Leftrightarrow f(i) = f(j)$.

As K is a spectral transformation of the unnormalised Laplacian L, it can be written in the following form, see (Smola and Kondor, 2003):

$$K = \sum_{j=1}^{n} \frac{1}{r(\lambda_j)} v_j v_j^T$$

Being v_j the eigenvectors as column vectors and λ_j the eigenvalues of L. The constant vector $v_1 = \frac{1}{\sqrt{n}}(1, \ldots, 1)^T$ is an eigenvector of L of eigenvalue 0. Therefore, it is also an eigenvector of K of eigenvalue $\frac{1}{r(\lambda_1)} = \frac{1}{r(0)} \in \mathbb{R}$ (by convention, $\frac{1}{0} \equiv 0$). Therefore, the i-th row of K, denoted K_{i*}, $1 \leqslant i \leqslant n$, has a constant sum, as $Kv_1 = \frac{1}{r(0)} v_1$, and because $v_1 = \frac{1}{\sqrt{n}}(1, \ldots, 1)^T$:

$$\sum_{j=1}^{n} K_{ij} = \frac{1}{r(0)}, 1 \leqslant i \leqslant n$$

On the other hand,

$$\begin{aligned} f_{raw}(i) \geqslant f_{raw}(j) &\Leftrightarrow K_{i*} y_{raw} \geqslant K_{j*} y_{raw} \\ &\Leftrightarrow (K_{i*} - K_{j*}) y_{raw} \geqslant 0 \end{aligned}$$

Note that if $\mathbb{1}_n$ is the column vector full of ones, then

$$y_{raw} = \frac{1}{y^+ - y^-}(y - y^- \mathbb{1}_n)$$

where $y^+ - y^- > 0$. Therefore,

$$\begin{aligned} (K_{i*} - K_{j*}) y_{raw} \geqslant 0 &\Leftrightarrow (K_{i*} - K_{j*}) \frac{1}{y^+ - y^-}(y - y^- \mathbb{1}_n) \geqslant 0 \\ &\Leftrightarrow (K_{i*} - K_{j*})(y - y^- \mathbb{1}_n) \geqslant 0 \\ &\Leftrightarrow (K_{i*} - K_{j*}) y - y^- (K_{i*} - K_{j*}) \mathbb{1}_n \geqslant 0 \\ &\Leftrightarrow (K_{i*} - K_{j*}) y - y^- \left(\frac{1}{r(0)} - \frac{1}{r(0)} \right) \geqslant 0 \\ &\Leftrightarrow (K_{i*} - K_{j*}) y \geqslant 0 \\ &\Leftrightarrow f(i) \geqslant f(j) \end{aligned}$$

□

Note how a more general version of this property can be proved in a very similar way for quantitative y_{raw}, so that transformations of the kind $y = \alpha y_{raw} + \beta \mathbb{1}_n$, with $\alpha, \beta \in \mathbb{R}, \alpha > 0$, lead to the same node ranking as f_{raw}.

Corollary 1. *The scores f_{raw}, f_{ml} and f_{gm} lead to the same node ranking (prioritisation) if there are no unlabelled nodes and the kernel K is a spectral transformation of the unnormalised graph Laplacian L of finite dimension.*

Note that this result does not hold in general if the kernel stems from the normalised Laplacian, or with the presence of unlabelled nodes besides positives and negatives (see counterexamples on figure 38). In both cases, the row sums are no longer constant and their respective sums do not cancel out.

The same property holds if instead of having only positives and negatives there are only positive and unlabelled nodes:

Proposition 2. *Consider $f_{raw} = K y_{raw}$ of finite dimension in a scenario with no negative nodes, i.e. the label of the i-th node $y_i = y^+_{raw} = 1$ for the positives and $y_i = y^u_{raw} = 0$ for the unlabelled nodes, and $n_- = 0$. Let f be another score using $y^+ > y^u$ as new real numbered weights for positives and unlabelled nodes. Then, if the kernel K is a spectral transformation of the unnormalised graph Laplacian L, the result of ranking (prioritising) the nodes using f_{raw} and using f is identical.*

Proof. The proof is identical to proposition 1, but switching the roles of unlabelled and negative nodes. □

Corollary 2. *The scores f_{raw}, f_{ml} and f_{gm} lead to the same node ranking if there are no negative nodes and the kernel K is a spectral transformation of the unnormalised graph Laplacian L of finite dimension.*

Proposition 3. *The ranking using f_{raw} and f_{ber_s} is identical within the positive, the negative and the unlabelled nodes, provided that $\epsilon > 0$*

Proof. If the i-th node is a positive, then:

$$f_{ber_s}(i) = \frac{f_{raw}(i)}{y_{raw}(i) + \epsilon} = \frac{f_{raw}(i)}{1 + \epsilon}$$

Therefore, there is only a positive, multiplicative constant between f_{raw} and f_{ber_s}.

If the i-th node is a negative or unlabelled, then:

$$f_{ber_s}(i) = \frac{f_{raw}(i)}{y_{raw}(i) + \epsilon} = \frac{f_{raw}(i)}{\epsilon}$$

Where f_{raw} and f_{ber_s} clearly lead to the same ranking (prioritisation) again. □

A.1.3 Normalisations are invariant under label codification

Proposition 4. *In f_z, the choice of y^+ and y^- is irrelevant. More generally, computing $\tilde{f}_z$ using $\tilde{y} = \alpha y + \beta \mathbb{1}_{n_l}$ instead of y, with $\alpha, \beta \in \mathbb{R}, \alpha > 0$, is equivalent to computing f_z.*

Proof. Using the same notation as in the first property, we start from the definition of the z-score from the main text, which normalises the f_{raw} score by the mean and standard deviation of its distribution when the labelled nodes are permuted:

$$f_z(i) = \frac{f_{raw}(i) - \mathbb{E}(X_f(i))}{\sqrt{\mathrm{Var}(X_f(i))}} = \frac{\mathcal{K}_{i*} y_{raw} - \mathcal{K}_{i*}\mathbb{E}(X_y)}{\sqrt{\mathcal{K}_{i*}\Sigma(X_y)\mathcal{K}_{i*}^T}}$$

According to the main text, X_y is a random permutation of the labelled nodes in the input y_{raw}, and $X_f = \mathcal{K}X_y$ is the random vector of null diffusion scores.

We prove that computing f_z with an input vector y is identical to doing the same with $\tilde{y} = \alpha y + \beta \mathbb{1}_{n_l}$. Let X_y be a random permutation of the input y, treated as a random vector, and let $X_{\tilde{y}}$ be the same permutation applied to $\tilde{y}$. The raw diffusion score of the i-th node is $\mathcal{K}_{i*}y$, whereas its null distribution is the random variable $\mathcal{K}_{i*}X_y$. Analogously, using $\tilde{y}$, the score is $\mathcal{K}_{i*}\tilde{y}$ and the null distribution is $\mathcal{K}_{i*}X_{\tilde{y}}$. The idea is that subtracting the expected value cancels out the constant term β, whereas dividing by the standard deviation cancels out the multiplicative constant α.

For any node i:

$$\begin{aligned}
\tilde{f}_z(i) &= \frac{\mathcal{K}_{i*}\tilde{y} - \mathcal{K}_{i*}\mathbb{E}(X_{\tilde{y}})}{\sqrt{\mathcal{K}_{i*}\Sigma(X_{\tilde{y}})\mathcal{K}_{i*}^T}} \\
&= \frac{\mathcal{K}_{i*}(\alpha y + \beta \mathbb{1}_{n_l}) - \mathcal{K}_{i*}\mathbb{E}(\alpha X_y + \beta \mathbb{1}_{n_l})}{\sqrt{\mathcal{K}_{i*}\Sigma(\alpha X_y + \beta \mathbb{1}_{n_l})\mathcal{K}_{i*}^T}} \\
&= \frac{\alpha\mathcal{K}_{i*}y + \beta\mathcal{K}_{i*}\mathbb{1}_{n_l} - \mathcal{K}_{i*}(\alpha\mathbb{E}(X_y) + \beta\mathbb{1}_{n_l})}{\sqrt{\alpha^2\mathcal{K}_{i*}\Sigma(X_y)\mathcal{K}_{i*}^T}} \\
&= \frac{\alpha\mathcal{K}_{i*}y - \alpha\mathcal{K}_{i*}\mathbb{E}(X_y)}{|\alpha|\sqrt{\mathcal{K}_{i*}\Sigma(X_y)\mathcal{K}_{i*}^T}} \\
&= \frac{\mathcal{K}_{i*}y - \mathcal{K}_{i*}\mathbb{E}(X_y)}{\sqrt{\mathcal{K}_{i*}\Sigma(X_y)\mathcal{K}_{i*}^T}} \\
&= f_z(i)
\end{aligned}$$

□

As a consequence, the score f_z is independent from the choice of the label weights: not just the final ranking is the same, but the values of the scores are identical.

Proposition 5. *In f_{mc}, the choice of y^+ and y^- is irrelevant. More generally, computing $\tilde{f}_{mc}$ using $\tilde{y} = \alpha y + \beta \mathbb{1}_{n_l}$ instead of y, with $\alpha, \beta \in \mathbb{R}, \alpha > 0$, is equivalent to computing f_{mc}.*

Proof. Remember that f_{mc} is defined, for the i-th node, as:

$$f_{mc}(i) = 1 - \frac{r_i + 1}{N + 1}$$

This measures the amount of permutations (null trials) r_i, out of a total of N, in which $f_{raw}^{null}(i) \geqslant f_{raw}(i)$, where $f_{raw}^{null}(i)$ is the f_{raw} score of the i-th node using the permuted input y_{raw}^{null} instead of y_{raw}.

We will focus on the outcome of a single random trial among the N null trials, denoted by the superindex *null*. In other words, $f_{raw}^{null}(i)$ is the null score of the i-th node on this random trial. Let $\tilde{f}_{mc}(i)$ be the scores computed from $\tilde{y}$ instead of y. It suffices to prove that $f_{raw}^{null}(i) \geqslant f_{raw}(i) \Leftrightarrow \tilde{f}_{raw}^{null}(i) \geqslant \tilde{f}_{raw}(i)$. If the former is true for any permutation, then $\tilde{r}_i = r_i$ (the same N random trials would lead to the same estimate), thus $\tilde{f}_{mc}(i) = f_{mc}(i)$.

$$\begin{aligned}
\tilde{f}_{raw}^{null}(i) \geqslant \tilde{f}_{raw}(i) &\Leftrightarrow \mathcal{K}_{i*}\tilde{y}_{raw}^{null} \geqslant \mathcal{K}_{i*}\tilde{y}_{raw} \\
&\Leftrightarrow \mathcal{K}_{i*}(\tilde{y}_{raw}^{null} - \tilde{y}_{raw}) \geqslant 0 \\
&\Leftrightarrow \mathcal{K}_{i*}(\alpha y_{raw}^{null} + \beta \mathbb{1}_{n_l} - \alpha y_{raw} - \beta \mathbb{1}_{n_l}) \geqslant 0 \\
&\Leftrightarrow \mathcal{K}_{i*}\alpha(y_{raw}^{null} - y_{raw}) \geqslant 0 \\
&\Leftrightarrow \mathcal{K}_{i*}(y_{raw}^{null} - y_{raw}) \geqslant 0 \\
&\Leftrightarrow f_{raw}^{null}(i) \geqslant f_{raw}(i)
\end{aligned}$$

□

As with f_z, the definition of f_{mc} conveniently avoids the choice of weights for positives and negatives. Also note that propositions 4 and 5 hold for kernels based on the normalised and the unnormalised graph Laplacian. See figure 38 for examples on both properties.

A.1.4 Expected values and covariance matrix of null scores

The following property provides the closed expressions of the null expected vector and covariance matrix. For instance, these are useful for computing f_z without the need of permutations to estimate the expected values and variances.

Proposition 6. *Let f_{raw} be the raw diffusion scores computed from a kernel $\mathcal{K}$ and an input y_{raw}. Let X_f be the random vector of diffusion scores computed from a permuted input, $X_f = \mathcal{K}X_y$, where X_y is the random vector that results from permuting (shuffling) y_{raw}, i.e. $X_y = \pi(y_{raw})$ for a random permutation π. Then, if $M_k = I_k - \frac{1}{k}\mathbb{1}_k\mathbb{1}_k^T$ and $n_l \geqslant 2$:*

(i) $\mathbb{E}(X_y) = \mu_y \mathbb{1}_{n_l}$

(ii) $\mathbb{E}(X_f) = \mu_y \mathcal{K}\mathbb{1}_{n_l}$

(iii) $\Sigma(X_y) = \sigma_y^2 M_{n_l}$

(iv) $\Sigma(X_f) = \sigma_y^2 \mathcal{K} M_{n_l} \mathcal{K}^T$

being $\mu_y = \frac{1}{n_l}\sum_{i=1}^{n_l} y_i$ *the mean of the labels and* $\sigma_y^2 = \frac{1}{n_l-1}\sum_{i=1}^{n_l}(y_i - \mu_y)^2$ *their variance.*

Proof. (i) $\mathbb{E}(X_y)$ is, by symmetry, a constant n_l-th dimensional vector, so we can write $\mathbb{E}(X_y) = \mu\mathbb{1}_{n_l}$ for some $\mu \in \mathbb{R}$. Under the permutations, all the elements of the original vector have a uniform probability of ending in a given position, therefore $\mu = \frac{1}{n_l}\sum_{i=1}^{n_l} y_i = \mu_y$

(ii) Using property (i), $\mathbb{E}(X_f) = \mathbb{E}(\mathcal{K}X_y) = \mathcal{K}\mathbb{E}(X_y) = \mu_y\mathcal{K}\mathbb{1}_{n_l}$

(iii) By the symmetry of the permutations, the covariance matrix $\Sigma(X_y)$ can only have two different elements: (1) the variances σ^2 on the diagonal, and (2) the covariances ρ on the off-diagonal. This can be written as:

$$\Sigma(X_y) = (\sigma^2 - \rho)I_{n_l} + \rho\mathbb{1}_{n_l}\mathbb{1}_{n_l}^T$$

To find σ^2, it suffices to see that σ^2 is actually the (exact) variance of each position in the permuted vector, i.e. $\sigma^2 = \frac{1}{n_l}\sum_{i=1}^{n_l}(y_i - \mu_y)^2 = \sigma_y^2\frac{n_l-1}{n_l}$.

To find the covariance between two positions ρ, it is useful to notice that $\mathbb{1}_{n_l}^T y = n_l\mu_y = \mathbb{1}_{n_l}^T X_y$ is constant, because the elements of X_y must have a constant sum regardless of the permutation. Therefore, its covariance is 0:

$$\Sigma(\mathbb{1}_{n_l}^T X_y) = 0 = \mathbb{1}_{n_l}^T\Sigma(X_y)\mathbb{1}_{n_l}$$

Using this, and knowing that $\mathbb{1}_{n_l}^T\mathbb{1}_{n_l} = n_l$,

$$\mathbb{1}_{n_l}^T\left[(\sigma^2 - \rho)I_{n_l} + \rho\mathbb{1}_{n_l}\mathbb{1}_{n_l}^T\right]\mathbb{1}_{n_l} = 0$$

$$(\sigma^2 - \rho)n_l + \rho n_l^2 = 0$$

$$\rho = \sigma^2\frac{-1}{n_l - 1}$$

Therefore, the desired covariance matrix is

$$\Sigma(X_y) = (\sigma^2 - \rho)I_{n_l} + \rho\mathbb{1}_{n_l}\mathbb{1}_{n_l}^T = \sigma^2\left[\frac{n_l}{n_l-1}I_{n_l} - \frac{1}{n_l-1}\mathbb{1}_{n_l}\mathbb{1}_{n_l}^T\right] = \\ = \sigma^2\frac{n_l}{n_l-1}M_{n_l} = \sigma_y^2 M_{n_l}$$

(iv) Using property (iii), $\Sigma(X_f) = \Sigma(\mathcal{K}X_y) = \mathcal{K}\Sigma(X_y)\mathcal{K}^T = \sigma_y^2\mathcal{K}M_{n_l}\mathcal{K}^T$

□

Note that these statistical moments are valid in the general case where y_{raw} is quantitative. Another remark is that the proofs for propositions 4, 5 and the statistical moments in proposition 6 do not actually use that $\mathcal{K}$ comes from a kernel. Therefore, they stand **valid in random-walk and other approaches that can be defined as a matrix-vector product but fall outside the kernel formalism**.

Corollary 3. *In the absence of unlabelled nodes and using a kernel from a spectral transformation of the unnormalised Laplacian, the expected values of all the nodes is coincidental.*

Proof. As shown in proposition 1, if the unnormalised Laplacian is used, the kernel rows have the same sum because $\mathbb{1}_n$ is an eigenvector of $\mathcal{K} = K$ with eigenvalue $\frac{1}{r(0)}$. Using property (ii) from proposition 6 and the fact that there are no unlabelled nodes, i.e. $n_l = n$, the vector with expected values becomes

$$\mathbb{E}(X_f) = \mu_y K \mathbb{1}_n = \mu_y \frac{1}{r(0)} \mathbb{1}_n$$

Therefore, all the expected value are coincidental and equal to $\mu_y \frac{1}{r(0)}$. □

This implies that f_z, in fact, modifies the ranking of f_{raw} only because of different standard deviations under these conditions. This does not hold in the general case with a nonempty set of unlabelled nodes.

On the other hand, property (iv) in proposition 6 has an implication in understanding the covariance of the null distribution:

Proposition 7. *The covariance of the null distribution of diffusion scores is directly related to the covariance between the kernel vectors. Specifically, the covariance between two nodes is, up to a multiplicative constant, their sample covariance using as samples the labelled nodes and as features their kernel values to all the nodes.*

Proof. Starting from point (iv) in proposition 6, $\Sigma(X_f) = \sigma_y^2 \mathcal{K} M_{n_l} \mathcal{K}^T$, note that $M_k M_k = I_k - 2\frac{1}{k}\mathbb{1}_k\mathbb{1}_k^T + \frac{1}{k^2}\mathbb{1}_k k \mathbb{1}_k^T = I_k - \frac{1}{k}\mathbb{1}_k\mathbb{1}_k^T = M_k$. Also, by its own definition, M_k is the matrix such that it centers the rows (resp. columns) of a matrix $A \in \mathbb{R}^{k\times k}$ when it is multiplied by the right (resp. left) of A.

Back to the expression of $\Sigma(X_f)$, we can write:

$$\Sigma(X_f) = \sigma_y^2 \mathcal{K} M_{n_l} \mathcal{K}^T = \sigma_y^2 \mathcal{K} M_{n_l} M_{n_l} \mathcal{K}^T = \sigma_y^2 \hat{\mathcal{K}} \hat{\mathcal{K}}^T$$

Being $\hat{\mathcal{K}} = \mathcal{K} M_{n_l}$ the row-centered matrix $\mathcal{K}$. This implies that the product $\hat{\mathcal{K}}\hat{\mathcal{K}}^T$ is just the sample covariance of $\mathcal{K}^T$, up to a multiplicative constant, and therefore the whole covariance $\Sigma(X_f)$ is proportional to the sample covariance of $\mathcal{K}^T$. □

To illustrate this, a feature matrix can be built from the kernel matrix, using as *samples* the labelled nodes in the network and as *features* their similarity, using the kernel, to all the network nodes.

On one hand, the sample covariance of such a dataset (matrix whose (i,j)-th entry is the covariance between features i and j) is proportional to the covariance of the null distribution in the diffusion process.

On the other hand, the leading eigenvectors of the null covariance are equivalent to the principal components (loadings) of such a dataset.

In turn, these eigenvectors are related to the spectral properties of the graph, such as the Fiedler-vector in graph partitioning (Smola and Kondor,

2003). The Fiedler-vector is defined as the eigenvector v_2 of the graph Laplacian with the second smallest eigenvalue λ_2. v_2 is also an eigenvector of any graph kernel as defined in (Smola and Kondor, 2003) with eigenvalue $\frac{1}{r(\lambda_2)}$.

To wrap up these properties, we prove a particular case in which the leading eigenvectors of the null covariance convey the same data as the Fiedler-vector and its successive components.

Proposition 8. *Let* K *be a kernel from a spectral transformation* $r(\lambda) > 0$ *on the unnormalised graph Laplacian* L*, of finite dimension. In the absence of unlabelled nodes, i.e.* $n_u = 0$ *and* $n_l = n$*, then the eigenvector* v_{i+1} *with the* $i+1$*-th smallest eigenvalue of* L*,* $1 \leqslant i \leqslant n-1$*, is equal to the eigenvector* u_i *with* i*-th largest eigenvalue from the null covariance* $\Sigma(X_f)$*.*

Proof. We use the definition of the kernel $K = \sum_{j=1}^{n} \frac{1}{r(\lambda_j)} v_j v_j^T$ (Smola and Kondor, 2003), the definition of $\hat{K} = KM_n$ from property 8 and that of $\Sigma(X_f) = \sigma_y^2 \hat{K}\hat{K}^T$ from property 6. Note that the matrices L, K, M_n, $\hat{K}$, $\Sigma(X_f)$ belong to $\mathbb{R}^{n\times n}$ and, in particular, are square. To build the proof, we will write the eigenspaces of such matrices in ascending eigenvalues.

We denote the eigenvectors of L as $v_1, v_2, \ldots, v_n$, with the following eigenvalues: $0 = \lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n$ Without loss of generality, we can take $v_1 = \frac{1}{\sqrt{n}}\mathbb{1}_n = \frac{1}{\sqrt{n}}(1, \ldots, 1)^T$ – that does not apply to the normalised Laplacian in general.

Because $r(\lambda) > 0$ and $r(\lambda)$ is increasing, the eigenspace of K consists of the eigenvectors in reversed order $u_1 = v_n, \ldots, u_n = v_1$, with eigenvalues $\frac{1}{r(\lambda_n)} \leqslant \ldots \leqslant \frac{1}{r(\lambda_2)} \leqslant \frac{1}{r(0)}$

Regarding M_n, we show that its eigenspace is $w_1 = v_1, \ldots, w_n = v_n$, with eigenvalues $0, 1, \ldots, 1$. On one hand, $M_n v_1 = v_1 - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T v_1 = v_1 - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\frac{1}{\sqrt{n}}\mathbb{1}_n = v_1 - \frac{1}{n}\mathbb{1}_n\sqrt{n} = v_1 - v_1 = 0$. On the other hand, if $1 < i \leqslant n$, $M_n v_i = v_i - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T v_i = v_i - 0 = v_i$, because v_i is orthogonal to v_1, a multiple of $\mathbb{1}_n$. We conclude that v_i has eigenvalue 1.

The eigensystem of $\hat{K} = KM_n$ can be characterised using that K and M_n share all the eigenvectors. The eigenvalues of $\hat{K}$ are the product of the respective eigenvalues of K and M_n. The eigenvectors $t_1 = v_1, t_2 = v_n, t_3 = v_{n-1}, \ldots, t_n = v_2$ have as eigenvalues $0 \leqslant \frac{1}{r(\lambda_n)} \leqslant \frac{1}{r(\lambda_{n-1})} \leqslant \ldots \leqslant \frac{1}{r(\lambda_2)}$, i.e. the leading eigenvalue of K, $\frac{1}{r(0)}$, has now collapsed to 0, whereas the rest are unchanged.

The proof is completed by pointing out that the eigenvectors of $\Sigma(X_f) = \sigma_y^2 \hat{K}\hat{K}^T$ are those of $\hat{K}$, and that their eigenvalues are $0 \leqslant \sigma_y^2 \frac{1}{r(\lambda_n)^2} \leqslant \sigma_y^2 \frac{1}{r(\lambda_{n-1})^2} \leqslant \ldots \leqslant \sigma_y^2 \frac{1}{r(\lambda_2)^2}$. The order of the eigenvectors and the eigenvalues is preserved because $\sigma_y^2 \geqslant 0$ and $r(\lambda) > 0$. From here, the vector with the largest eigenvalue from $\Sigma(X_f)$, t_n, is equal to v_2, the eigenvector with second smallest eigenvalue from L; the second largest is $t_{n-1} = v_3$, the eigenvector with the third smallest eigenvalue from L, and so on.

□

Corollary 4. *On a graph with the same premises as in proposition 8, provided that* $\lambda_2 < \lambda_3$*, the leading eigenvector of the null covariance* $\Sigma(X_f)$ *is the Fiedler-vector, up to a change of sign.*

Proof. The only observation is that, given that $\lambda_2 < \lambda_3$, the Fiedler-vector is unique and, by property 8, is the leading vector of the null covariance, up to a sign change. □

Proposition 8 is illustrated in toy graphs with the lattice (figure 39) and the Barabási-Albert (figure 40) architectures. The presence of unlabelled nodes leads to a leading covariance eigenvector analogous to the Fiedler-vector, but taking into account the unobservable nature of part of the network.

In views of these results, the null covariance of a given instance can be of interest per se, because it reflects the effect of measuring (and normalising) a specific set of nodes in terms of spectral network properties.

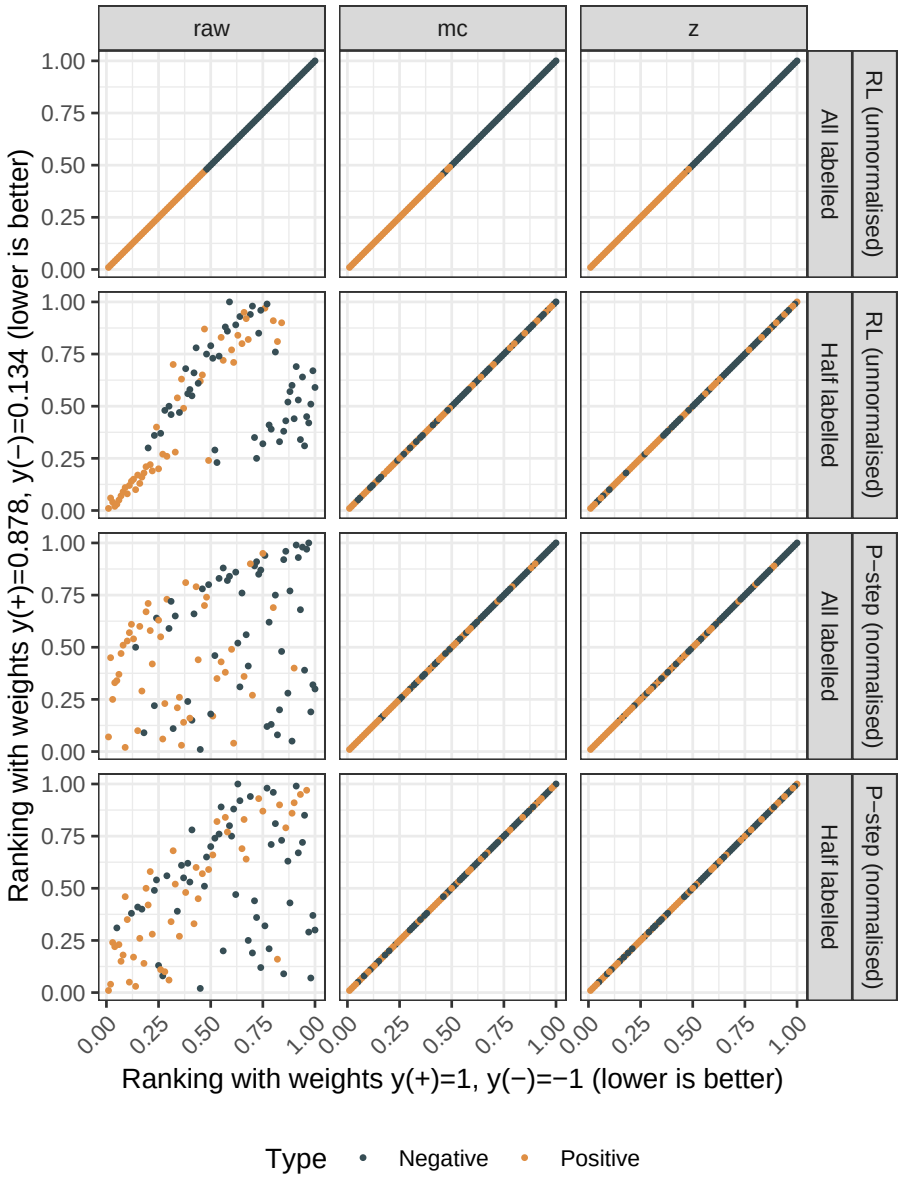


Figure 38: Effect of label encoding on f_{raw}, f_{mc} and f_z, depending on (i) the kernel, and (ii) presence of unlabelled nodes. A small graph of order 100 was generated with `igraph::barabasi.game(n = 100, m = 3, directed = F)`. The 100 nodes were assigned to the positive and negative classes once, with $\frac{1}{2}$ probability each. Those labels were either all available or half available (only the first 50), considering the other half as unlabelled. Two encodings were used to rank the nodes: $y^+ = 1$, $y^- = -1$ (like f_{ml}) and two random weights $y^+ > y^-$. Two kernels were compared, one from the normalised Laplacian (p-step kernel, $\alpha = 2$, $p = 5$) and one from the unnormalised (RL: regularised Laplacian, $\sigma^2 = 1$). f_{raw}: both encodings are equivalent only for the RL kernel without unlabelled nodes, consistent with proposition 1. f_z and f_{mc}: as stated in propositions 4 and 5, both scores are weight-independent, regardless of the kernel and the presence of unlabelled nodes.

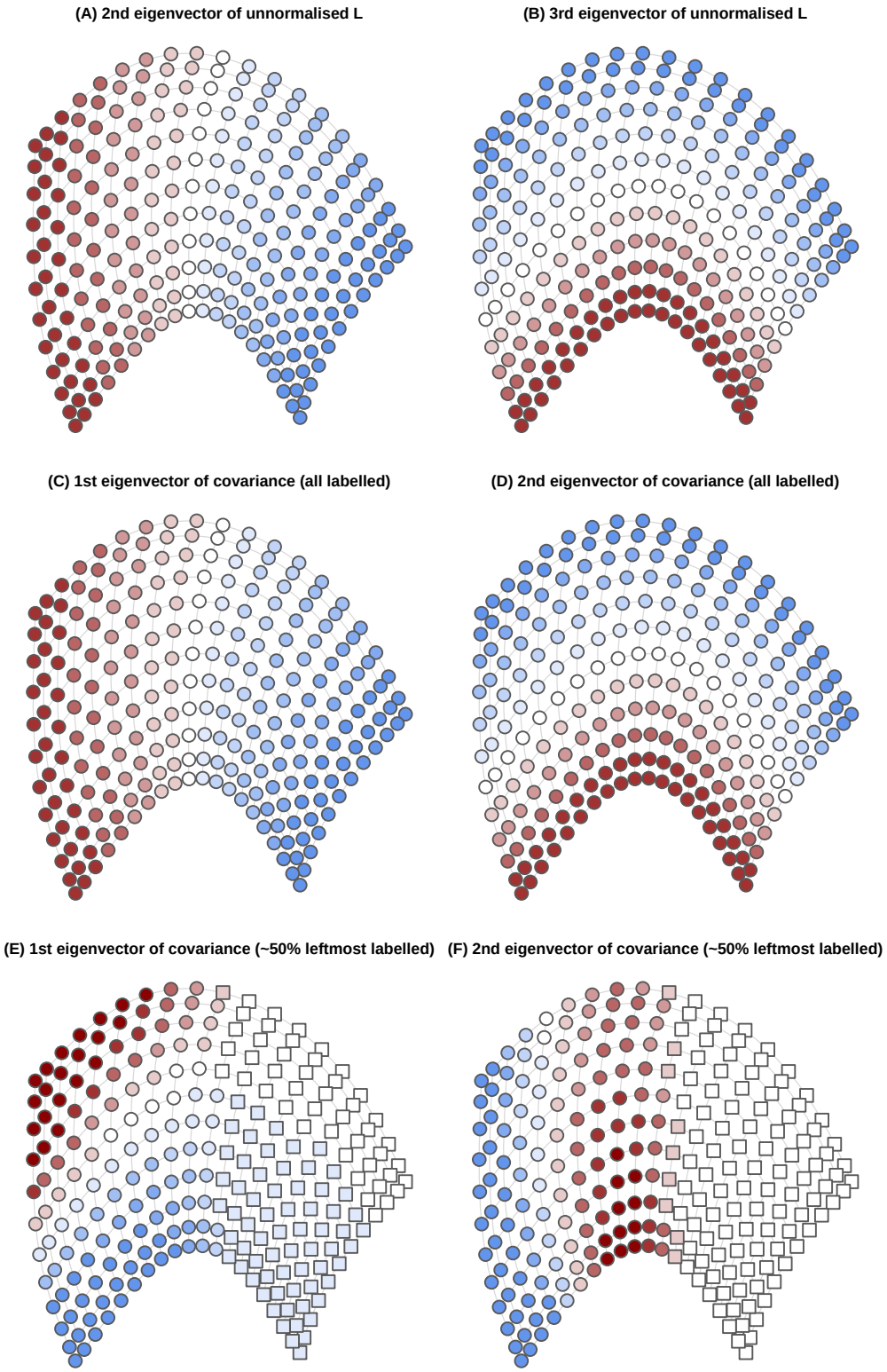


Figure 39: Comparison of the eigenvectors of the unnormalised Laplacian (panels **A-B**) and the null covariance using the regularised Laplacian kernel (panels **C-F**), on a toy lattice graph of 20×12 nodes, `igraph::graph.lattice(dimvector = c(20, 12))` in R. The null covariance is computed in two scenarios: all the nodes are labelled, i.e. $\mathcal{K} = \mathsf{K}$ (panels **C-D**), or only half of them are, so $\mathcal{K}$ contains half of the columns in K (panels **E-F**).

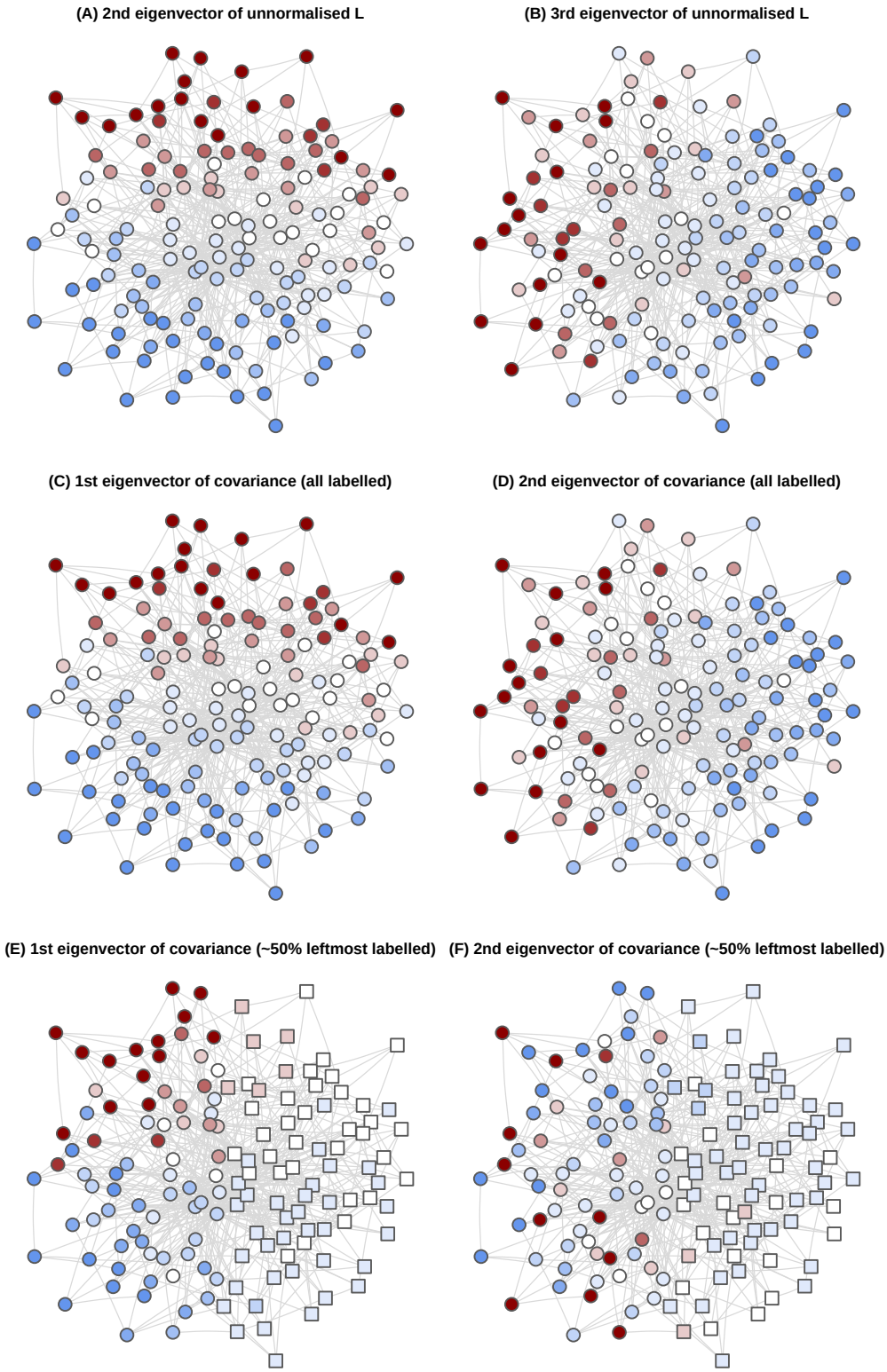


Figure 40: Comparison of the eigenvectors of the unnormalised Laplacian (panels **A-B**) and the null covariance using the regularised Laplacian kernel (panels **C-F**), on a toy synthetic Barabási graph, `igraph::barabasi.game(n = 150, m = 4, directed = F)` in R. The null covariance is computed in two scenarios: all the nodes are labelled, i.e. $\mathcal{K} = \mathsf{K}$ (panels **C-D**), or only half of them are, so $\mathcal{K}$ contains half of the columns in K (panels **E-F**).

A.2 SUPPLEMENT 2: SYNTHETIC SIGNALS

A.2.1 Introduction

This additional file contains details on the synthetic signals generated on a yeast interactome, both in biased and unbiased ways. Using a controlled environment, we characterised the behaviour of the diffusion scores to derive guidelines in terms of two key factors: the presence of bias in the positives and the class imbalance. This document can be re-built anytime by knitting its corresponding `.Rmd` file.

The network

The yeast interactome was originally published in (Von Mering et al., 2002) and downloaded using the `igraphdata` R package (Csardi, 2015). Only the largest connected component was used, which consisted of 2375 nodes and 1.1693×10^4 edges. A summary of the network is provided below:

```
## IGRAPH 3d30c0a UN-- 2375 11693 -- Yeast protein interactions, von Mering e
## + attr: name (g/c), Citation (g/c), Author (g/c), URL (g/c),
## | Classes (g/x), name (v/c), Class (v/c), Description (v/c),
## | Confidence (e/c)
```

Synthetic signal generation

Biased and unbiased signals were generated in order to compare normalised and unnormalised diffusion scores. As shown in the diffusion scores properties in Supplement 1, if all the nodes are labelled and the regularised unnormalised Laplacian kernel is used, then the expected values of the null distribution are constant for all the nodes in the network. In order to have differences in expected values (and therefore noticeable biases), nodes were randomly divided in three classes:

- Labelled nodes: the labelled nodes in the input
- Target nodes: the unlabelled nodes that had to be prioritised
- Filler nodes: the rest of unlabelled nodes

The presence of filler and target nodes, considered as unlabelled in the diffusion inputs, promoted differences in the expected values of all nodes. Each class contained around one third of the nodes:

```
##
##   Filler Labelled   Target
##      793      791      791
```

The purpose was to sample $n_{labelled}$ nodes from the labelled nodes and n_{target} nodes from the target nodes in each instance. The sampled nodes were deemed positives, whereas the rest were negatives. Diffusion scores were fed with the labelled nodes in order to prioritise the target nodes, on which the performance metrics were computed.

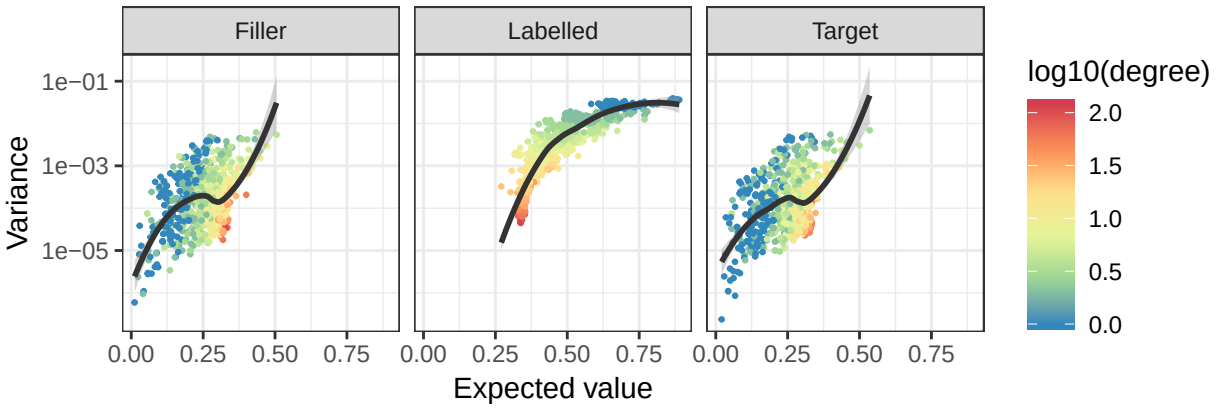


Figure 41: Expected value, variance and node degree in every node category. Loess fit in black, shaded with 0.95 confidence interval. Note how the effect of node degree on its expected value had opposed directions in labelled and unlabelled nodes. Differences were also present in the magnitude of expected values, variances, and their trend.

BIASED SAMPLING First, the $n_{labelled}$ nodes were uniformly sampled from the labelled nodes, giving a binary input vector y. Then, the raw scores were computed: $f_{raw} = Ky$. Exactly n_{target} nodes were sampled, where the probability of the i-th node was proportional to $f_{raw}(i)$. This sampling scheme was biased because, by hypothesis, nodes with higher expected value would become positives more frequently.

UNBIASED SAMPLING Like in the biased sampling, the $n_{labelled}$ nodes were uniformly sampled to obtain the binary input vector y. The n_{target} nodes were sampled with a probability proportional to $f_{mc}(i) + \frac{1}{N+1}$, where $N = 10^4$ is the number of simulations. $f_{mc}(i)$ was (roughly) their empirical cumulative distribution function applied to the scores $f_{raw} = Ky$, which removed the bias by its own definition.

A.2.2 Descriptive statistics

Expected value and covariance matrices

After defining the node classes and the basic input parameters, we computed the theoretical mean vector and covariance matrix. The fact that the number of positives in the input was constant in these simulations led to fixed μ_y and σ_y^2 values, allowing a single representation of the expected values and variances of the null distributions in figure 41. The figure confirms that *labelled* nodes exhibited properties different that those of *filler* and *target* nodes: *labelled* nodes had higher expected values, variances, and different trends between expected value, variance, and degree. Likewise, *filler* and *target* nodes were undistinguishable, expected by their definition.

Figure 42 offers a closer look at differences in reference mean values the *target* nodes, which is a property of the network. The *target* nodes were of special interest because predictions and performance metrics were computed on them.

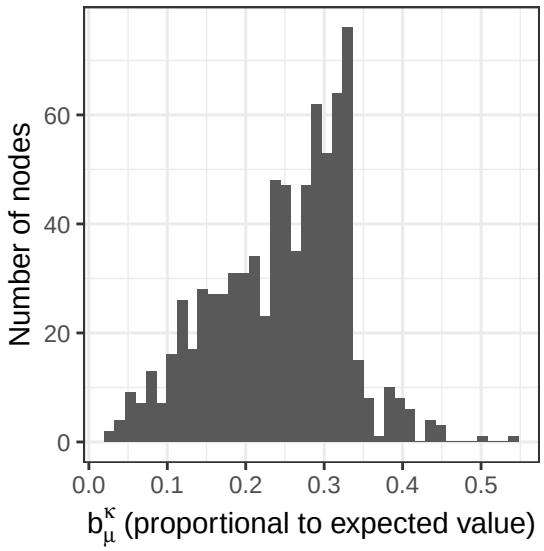


Figure 42: Histogram of the reference expected value of the target nodes.

Input lists

A total of 100 biased and 100 unbiased instances were generated, each with a proportion of 0.1 labelled nodes and a proportion of 0.1 target nodes with positive labels. To generate the unbiased inputs, `mc` scores were computed by permuting 10^4 times. The regularised (unnormalised) Laplacian kernel was used.

The frequency of target nodes and the reference expected value were expected to be uncorrelated in the unbiased signals, whilst positively correlated in the biased case. By definition, the input nodes should be independent from the reference expected value as well. Figure 43 supports all the claims above.

Diffusion scores

10^4 permutations were used to compute `mc` and `ber_p` scores. Figure 44 compares the rankings from each method, stratified by positives and negatives, and shows their correlation. This suggests groups of methods with similar behaviours: (i) `ml` and `gm`, or (ii) `ber_p`, `mc` and `z`. Also, top ranked `raw` nodes were usually top ranked in `z`, but the converse was not true.

Figure 45 depicts the correlations between methods. First, this shows how `ber_s` is equivalent to `raw` in terms of ranking, as proven in the properties in Supplement 1. `raw` correlates with normalised scores `mc` and `z`, the hybrid `ber_p` and `pagerank`. On the other hand, `pagerank` strongly anticorrelates with `ml` and `gm` (`raw` does as well, but only slightly). The scores `ml` and `gm` diffuse -1 on the negatives, which outnumber the positives 9 to 1 and dominate them. The nodes are expected to be ranked roughly by the (negative) reference expected value, also correlated with `pagerank`. This is supported by the strong anticorrelation between the node ranking (`ml`, `gm`) and `pagerank`.

Figure 46 depicts the concordance between the top 10 ranked nodes under each diffusion score. This scenario is slightly different from that in figure 45

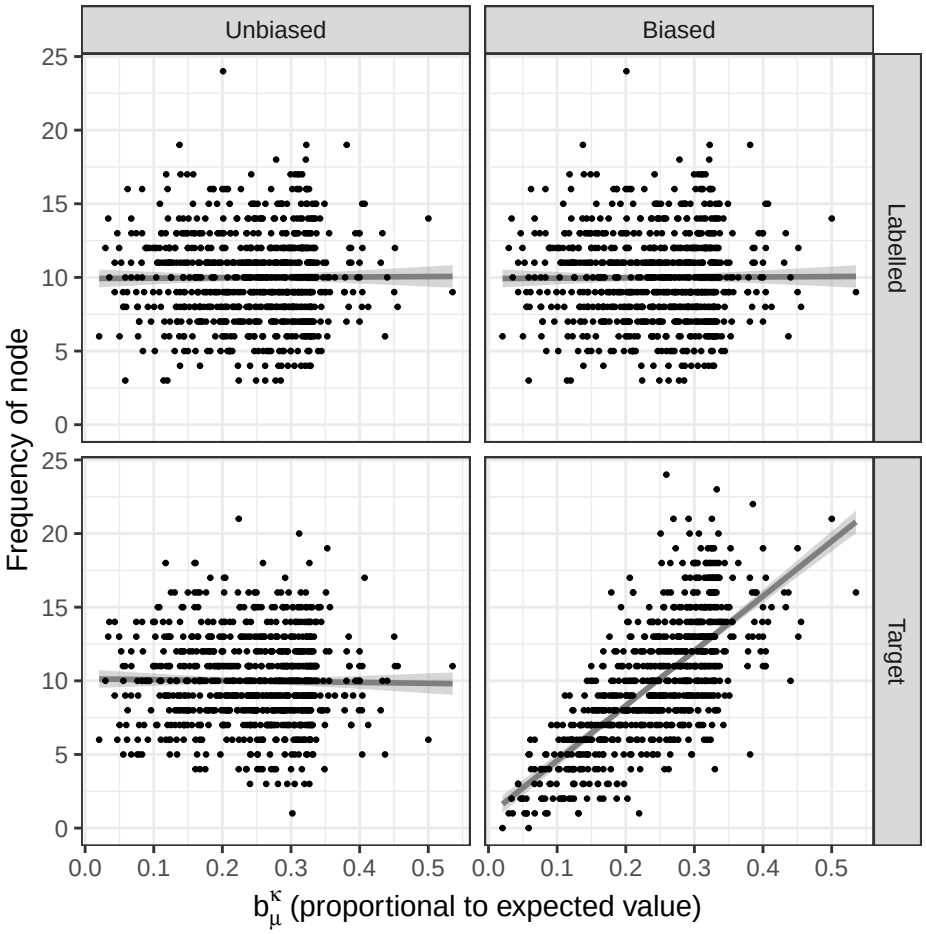


Figure 43: Frequency of positive nodes among the targeted nodes, as a function of the node reference expected value. Gray lines correspond to linear models with a 0.95 confidence interval.

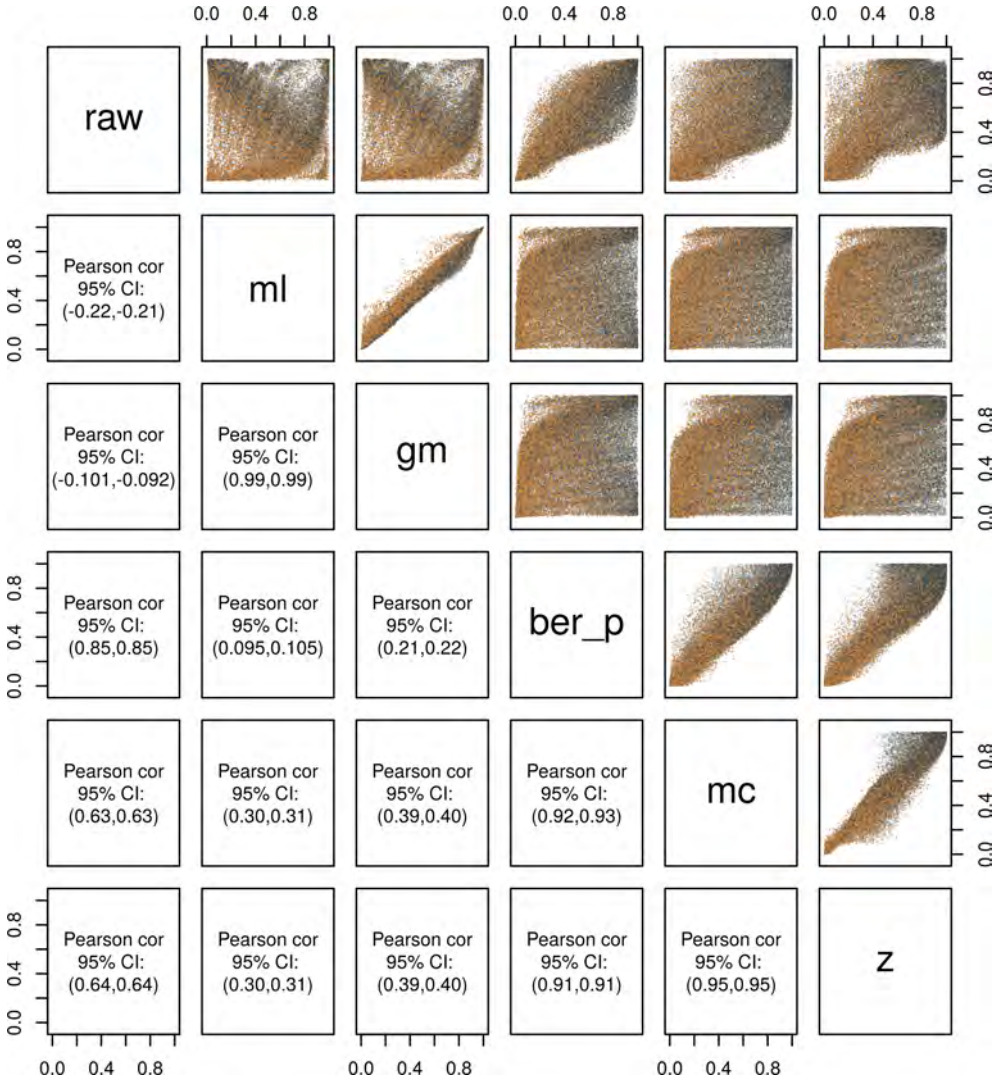


Figure 44: Pairs plot between the rankings by each diffusion score (and baseline). Top-ranked nodes are closer to 0. Positives and negatives are represented in orange and gray. The color legend has an adjusted transparency that corrects the fact that negatives greatly outnumbered positives.

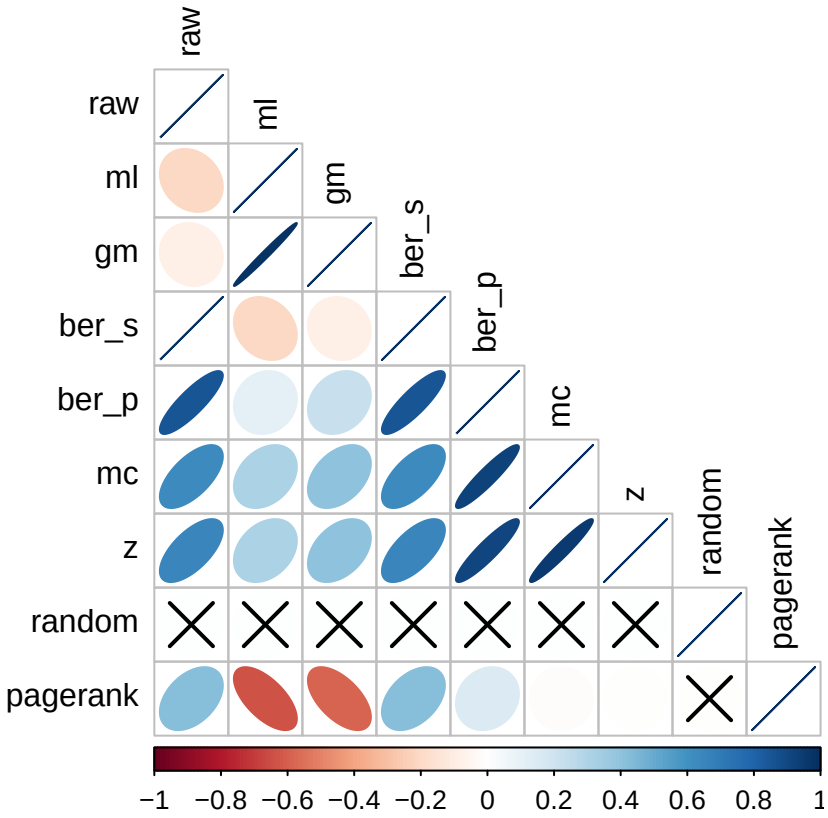


Figure 45: Spearman correlation between node rankings for all the diffusion scores and baselines. Crossed correlations had false discovery rates larger than 0.05.

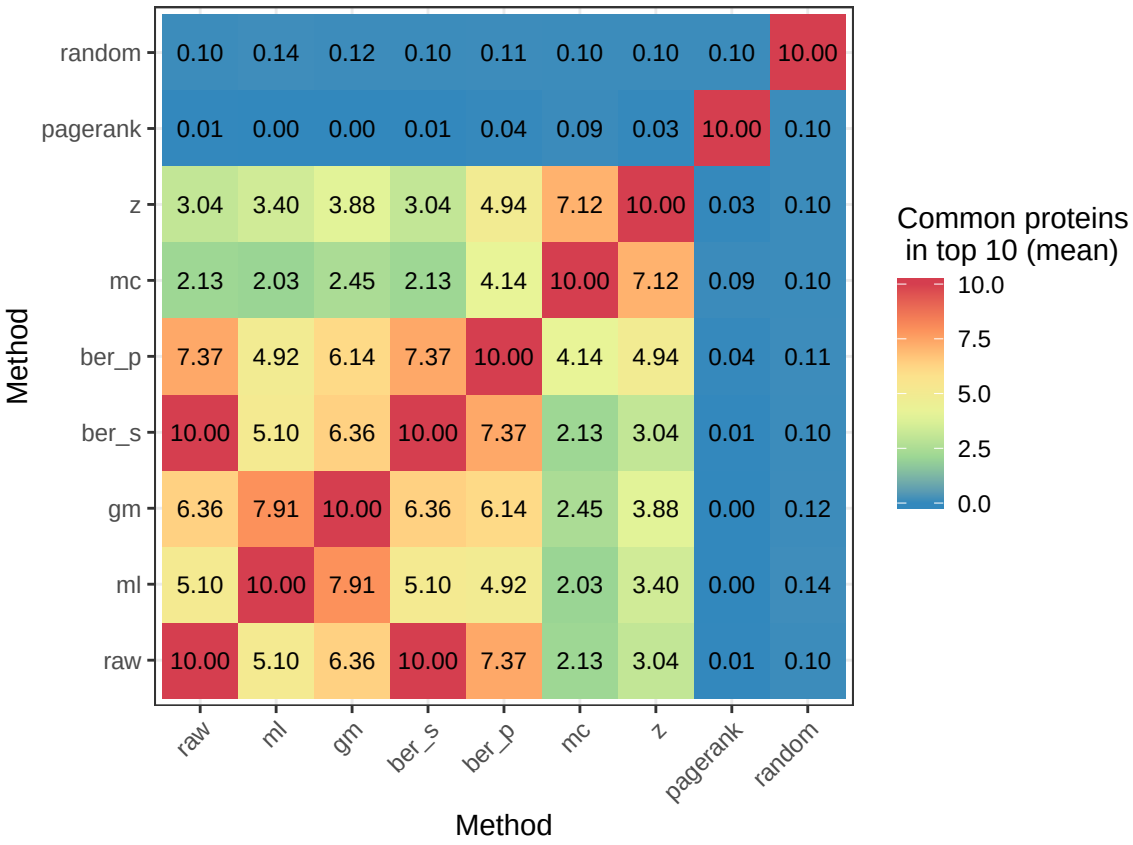


Figure 46: Common hits within the top 10 suggestions of all methods.

and suggests that methods with highest similarity are (i) `raw` and `ber_p`, (ii) `ml` and `gm`, (iii) `mc` and `z`.

Bias within prioritisations

Our main hypothesis on the fundamental impact of normalising the scores (`mc`, `z` versus `raw`) was that normalisation attained a more uniform power across the nodes of the network. In other words, unnormalised scores kept a higher power on a certain kind of nodes, driven by the reference expected value $b_\mu^{\mathcal{K}}$ in the null distribution. Figure 47 illustrates this behaviour, present in biased and unbiased signals: positives with high $b_\mu^{\mathcal{K}}$ were top ranked by `raw`, at the expenses of missing positives with low $b_\mu^{\mathcal{K}}$. However, the overall impact on performance was not obviously derived from figure 47 alone, because we needed to account for the density of true positives across the reference expected value (i.e. "are the positive nodes biased?"), as shown in figure 43.

Other remarks from figure 47: `ber_p` behaves halfway between `raw` and `mc`; and `ml` is biased in the other direction, that is, favouring nodes with a low reference expected value. The latter relates with the prior observations on how `ml` anticorrelates with `pagerank` because it diffuses -1 on the negatives, which outnumber the positives. Figure 47 casts doubt on the peformance of `ml` because the mean ranking of positives and negatives is qualitatively indistinguishable at the low reference expected value region, where `ml` should excel.

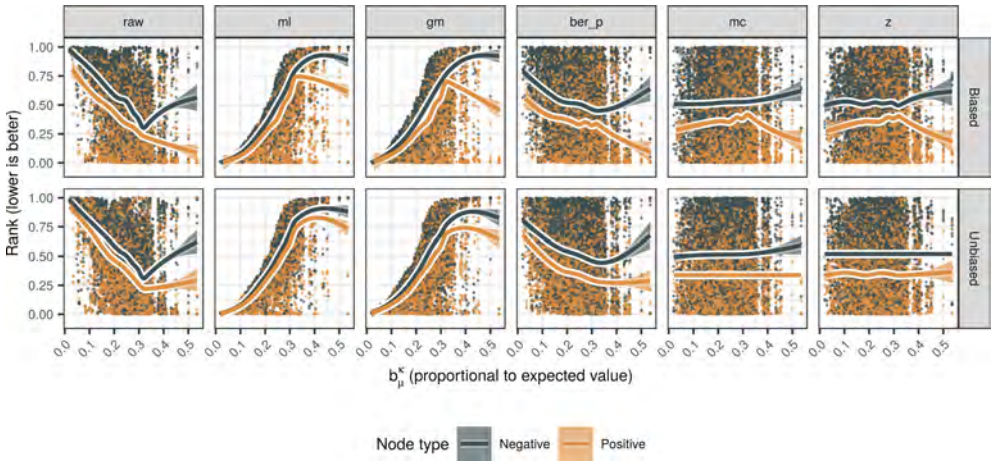


Figure 47: Ranking of the positives and negatives within the target nodes as a function of their reference expected value, divided by method and signal bias. Best rankings are those close to 0. The smoothing was fitted using the default gam method in ggplot2. For visual and computational purposes, only a fraction of 0.1 of the negatives were represented.

A.2.3 Performance

Aditive model

The AUROC and AUPRC were computed for each diffusion score and input, with its corresponding simulated ground truth (figure 48). Each box contained 100 data points. Differences were described in terms of the following additive quasibinomial (logit link) model, summarised in table 16:

$$\text{performance} \sim \text{method} + \text{method:biased} + \text{metric}$$

Provided that AUROC and AUPRC showed similar trends (figure 48) and that both range between 0 and 1, they were combined and modelled with the `metric` categorical covariate. `biased` referred to the nature of the signal, biased or unbiased.

Figure 48 suggests that the unnormalised scores `raw` were preferrable if the signal was biased, whereas `mc` and `z` were best suited for unbiased signals. Likewise, the hybrid scores `ber_p` standed out as a good compromise between both.

Table 17 contains confidence intervals on the predictions of the model for each combination of factors.

We tested for differences between the predictions of `raw` and `z` in the four cases using Tukey's method, confirming that the differences discussed above were statistically significant:

```
##   metric                  contrast   estimate         SE  df   z.ratio
## 1  AUPRC     Biased,raw - Biased,z  0.2657691 0.01824644 Inf  14.56553
## 2  AUPRC Unbiased,raw - Unbiased,z -0.2006233 0.01836834 Inf -10.92223
## 3  AUROC     Biased,raw - Biased,z  0.2657691 0.01824644 Inf  14.56553
## 4  AUROC Unbiased,raw - Unbiased,z -0.2006233 0.01836834 Inf -10.92223
##         p.value
## 1 0.000000e+00
```

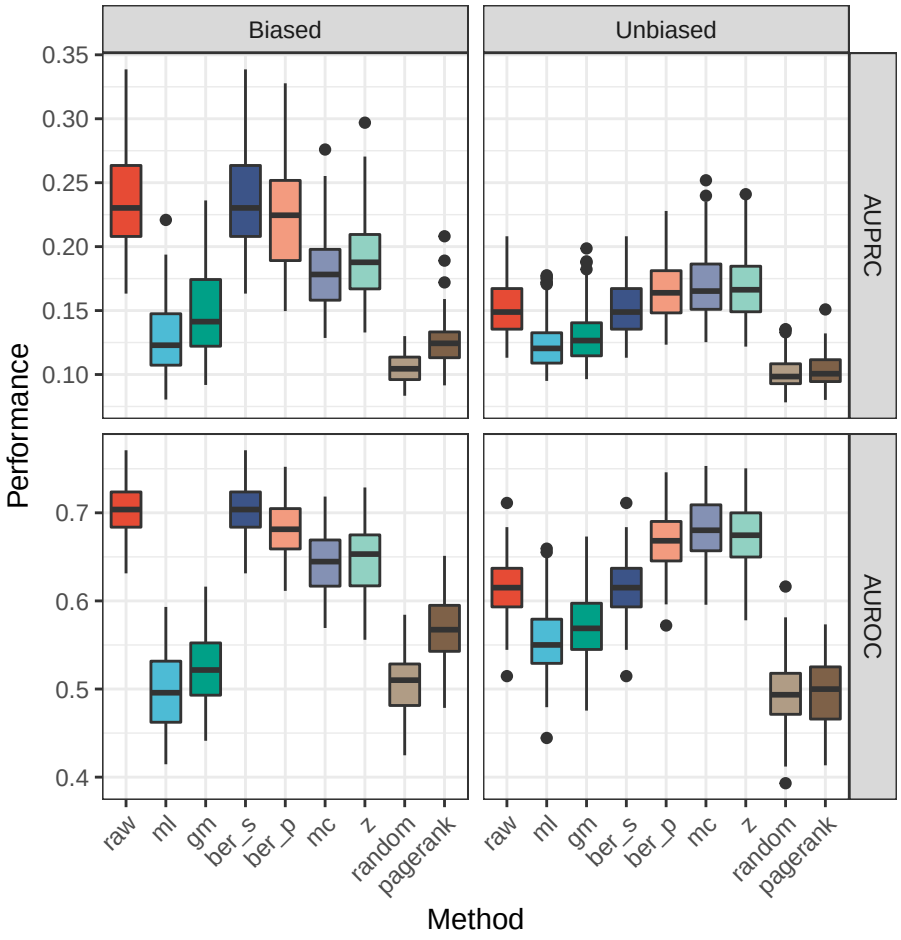


Figure 48: AUROC and AUPRC of diffusion scores for biased and unbiased signals.

Table 16: Quasibinomial model for AUROC and AUPRC. Estimates with 0.95 confidence intervals.

methodml	-0.848^{***} (-0.884, -0.811)
methodgm	-0.719^{***} (-0.755, -0.683)
methodber_p	-0.086^{***} (-0.122, -0.051)
methodmc	-0.304^{***} (-0.339, -0.268)
methodz	-0.266^{***} (-0.302, -0.230)
methodrandom	-0.895^{***} (-0.932, -0.859)
methodpagerank	-0.653^{***} (-0.690, -0.617)
metricAUROC	2.131^{***} (2.118, 2.145)
methodraw:biasedUnbiased	-0.455^{***} (-0.491, -0.419)
methodml:biasedUnbiased	0.155^{***} (0.118, 0.192)
methodgm:biasedUnbiased	0.092^{***} (0.056, 0.129)
methodber_p:biasedUnbiased	-0.186^{***} (-0.221, -0.150)
methodmc:biasedUnbiased	0.075^{***} (0.039, 0.111)
methodz:biasedUnbiased	0.011 (-0.025, 0.047)
methodrandom:biasedUnbiased	-0.031 (-0.069, 0.006)
methodpagerank:biasedUnbiased	-0.271^{***} (-0.308, -0.234)
Constant	-1.229^{***} (-1.255, -1.202)
Observations	3,200
Note:	*p<0.05; **p<0.01; ***p<0.001

```
## 2 2.137179e-13
## 3 0.000000e+00
## 4 2.137179e-13
```

Another interesting remark from figure 48: `pagerank` had predictive power only in the biased setup. The predictive power of an input-naive centrality measure like `pagerank` can be a reason to suspect that the signal is biased towards high-degree nodes.

```
##   metric                           contrast     estimate         SE  df
## 1  AUPRC     Biased,random - Biased,pagerank -0.241916366 0.01888432 Inf
## 2  AUPRC Unbiased,random - Unbiased,pagerank -0.002012079 0.01915065 Inf
## 3  AUROC     Biased,random - Biased,pagerank -0.241916366 0.01888432 Inf
## 4  AUROC Unbiased,random - Unbiased,pagerank -0.002012079 0.01915065 Inf
##       z.ratio p.value
## 1 -12.8104369       0
## 2  -0.1050659       1
## 3 -12.8104369       0
## 4  -0.1050659       1
```

Correlation between method performances

Finally, we examined the similarities between diffusion scores at the performance level. Figure 49 shows the Spearman correlation between the performance metrics of the diffusion scores. Small differences were observed

Bias_signal	Method	AUPRC	AUROC
Biased	raw	(0.222, 0.231)	(0.706, 0.717)
Biased	ml	(0.109, 0.114)	(0.507, 0.520)
Biased	gm	(0.122, 0.128)	(0.539, 0.552)
Biased	ber_p	(0.207, 0.216)	(0.688, 0.699)
Biased	mc	(0.174, 0.182)	(0.639, 0.651)
Biased	z	(0.179, 0.187)	(0.648, 0.660)
Biased	random	(0.104, 0.110)	(0.495, 0.508)
Biased	pagerank	(0.129, 0.135)	(0.555, 0.568)
Unbiased	raw	(0.153, 0.160)	(0.604, 0.616)
Unbiased	ml	(0.125, 0.131)	(0.546, 0.559)
Unbiased	gm	(0.132, 0.138)	(0.562, 0.575)
Unbiased	ber_p	(0.178, 0.186)	(0.647, 0.658)
Unbiased	mc	(0.185, 0.193)	(0.657, 0.668)
Unbiased	z	(0.181, 0.189)	(0.651, 0.662)
Unbiased	random	(0.101, 0.107)	(0.487, 0.501)
Unbiased	pagerank	(0.101, 0.107)	(0.488, 0.501)

Table 17: Confidence intervals (0.95) on predicted AUROC and AUPRC.

between AUROC and AUPRC: `ml` and `gm` correlated with `raw` with AUPRC but not with AUROC. In general, all the proper diffusion scores tended to correlate, even more than we observed in figure 45.

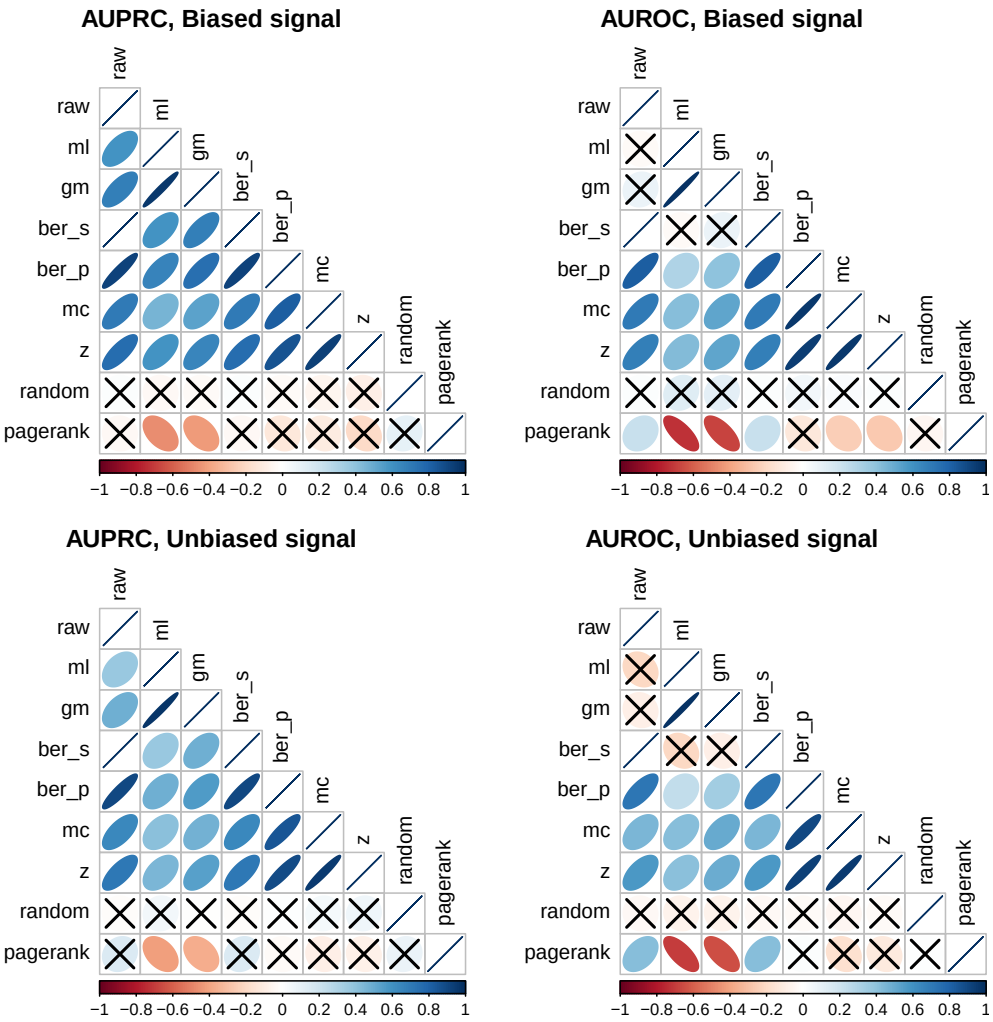


Figure 49: Spearman correlation between performance metrics of diffusion scores and baselines. Correlations not significant at $p < 0.05$ after multiple testing were crossed out.

A.2.4 Conclusions

The main findings from this proof of concept:

- Our definitions of biased and unbiased signals seemed consistent: unbiased signals were uncorrelated with the reference expected value, whereas biased ones showed a positive correlation.
- Changing the labels for diffusion (e.g. `ml` versus `raw`) and normalising the scores had a noticeable impact on the prioritisations and their performance.
- `mc` and `z` had a similar behaviour. This was expected, as they are the parametric and non-parametric alternatives for normalising.
- The adequateness of normalising lied on the distribution of positives across the reference expected value $\mathrm{b}_{\mu}^{\mathcal{K}}$. Biased signals favoured `raw` by definition, whereas `mc` and `z` were preferrable on unbiased signals. Even within a hypothetical case study without overall performance differences, `raw` and `z`/`mc` would be expected to behave differently.
- Class imbalance backfired in `ml` and `gm`, as the properties of the negatives overshadowed those of the positives. A hypothetical case where positives outnumbered negatives might cause a similar effect `raw` as well.
- The complementarity of `raw` and `mc` leaves `ber_p` as a good compromise between both.

A.2.5 Metadata

```
## [1] "Sun Feb 23 11:07:09 2020"
```

```
## R version 3.5.3 (2019-03-11)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.6 LTS
##
## Matrix products: default
## BLAS: /usr/lib/atlas-base/atlas/libblas.so.3.0
## LAPACK: /usr/lib/atlas-base/atlas/liblapack.so.3.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] bindrcpp_0.2.2    xtable_1.8-3      data.table_1.11.8
##  [4] extrafont_0.17    gtable_0.2.0      GGally_1.4.0
##  [7] ggsci_2.9         ggplot2_3.1.0     tidyr_0.8.2
## [10] dplyr_0.7.8       plyr_1.8.4        reshape2_1.4.3
## [13] magrittr_1.5      diffuStats_1.2.0  igraphdata_1.0.1
## [16] igraph_1.2.2      rmarkdown_1.10
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.0                mvtnorm_1.0-8
##  [3] lattice_0.20-38           zoo_1.8-4
##  [5] assertthat_0.2.0          rprojroot_1.3-2
##  [7] digest_0.6.18             packrat_0.5.0
##  [9] R6_2.3.0                  backports_1.1.2
## [11] evaluate_0.12             pillar_1.3.0
## [13] rlang_0.3.0.1             lazyeval_0.2.1
## [15] multcomp_1.4-8            extrafontdb_1.0
## [17] Matrix_1.2-15             labeling_0.3
## [19] splines_3.5.3             stringr_1.3.1
## [21] munsell_0.5.0             compiler_3.5.3
## [23] pkgconfig_2.0.2           mgcv_1.8-27
## [25] htmltools_0.3.6           tidyselect_0.2.5
## [27] tibble_1.4.2              expm_0.999-3
## [29] codetools_0.2-16          reshape_0.8.8
## [31] crayon_1.3.4              withr_2.1.2
## [33] MASS_7.3-51.1             nlme_3.1-137
```

```
## [35] Rttf2pt1_1.3.7           scales_1.0.0
## [37] RcppParallel_4.4.1       estimability_1.3
## [39] stringi_1.2.4            RcppArmadillo_0.9.200.4.0
## [41] sandwich_2.5-0           TH.data_1.0-9
## [43] stargazer_5.2.2          RColorBrewer_1.1-2
## [45] tools_3.5.3              glue_1.3.0
## [47] purrr_0.2.5              emmeans_1.3.0
## [49] survival_2.43-3          yaml_2.2.0
## [51] colorspace_1.3-2         corrplot_0.84
## [53] knitr_1.20               bindr_0.1.1
## [55] precrec_0.9.1
```

A.3 SUPPLEMENT 3: DLBCL DATASET

A.3.1 Introduction

This additional file contains details on the DLBCL dataset, the human proteome and the synthetic signals generated on it. This document can be re-built anytime by knitting its corresponding `.Rmd` file.

A.3.2 The network

We used the HPRD network (Mishra et al., 2006) as used in the DLBCL package (M. Dittrich and Beisser, 2010), which provides a case study for the BioNet R package (M. T. Dittrich et al., 2008). Below is a summary of the network, obtained by taking the largest connected component from the original network `interactome`:

```
## IGRAPH e764cea UNW- 8989 34325 --
## + attr: kegg_mapped (g/x), info (g/c), name (v/c), geneID (v/c),
## | geneSymbol (v/c), obs_lym (v/l), obs_all (v/l), weight (e/n)
```

The network contained 8989 nodes and 34325 edges and was connected by construction. The edges were unweighted, as they had a constant, unitary weight:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

A.3.3 Descriptive statistics

Simulated signals

Signals to benchmark the diffusion scores were obtained by sub-sampling the Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways (Kanehisa et al., 2017), using the release:

```
## [1] "T01001          Homo sapiens (human) KEGG Genes Database"
## [2] "hsa             Release 83.0+/09-09, Sep 17"
## [3] "                Kanehisa Laboratories"
## [4] "                39,524 entries"
```

Pathways were used like gene sets, without taking further network data from the KEGG database. After mapping the pathways to the network, their size followed the distribution in figure 50. Only pathways with a minimum of $N_{min} = 30$ genes were considered.

Likewise, figure 51 depicts the amount of pathways in which each gene participates. Although some genes are ubiquitous, most of them belong to less than 10 pathways.

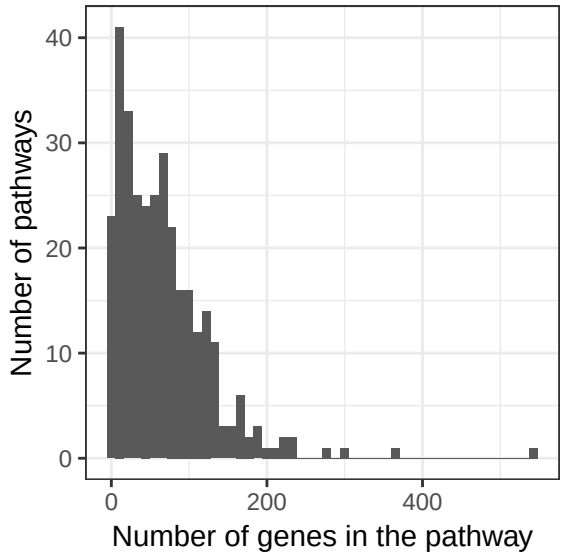


Figure 50: Histogram with number of pathways involving each gene

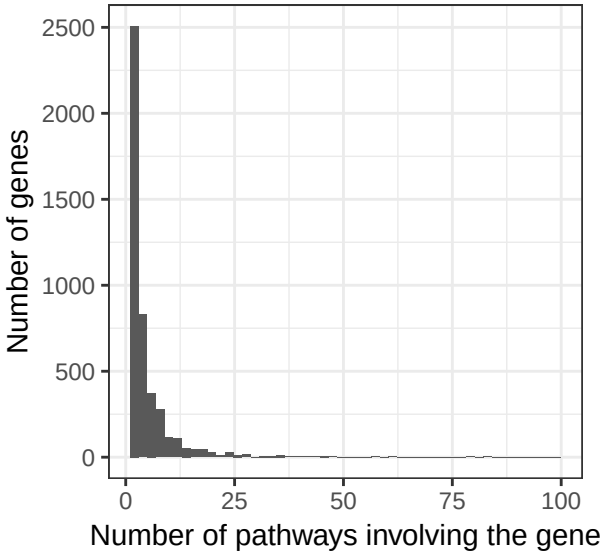


Figure 51: Histogram with number of genes in each pathway

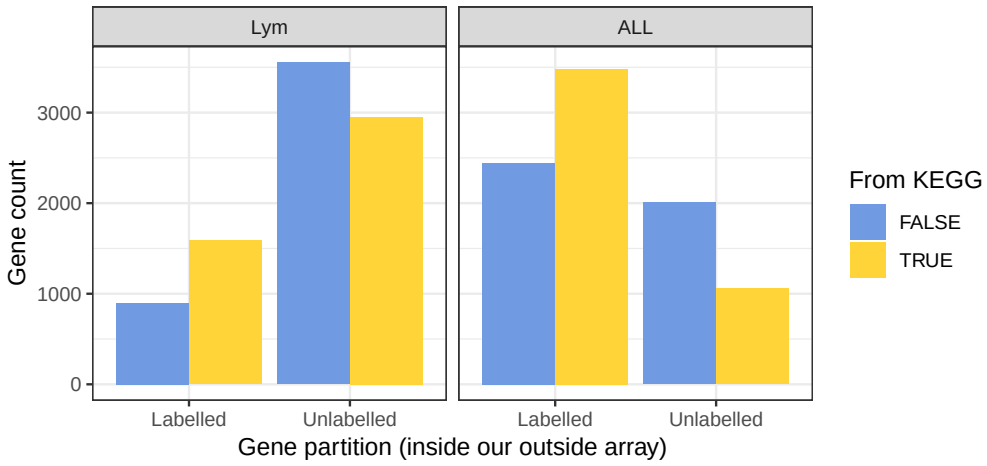


Figure 52: Number of genes inside and outside pathways, stratified by observability and array

Sub-sampling

The sub-sampling was governed by three key parameters: the number of affected pathways $k \in \{1, 3, 5, 10\}$, the proportion of differentially expressed genes $r \in \{0.3, 0.5, 0.7\}$ and the maximum p-value for differential expression, $p_{max} \in \{0.01, 0.001, 10^{-4}, 10^{-5}\}$. The extreme values $k = 10$ and $p_{max} = 10^{-5}$ led to redundant results and were left out of the main analyses.

As described in the main body, in each run k pathways were uniformly sampled and their genes were tagged as positives. A proportion of r positives was uniformly sampled to show differential expression, with their p-values uniformly sampled in $[0, p_{max}]$. The remaining proportion of $1 - r$ genes were not differentially expressed, imposed by sampling their p-values uniformly in $[0, 1]$. For each combination of parameters, a total of $N = 50$ repetitions were generated. Regardless of which nodes were considered as *unlabelled* or *labelled*, the p-values were generated for all the nodes in the network.

Array-based backgrounds

In order to evaluate the effect of the statistical background, genes from the network were partitioned by **observability** into *labelled* and *unlabelled*. *Labelled* nodes were defined as those belonging to an array, whereas *unlabelled* nodes were those outside it. Two arrays were used: the *ALL* array (Chiaretti et al., 2004), obtained from the ALL R package (Li, 2009), and the *Lym* array (Rosenwald et al., 2002) from the DLBCL package (M. Dittrich and Beisser, 2010). Gene identifiers in *ALL* were mapped to the network through `BioNet::mapByVar()` from the BioNet package (M. T. Dittrich et al., 2008), whereas those of *Lym* were already mapped in the data package. Each array had its own *labelled* and *unlabelled* genes: figure 52 represents the amount of genes within each background and their belonging to the KEGG pathways.

The exact numbers are found in the following snippet, which also includes the proportion of KEGG pathways that could be observed in both arrays. The size of *ALL* exceeded that of *Lym* by more than two-fold and was therefore expected to outperform it.

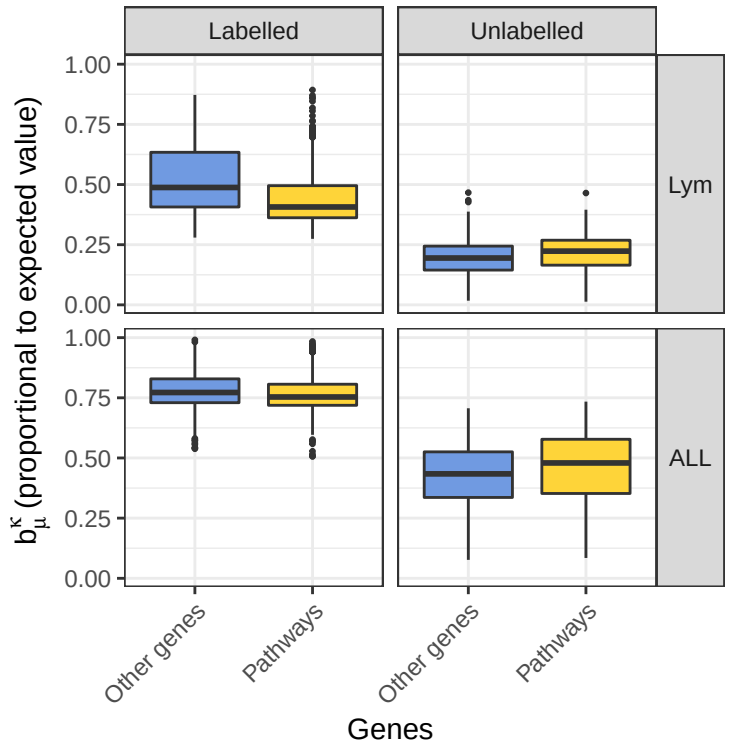


Figure 53: Expected values inside and outside pathways, stratified by observability and array

```
##   array n_labelled n_unlabelled n_labelled_kegg n_unlabelled_kegg
## 1   Lym       2482         6507            1586              2953
## 2   ALL       5921         3068            3479              1060
##   prop_labelled_kegg
## 1          0.3494162
## 2          0.7664684
```

Finally, we show the overlap between the KEGG pathways and the *obs* and *Lym* arrays. The table below counts the number of genes lying in the intersections. Most of the genes of the smaller array *Lym* are part of *ALL* as well.

```
##      kegg  ALL  Lym
## kegg 4539 3479 1586
## ALL  3479 5921 2006
## Lym  1586 2006 2482
```

Theoretical bias in diffusion scores

As exposed in the main body, the diffusion scores are expected to be biased in terms of their expected value for each node under input permutations. According to the definitions therein, the expected value of a node i is proportional to its reference expected value $b_\mu^{\mathcal{K}}(i)$. Figure 53 depicts this magnitude, stratified by pathway membership, observability and array.

The following claims were statistically significant in both arrays (Wilcoxon rank-sum test):

1. In the *labelled* genes, pathway genes had a **lower** reference expected value than non-pathway genes.
2. In the *unlabelled* genes, pathway genes had a **higher** reference expected value than non-pathway genes.
3. *Labelled* genes had a **higher** reference expected value than *unlabelled* genes.

Claims 1 and 2:

```
##    obs_label array difference_medians pvalue_wilcox          fdr
## 1   Labelled   Lym        -0.08107461  2.355835e-44 4.711670e-44
## 2   Labelled   ALL        -0.01861420  6.771854e-17 6.771854e-17
## 3 Unlabelled   Lym         0.02852555  4.061263e-39 8.122525e-39
## 4 Unlabelled   ALL         0.04525165  2.472910e-18 2.472910e-18
```

Claim 3:

```
##   array difference_median_bias pvalue_wilcox fdr
## 1   Lym              0.2262074             0   0
## 2   ALL              0.3119931             0   0
```

As every pathway gene was a potential positive, in general terms `raw` should benefit from the bias in (2) and `z` from that in (1). As for *overall* performance (3), `z` equalises *labelled* and *unlabelled* nodes, mixing high and low-confidence predictions. Reliable predictions from the *labelled* part should be masked by those in the *unlabelled* part and the *overall* performance is expected to decrease.

An important difference exists between indirect bias measurements and the direct quantification of $b_{\mu}^{\mathcal{K}}$. The claims above would be different if PageRank was used as a measure of centrality, under the hypothesis that the bias favours central genes. Figure 54 depicts the PageRank scores (`damping = 0.85`) of all the genes, organised into: both arrays, inside and outside KEGG pathways, labelled and unlabelled nodes. Two PageRank flavours are included: (i) uniform prior and (ii) personalised prior, starting at the labelled genes of each array. In both alternatives, this point of view suggests that `raw` should outperform `z` in the three scenarios, implying that claim (1) would reverse and (2) and (3) would hold.

Genes inside pathways have significantly higher PageRank scores than those outside, in each one of the eight combinations in figure 54:

```
##    obs_label array                 Prior difference_medians pvalue_wilcox
## 1   Labelled   Lym Personalized PageRank       6.906910e-05  2.969662e-76
## 2   Labelled   Lym      Uniform PageRank       5.949779e-05  3.422633e-70
## 3   Labelled   ALL Personalized PageRank       3.437695e-05  3.309309e-90
## 4   Labelled   ALL      Uniform PageRank       3.122078e-05  4.584032e-82
## 5 Unlabelled   Lym Personalized PageRank       1.903473e-05  1.399483e-56
## 6 Unlabelled   Lym      Uniform PageRank       2.034443e-05  8.219762e-58
## 7 Unlabelled   ALL Personalized PageRank       1.174467e-05  8.556411e-19
## 8 Unlabelled   ALL      Uniform PageRank       1.271804e-05  2.362005e-18
##            fdr
```

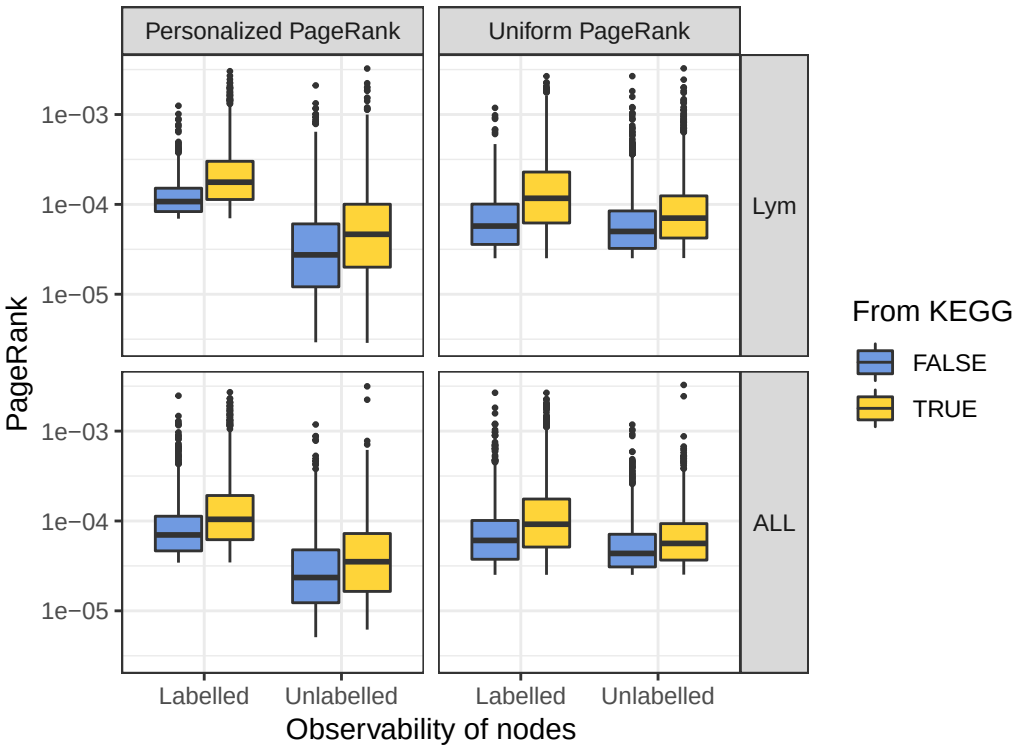


Figure 54: PageRank centralities inside and outside pathways, stratified by observability and array

```
## 1 5.939324e-76
## 2 3.422633e-70
## 3 6.618618e-90
## 4 4.584032e-82
## 5 1.399483e-56
## 6 1.643952e-57
## 7 1.711282e-18
## 8 2.362005e-18
```

Diffusion inputs

In order to binarise the labels for the diffusion, the false discovery rate, or FDR (Benjamini and Hochberg, 1995), of the *labelled* nodes was computed. *Labelled* nodes were defined as positive if their FDR was below 0.1 and negative otherwise. Nodes from the *unlabelled* pool were naturally deemed unlabelled for the diffusion process.

Note that the input could contain false positives due to false positives in hypothesis testing. Likewise, false negatives were expected by the definition of the signal, because only a portion of the genes of the affected pathways will show changes. Occasionally, especially in weak signals (low r, k and high p_{max}), none of the *labelled* genes would be significant at the specified FDR. Along with other degenerate cases (i.e. no positives in the *unlabelled* nodes), these instances were discarded.

A summary of the metrics table illustrates how the number of instances increased with increasing r, k and decreasing p_{max}:

```
##  array              strat           method          auroc
##  Lym:61440   Labelled  :40080   raw    :12024   Min.   :0.01272
##  ALL:58800   Unlabelled:40080   ml     :12024   1st Qu.:0.60475
##              Overall   :40080   gm     :12024   Median :0.75456
##                                 ber_s  :12024   Mean   :0.72590
##                                 ber_p  :12024   3rd Qu.:0.86399
##                                 mc     :12024   Max.   :1.00000
##                                 (Other):48096
##      auprc              Column          k           r
##  Min.   :0.0001651   Length:120240      1 :25920   0.3:38160
##  1st Qu.:0.0676635   Class :character   3 :29610   0.5:40650
##  Median :0.1925403   Mode  :character   5 :30900   0.7:41430
##  Mean   :0.2813699                      10:33810
##  3rd Qu.:0.4753768
##  Max.   :1.0000000
##
##     pmax
##  1e-02:15180
##  1e-03:33120
##  1e-04:35940
##  1e-05:36000
##
##
##
```

For methods requiring permutations, the number of permutations was set to 1000 for computational reasons. In all cases, the regularised (unnormalised) Laplacian kernel was used.

A.3.4 Models

Model definition

The performance of the diffusion scores in the two arrays under the three signal parameters was best described through explanatory models. Positives in validation were defined as the union of the k pathways that generated each signal. AUROC and AUPRC were computed in three ways: in all the nodes (*overall*), only in the *labelled* part and only in the *unlabelled* part.

Three reference methods were kept. First, `original` ranked the nodes according to their p-value before computing the FDR. In the *labelled* genes, this quantifies the added value of the diffusion process beyond the original signal, i.e. does the diffusion improve the findings obtained by prioritising the genes by their p-value? Regarding the *unlabelled* genes, `original` serves as a reference, as diffusion ignored such p-values by design was not expected to outperform them, especially if r was high or p_{max} was small. Diffusion performance was compared to a hypothetical case in which we knew the original signal – although in general an imperfect one, with false positives and false negatives.

The remaining baselines were `pagerank`, a centrality measure that ignored every input and suggested central genes as top candidates, and `random`, a uniformly random re-ordering of the genes.

The metrics AUROC and AUPRC were modelled through dispersion-adjusted quasibinomial logit models, see `?stats::quasibinomial` in an R console:

$$\text{metric} \sim \text{method} + \text{method:strat} + \text{array} + k + r + p_{max}$$

All the variables were treated as categorical. The interaction term `method:strat` ensured that methods were allowed to have differential performance in the *labelled*, *unlabelled* and *overall* node stratifications. The values $p_{max} = 10^{-5}$ and $k = 10$ were left out due to their respective similarity to $p_{max} = 10^{-4}$ and $k = 5$. Each model is described in its own section.

AUROC

In this instance, AUROC did not stand out as the ideal metric – details on its model can be found in table 18.

One reason is that, although significant differences existed between methods among *labelled*, *unlabelled* and all nodes, such differences always happened in a narrow range.

More importantly, the performances of diffusion scores (except the ones diffusing -1 on the negatives, `ml` and `gm`) were comparable to the `original` p-values in the *unlabelled* genes. The fact that diffusion-based method had no prior data on the *unlabelled* genes should hinder their performance within them, compared to (i) the *labelled* fold, and especially (ii) to the original, unobserved p-values, more notably if r was large. This was not the case, as depicted in figure 55, with predictions by array and partition.

AUPRC

Contrary to AUROC, AUPRC (see table 19) was more informative for this task.

The quasibinomial model confirmed expected phenomena regarding performance, such as the positive influence of increasing k and r and decreasing p_{max} and the superiority of the *ALL* array. Contrary to AUROC, performance of diffusion scores suffered a pronounced drop in the *unlabelled* genes. Therefore, there was a notable gap in terms of early retrieval between both, something not that apparent from AUROC alone.

Figure 56 shows the expected behaviour of the diffusion scores (actual values in Table 20), in terms of the aforementioned reference expected value $b_\mu^{\mathcal{K}}$. As anticipated, `raw` outperformed `z` in the *unlabelled* nodes and *overall*, whereas `z` outperformed `raw` in the *labelled* nodes.

Table 18: Quasilogistic model for AUROC

methodml	−0.393*** (−0.432, −0.355)
methodgm	−0.274*** (−0.313, −0.235)
methodber_s	−0.135*** (−0.175, −0.095)
methodber_p	−0.044* (−0.085, −0.004)
methodmc	−0.199*** (−0.238, −0.159)
methodz	−0.162*** (−0.201, −0.122)
methodoriginal	−0.750*** (−0.787, −0.714)
methodpagerank	−1.059*** (−1.095, −1.023)
methodrandom	−1.953*** (−1.988, −1.918)
k3	0.044*** (0.034, 0.055)
k5	0.019*** (0.009, 0.030)
r0.5	0.237*** (0.227, 0.247)
r0.7	0.426*** (0.415, 0.436)
pmax1e-03	0.717*** (0.705, 0.729)
pmax1e-04	0.797*** (0.785, 0.809)
arrayALL	0.145*** (0.137, 0.153)
methodraw:stratUnlabelled	−0.786*** (−0.822, −0.749)
methodml:stratUnlabelled	−1.887*** (−1.919, −1.855)
methodgm:stratUnlabelled	−1.818*** (−1.851, −1.785)
methodber_s:stratUnlabelled	−0.650*** (−0.686, −0.615)
methodber_p:stratUnlabelled	−0.700*** (−0.736, −0.663)
methodmc:stratUnlabelled	−0.657*** (−0.692, −0.622)
methodz:stratUnlabelled	−0.706*** (−0.741, −0.671)
methodoriginal:stratUnlabelled	−0.051** (−0.083, −0.019)
methodpagerank:stratUnlabelled	−0.365*** (−0.395, −0.336)
methodrandom:stratUnlabelled	0.004 (−0.024, 0.031)
methodraw:stratOverall	−0.309*** (−0.348, −0.270)
methodml:stratOverall	−1.352*** (−1.384, −1.320)
methodgm:stratOverall	−1.270*** (−1.303, −1.237)
methodber_s:stratOverall	−0.234*** (−0.271, −0.196)
methodber_p:stratOverall	−0.274*** (−0.313, −0.236)
methodmc:stratOverall	−0.289*** (−0.325, −0.252)
methodz:stratOverall	−0.349*** (−0.386, −0.313)
methodoriginal:stratOverall	−0.017 (−0.050, 0.015)
methodpagerank:stratOverall	−0.041** (−0.071, −0.011)
methodrandom:stratOverall	0.001 (−0.027, 0.028)
Constant	0.974*** (0.942, 1.005)
Observations	59,430
Note:	*p<0.05; **p<0.01; ***p<0.001

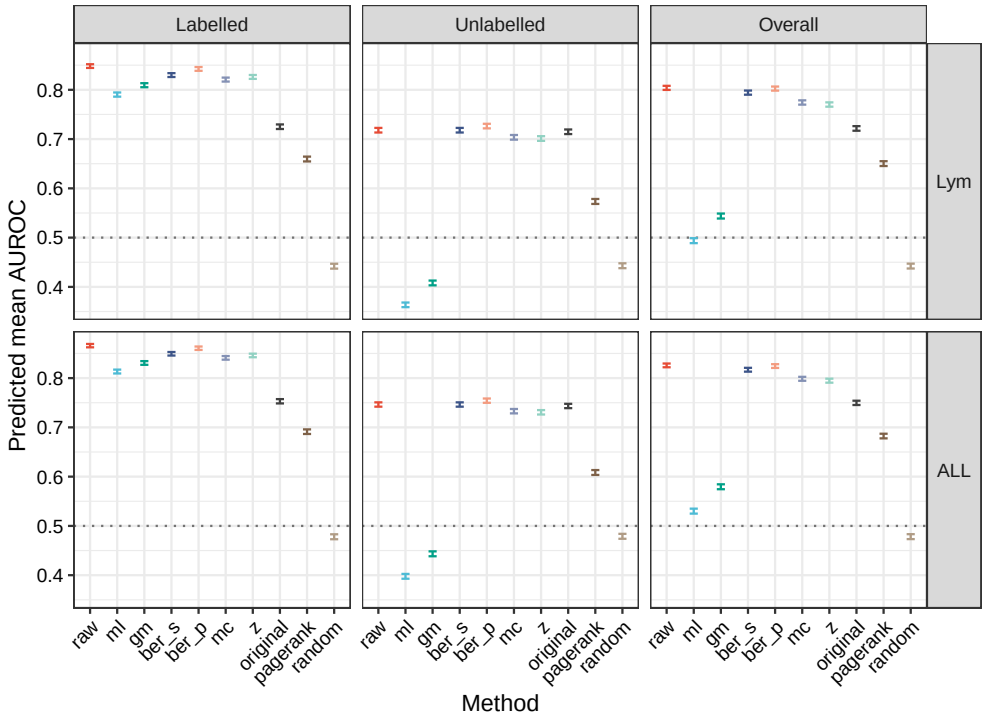


Figure 55: Predictions using the AUROC model (0.95 confidence intervals). Predictions were averaged over the other categorical covariates.

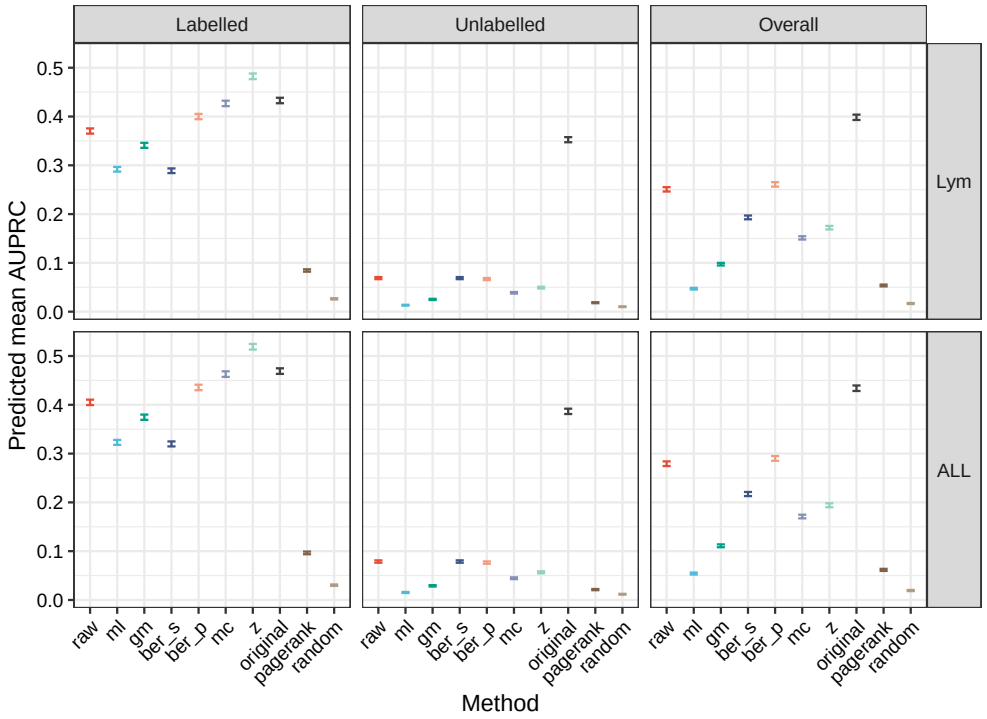


Figure 56: Predictions using the AUPRC model (0.95 confidence intervals). Predictions were averaged over the other categorical covariates.

Table 19: Quasilogistic model for AUPRC

methodml	−0.355*** (−0.386, −0.324)
methodgm	−0.128*** (−0.159, −0.097)
methodber_s	−0.370*** (−0.401, −0.339)
methodber_p	0.125*** (0.094, 0.156)
methodmc	0.236*** (0.205, 0.267)
methodz	0.461*** (0.429, 0.492)
methodoriginal	0.261*** (0.230, 0.292)
methodpagerank	−1.853*** (−1.890, −1.815)
methodrandom	−3.082*** (−3.136, −3.028)
k3	0.447*** (0.434, 0.460)
k5	0.618*** (0.605, 0.631)
r0.5	0.512*** (0.499, 0.525)
r0.7	0.912*** (0.899, 0.924)
pmax1e-03	1.548*** (1.528, 1.568)
pmax1e-04	1.672*** (1.652, 1.692)
arrayALL	0.147*** (0.136, 0.157)
methodraw:stratUnlabelled	−2.075*** (−2.114, −2.035)
methodml:stratUnlabelled	−3.424*** (−3.496, −3.353)
methodgm:stratUnlabelled	−3.001*** (−3.056, −2.946)
methodber_s:stratUnlabelled	−1.705*** (−1.745, −1.665)
methodber_p:stratUnlabelled	−2.230*** (−2.270, −2.190)
methodmc:stratUnlabelled	−2.918*** (−2.965, −2.871)
methodz:stratUnlabelled	−2.889*** (−2.933, −2.845)
methodoriginal:stratUnlabelled	−0.338*** (−0.369, −0.307)
methodpagerank:stratUnlabelled	−1.594*** (−1.660, −1.529)
methodrandom:stratUnlabelled	−0.969*** (−1.061, −0.877)
methodraw:stratOverall	−0.564*** (−0.595, −0.532)
methodml:stratOverall	−2.121*** (−2.165, −2.077)
methodgm:stratOverall	−1.568*** (−1.604, −1.531)
methodber_s:stratOverall	−0.528*** (−0.561, −0.496)
methodber_p:stratOverall	−0.636*** (−0.667, −0.604)
methodmc:stratOverall	−1.430*** (−1.464, −1.397)
methodz:stratOverall	−1.500*** (−1.533, −1.467)
methodoriginal:stratOverall	−0.142*** (−0.173, −0.111)
methodpagerank:stratOverall	−0.486*** (−0.533, −0.438)
methodrandom:stratOverall	−0.462*** (−0.541, −0.384)
Constant	−2.434*** (−2.465, −2.403)
Observations	59,430
Note:	*p<0.05; **p<0.01; ***p<0.001

array	method	Labelled	Unlabelled	Overall
Lym	raw	(0.365, 0.376)	(0.067, 0.071)	(0.246, 0.255)
Lym	ml	(0.287, 0.297)	(0.012, 0.014)	(0.045, 0.049)
Lym	gm	(0.336, 0.346)	(0.024, 0.026)	(0.095, 0.100)
Lym	ber_s	(0.284, 0.294)	(0.067, 0.071)	(0.189, 0.197)
Lym	ber_p	(0.394, 0.405)	(0.065, 0.069)	(0.256, 0.265)
Lym	mc	(0.421, 0.432)	(0.037, 0.040)	(0.148, 0.155)
Lym	z	(0.477, 0.488)	(0.048, 0.051)	(0.169, 0.176)
Lym	original	(0.427, 0.438)	(0.347, 0.358)	(0.393, 0.404)
Lym	pagerank	(0.082, 0.087)	(0.017, 0.019)	(0.052, 0.056)
Lym	random	(0.025, 0.028)	(0.009, 0.011)	(0.016, 0.018)
ALL	raw	(0.399, 0.411)	(0.076, 0.081)	(0.274, 0.284)
ALL	ml	(0.318, 0.328)	(0.014, 0.016)	(0.052, 0.056)
ALL	gm	(0.369, 0.380)	(0.028, 0.030)	(0.108, 0.114)
ALL	ber_s	(0.315, 0.325)	(0.076, 0.081)	(0.213, 0.221)
ALL	ber_p	(0.430, 0.441)	(0.074, 0.079)	(0.285, 0.295)
ALL	mc	(0.457, 0.469)	(0.043, 0.046)	(0.167, 0.175)
ALL	z	(0.513, 0.525)	(0.055, 0.059)	(0.190, 0.198)
ALL	original	(0.463, 0.475)	(0.381, 0.392)	(0.428, 0.440)
ALL	pagerank	(0.094, 0.099)	(0.020, 0.022)	(0.060, 0.064)
ALL	random	(0.029, 0.032)	(0.011, 0.013)	(0.018, 0.020)

Table 20: Confidence intervals (0.95) on predicted AUPRC, averaged over covariates.

Below are the results of the statistical test between raw and z that back up the claims in this section.

```
##                                contrast odds.ratio p.value
## 1     raw,Labelled,Lym / z,Labelled,Lym  0.6308394       0
## 2 raw,Unlabelled,Lym / z,Unlabelled,Lym  1.4240117       0
## 3       raw,Overall,Lym / z,Overall,Lym  1.6090716       0
## 4     raw,Labelled,ALL / z,Labelled,ALL  0.6308394       0
## 5 raw,Unlabelled,ALL / z,Unlabelled,ALL  1.4240117       0
## 6       raw,Overall,ALL / z,Overall,ALL  1.6090716       0
```

Other remarks

- Using an indirect measure of bias might be misleading. Here, by using PageRank as a centrality measure and assuming that raw scores will favour highly connected nodes, we would expect that raw outperforms z in the *labelled* nodes. However, this is inded the opposite to what $b_{\mu}^{\mathcal{K}}$ (a direct quantification of the expected value-related bias) suggests and to what we observe in terms of performance.
- The original baseline was difficult to improve upon, even in the *labelled* genes, in terms of AUPRC. This was not the case for AUROC, implying that although diffusion had a positive and noticeable impact in the overall ranking, early retrieval was a challenging task.
- ber_p had the best *overall* performance, suggesting that a consensus between normalised and unnormalised scores can be beneficial.
- ml and gm suffered from this imbalanced datasets, where positives were vastly outnumbered by negatives.
- Within normalised scores, z outperformed mc, possibly due to the presence of ties and the stochastic nature of the latter.

A.3.5 Reproducibility

```
##  [1] "R version 3.5.3 (2019-03-11)"
##  [2] "Platform: x86_64-pc-linux-gnu (64-bit)"
##  [3] "Running under: Ubuntu 16.04.6 LTS"
##  [4] ""
##  [5] "Matrix products: default"
##  [6] "BLAS: /usr/lib/atlas-base/atlas/libblas.so.3.0"
##  [7] "LAPACK: /usr/lib/atlas-base/atlas/liblapack.so.3.0"
##  [8] ""
##  [9] "locale:"
## [10] " [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C              "
## [11] " [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8    "
## [12] " [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8   "
## [13] " [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                 "
## [14] " [9] LC_ADDRESS=C               LC_TELEPHONE=C            "
## [15] "[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C       "
## [16] ""
## [17] "attached base packages:"
## [18] "[1] grid      stats     graphics  grDevices utils     datasets  methods
## [19] "[8] base     "
## [20] ""
## [21] "other attached packages:"
## [22] " [1] stargazer_5.2.2   emmeans_1.3.0     bindrcpp_0.2.2   "
## [23] " [4] xtable_1.8-3      data.table_1.11.8 extrafont_0.17   "
## [24] " [7] gtable_0.2.0      GGally_1.4.0      ggsci_2.9        "
## [25] "[10] ggplot2_3.1.0     tidyr_0.8.2       dplyr_0.7.8      "
## [26] "[13] plyr_1.8.4        reshape2_1.4.3    magrittr_1.5     "
## [27] "[16] diffuStats_1.2.0  igraphdata_1.0.1  igraph_1.2.2     "
## [28] "[19] rmarkdown_1.10   "
## [29] ""
## [30] "loaded via a namespace (and not attached):"
## [31] " [1] Rcpp_1.0.0                mvtnorm_1.0-8            "
## [32] " [3] lattice_0.20-38           zoo_1.8-4                "
## [33] " [5] assertthat_0.2.0          rprojroot_1.3-2          "
## [34] " [7] digest_0.6.18             packrat_0.5.0            "
## [35] " [9] R6_2.3.0                  backports_1.1.2          "
## [36] "[11] evaluate_0.12             pillar_1.3.0             "
## [37] "[13] rlang_0.3.0.1             lazyeval_0.2.1           "
## [38] "[15] multcomp_1.4-8            extrafontdb_1.0          "
## [39] "[17] Matrix_1.2-15             labeling_0.3             "
## [40] "[19] splines_3.5.3             stringr_1.3.1            "
## [41] "[21] munsell_0.5.0             compiler_3.5.3           "
## [42] "[23] pkgconfig_2.0.2           mgcv_1.8-27              "
## [43] "[25] htmltools_0.3.6           tidyselect_0.2.5         "
## [44] "[27] tibble_1.4.2              expm_0.999-3             "
## [45] "[29] codetools_0.2-16          reshape_0.8.8            "
## [46] "[31] crayon_1.3.4              withr_2.1.2              "
## [47] "[33] MASS_7.3-51.1             nlme_3.1-137             "
```

```
## [48] "[35] Rttf2pt1_1.3.7           scales_1.0.0             "
## [49] "[37] RcppParallel_4.4.1       estimability_1.3         "
## [50] "[39] stringi_1.2.4            RcppArmadillo_0.9.200.4.0"
## [51] "[41] sandwich_2.5-0           TH.data_1.0-9            "
## [52] "[43] RColorBrewer_1.1-2       tools_3.5.3              "
## [53] "[45] glue_1.3.0               purrr_0.2.5              "
## [54] "[47] survival_2.43-3          yaml_2.2.0               "
## [55] "[49] colorspace_1.3-2         corrplot_0.84            "
## [56] "[51] knitr_1.20               bindr_0.1.1              "
## [57] "[53] precrec_0.9.1           "
```

A.4 SUPPLEMENT 4: PATHWAY PREDICTION

A.4.1 Introduction

This additional file contains details on the prospective pathway prediction case study. A protein-protein interaction network and biological pathways, both from year 2011, were used to predict new genes in the same pathways from 2018.
This document can be re-built anytime by knitting its corresponding `.Rmd` file.

The network

We used the BioGRID network (Chatr-aryamontri et al., 2017), weighting its interactions according to (Cao et al., 2014). Weights depend on the amount of experiments reporting an interaction and their throughput, favouring low-throughput methodologies.
In addition, in order to avoid circularity between the new pathway genes and the network construction, the network was restricted to interactions from publications in 2010 or older. This posed a realistic prospective scenario, in which the network might not consistently reflect the novel biology behing the newly added genes.

Below is a summary of the network:

```
## IGRAPH 5fd82d3 UNW- 11394 67573 --
## + attr: name (v/c), weight (e/n)
```

The network contained 11394 nodes and 67573 edges and was connected by construction (only the largest connected component was kept). The edges weights are displayed in figure 57, revealing two broad categories: low-confidence ones, with a weight of 0.25, and high-confidence ones, with a weight of 0.8 or higher.

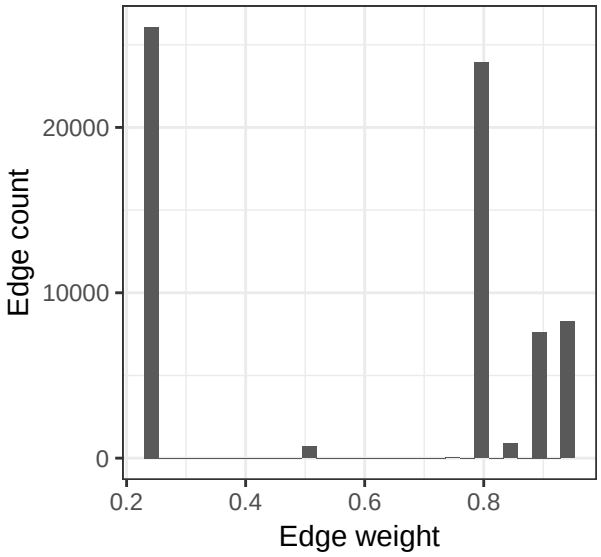


Figure 57: Distribution of the edge weights in the BioGRID network.

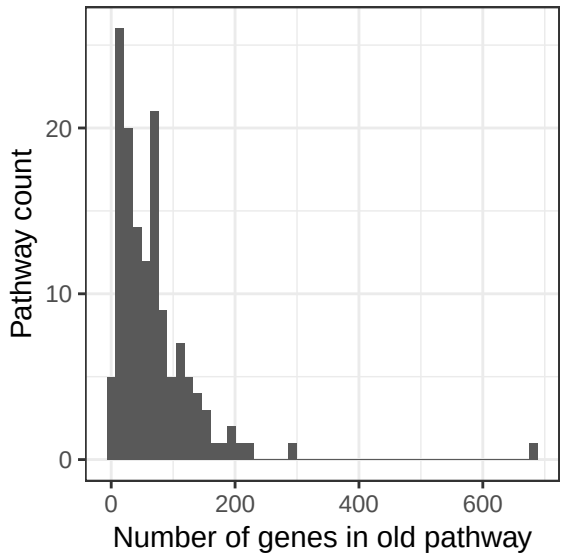


Figure 58: Number of genes per pathway in the older KEGG release

A.4.2 Descriptive statistics

KEGG pathways

The Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways (Kanehisa et al., 2017) was used as input and validation for the diffusion scores. Pathways were treated as gene sets, only relying on the network data from BioGRID.

An older version of the pathways, dating from 2011, was used to predict new pathway genes in 2018. The last public version of KEGG, dated from March 14th 2011, was obtained from the `KEGG.db` package (Carlson, 2016). Likewise, a more recent KEGG release was downloaded in August 18th, 2018 from `https://www.kegg.jp/kegg/rest/keggapi.html`.

A total of 139 KEGG pathways had at least one additional gene in the latest version, after mapping the genes to the BioGRID network. Figure 58 shows that most pathways contained up to 200 genes, while figure 59 depicts how they typically involved less than 20 new genes. Likewise, figure 60 describes how ubiquitous new genes were: most of the new genes belonged to a single pathway.

Theoretical bias in diffusion scores

In this occasion, the inherent bias of the diffusion scores was not related to the expected value of each node under input permutations. Given the present setup, where all the nodes were considered as *labelled*, $b_{\mu}^{\mathcal{K}}$ is constant and thus the `raw` scores must have a constant expected value on all the nodes (see proofs on properties of diffusion scores from Supplement 1). However, differences existed in terms of **variance**. We hypothesised that this led to a variance-related bias, where some nodes would exhibit more stable diffusion scores whereas others could greatly vary under input permutations. Specifically, we hypothesised that `z` would improve the power on low-variance nodes. Variance-related bias was quantified through ther refer-

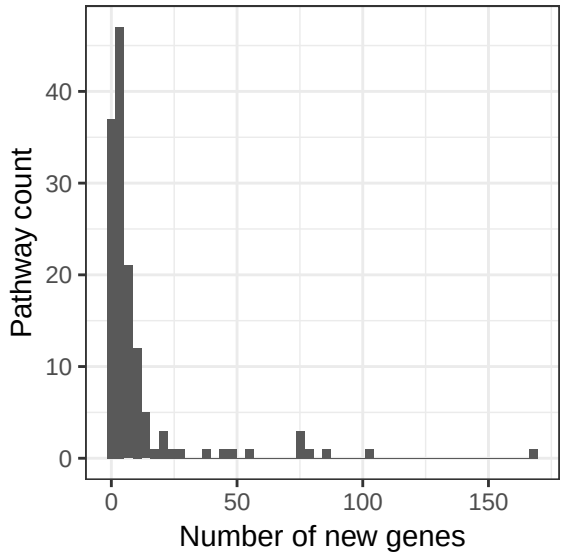


Figure 59: Number of new genes per pathway in the latest KEGG release

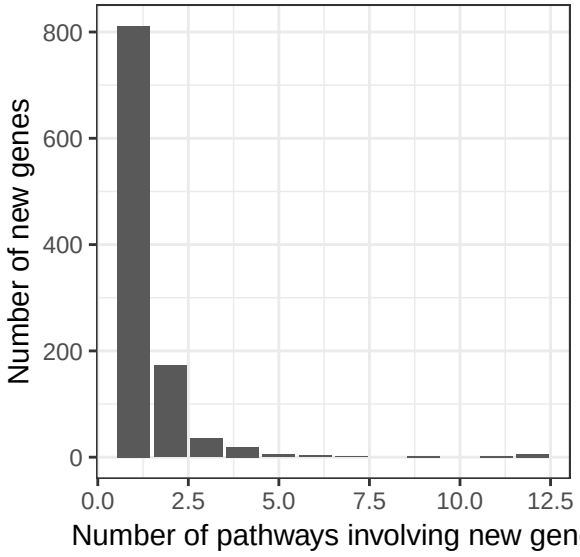


Figure 60: Number of pathways involving each new gene

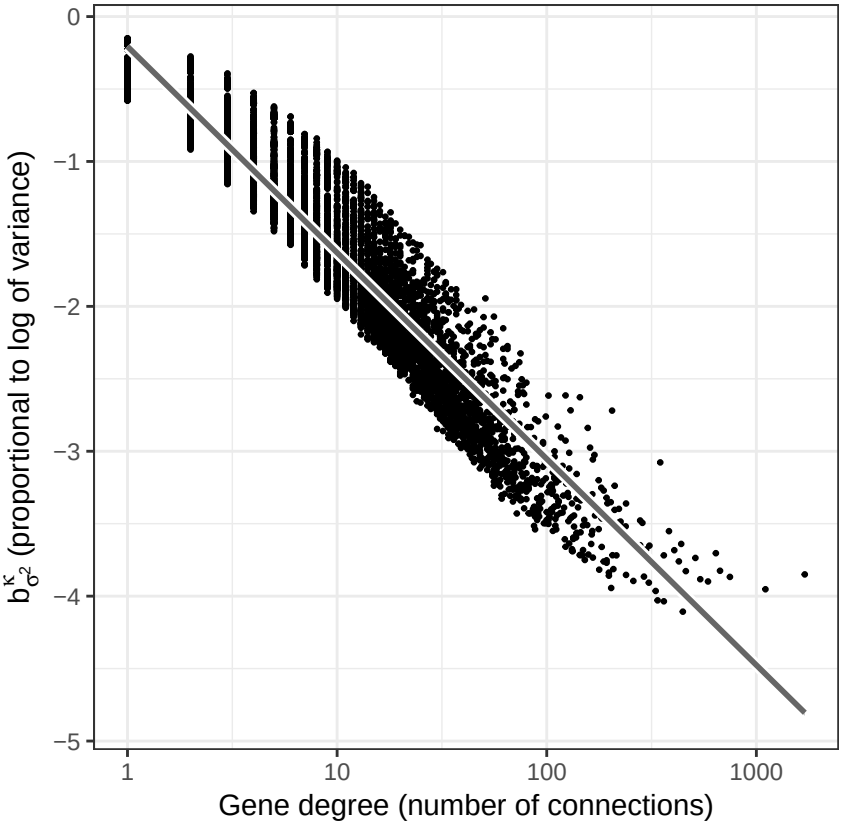


Figure 61: Variance-related bias across all the genes in terms of degree. The gray line shows the best linear fit.

ence variance $b^{\mathcal{K}}_{\sigma^2}$, defined in the main text as proportional to the logarithm of the node variance.

Before framing the genes into pathways, figure 61 suggests that the variance was mainly driven by the node degree. The diffusion scores of highly connected nodes were therefore expected to be less sensitive to perturbations in the input.

Figure 62 depicts the reference variance $b^{\mathcal{K}}_{\sigma^2}$, dividing genes into four categories: *old* for the genes in the old and new pathway, *new* for the genes only in the new pathway, *old_fp* for the genes only in the old pathway and *other* for the rest of genes. Note that a gene can belong to several categories, i.e. *new* for one pathway and *other* for another. Figure 62 suggests that the properties of *old*, *new* and *other* genes are essentially different and linked to their topological properties.

The same magnitude was depicted in terms of pathways, representing the median value of $b^{\mathcal{K}}_{\sigma^2}(i)$ for its *new* genes, see figure 63. The plot suggest that the *new* genes can have two sorts of biases, specifically a standard deviation either (i) lower or (ii) higher than that of the *other* network genes in general.

Differences in $b^{\mathcal{K}}_{\sigma^2}(i)$ between *new* and *other* genes were tested using `wilcox.test` and correcting for False Discovery Rate (FDR) (Benjamini and Hochberg, 1995), see figure 64. Differences at FDR < 0.1 could be proven for pathways some pathways, almost always with more than 5 *new* genes. Significant differences were usually negative (i.e. *other* genes having a greater median, in line with figure 63), but positive differences existed too.

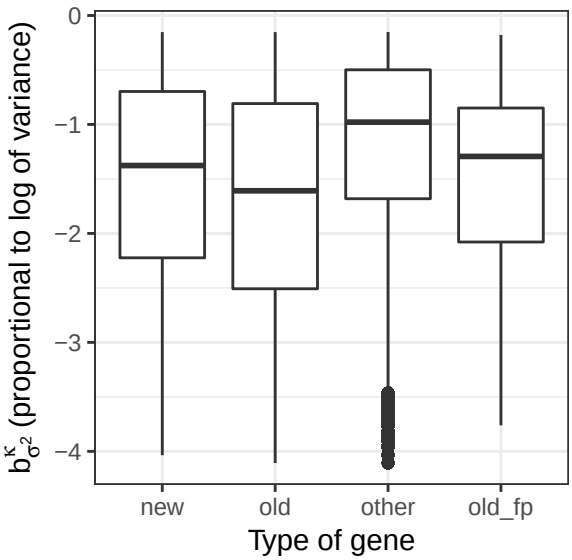


Figure 62: Variance-related bias across all the genes. Each unique gene appears exactly once for every pathway.

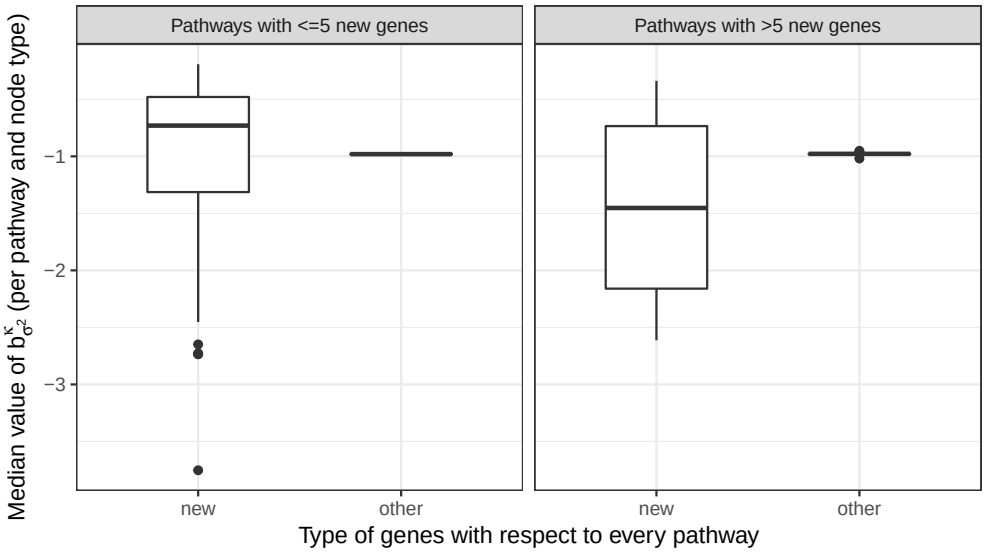


Figure 63: Variance-related bias across all the pathways. The median reference variance of the new and the other genes for each pathway is represented, leading to two data points per pathway. Pathways were divided in two groups according to their number of new genes.

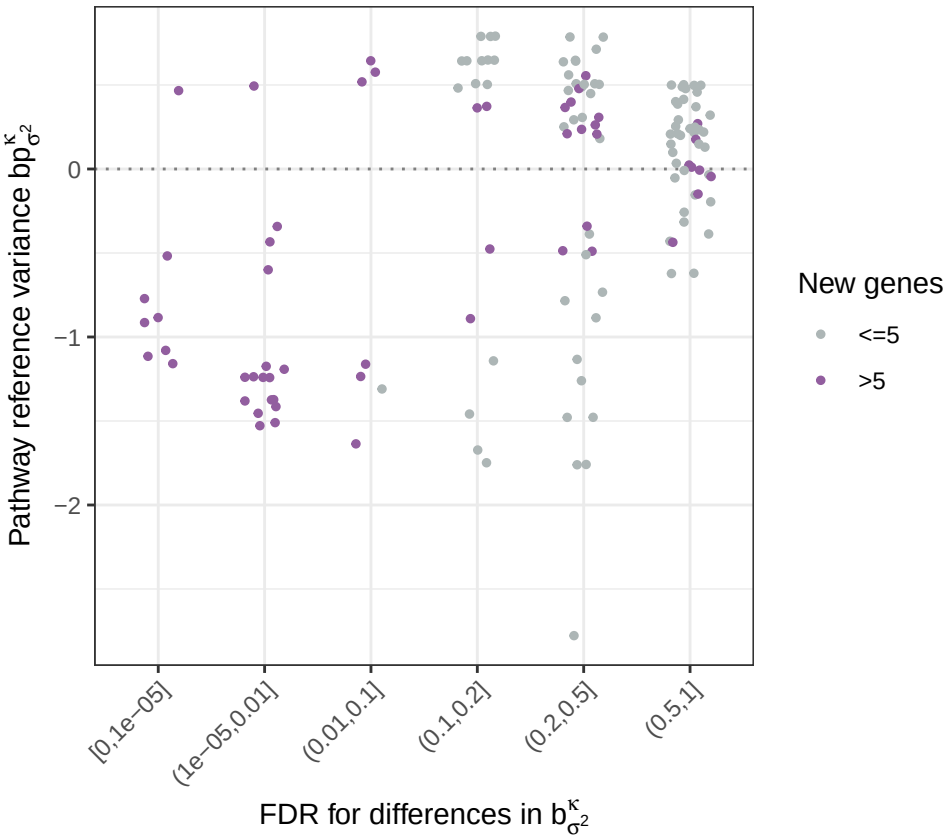


Figure 64: Statistical differences on the reference variances of new versus other genes in each pathway. Each data point represents a pathway. Differences were stratified by the amount of new genes, which affects the statistical power to spot differences.

Diffusion inputs

As the pathways were treated as gene sets, inputs were naturally defined as binary labels without further modification. Note that pathways could contain genes that were present in the old release but dropped in the last one, acting as a *false positive*. In total, 139 instances (one per pathway) were defined and genes outside the original pathway were ranked. Afterwards, the AUROC and AUPRC metrics were computed and compared through explanatory models.

Diffusion scores and bias

Before diving into pathway-wise performance metrics, diffusion scores `raw` and `z` were compared in views of the variance-related bias. Figure 65 sheds light on the expected behaviour of the statistical normalisation and supports the hypothesis that normalising the scores helps decorrelate power from the reference variance values. The actual impact on overall performance still depends on other factors, such as the density of positives throughout the reference variances.

As for method parameters, the regularised (unnormalised) Laplacian kernel was used and permutation-based scores used 10^4 random trials. Methods `ml`, `gm` and `ber_s` were excluded from this comparison because their ranking is identical to that of `raw` in the current settings, see the diffusion scores equivalence properties 1 (`ml`, `gm`) and 3 (`ber_s`) in Supplement 1. Two baselines were considered: `pagerank` (with `damping = 0.85`), which tends to suggest central genes regardless of the input, and `random` (random prioritisation).

A.4.3 Models

Model definition

The metrics AUROC and AUPRC were modelled through dispersion-adjusted quasibinomial logit models, see `?stats::quasibinomial` in an R console:

$$\text{metric} \sim \text{method} + \text{method:path\_var\_ref}$$

The categorical variable `method` could be `raw`, `ber_p`, `mc`, `z` or the baselines `pagerank` and `random`. The term `path_var_ref` was a pathway property, computed as the difference between the median of the reference variance $b^{\mathcal{K}}_{\sigma^2}(i)$ for the *new* genes in the pathway, and the median of $b^{\mathcal{K}}_{\sigma^2}(i)$ for the *other* genes, as depicted in figure 64. `path_var_ref` intended to summarise the bias of a whole pathway in a single number: positive (negative) values indicated that the *new* genes had more (less) variance than the average gene in the network. In order to test our hypothesis, the interaction term `method:path_var_ref` allowed methods to be affected in different ways by the pathway-wise bias.

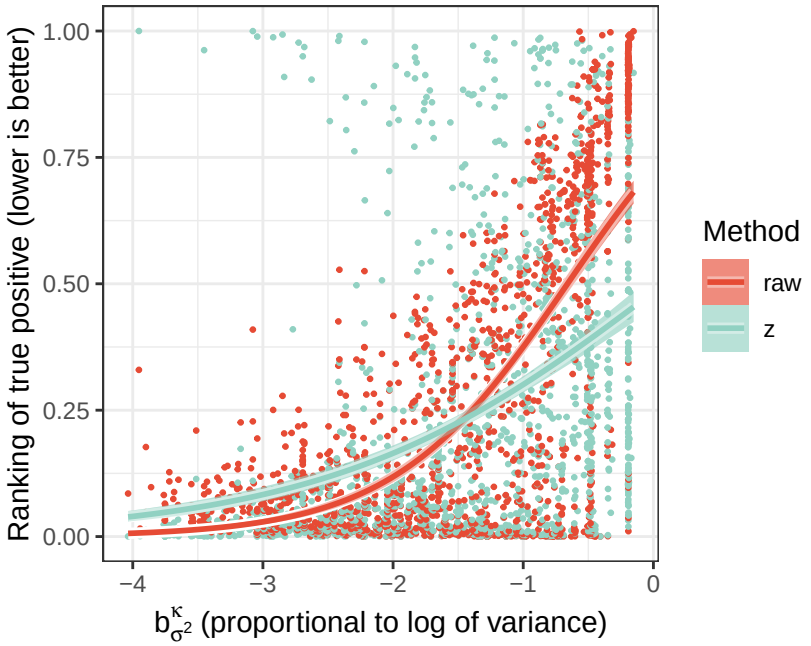


Figure 65: Ranking of true positives as a function of the variance-related bias; lines correspond to a logistic fit with 0.95 confidence intervals. This plot represents the union of the positives of each pathway and their relative ranking in their prioritisation. Nodes closer to 0 were top ranked for that specific pathway, and therefore well prioritised, whereas worst ranked nodes were close to 1. The unnormalised scores raw had more power on nodes with lower standard deviation, at the cost of being less sensitive among larger standard deviations. The normalised scores z showed a more bias-independent power, at the cost of missing positives with smaller standard deviations.

AUROC

Table 21 summarises the AUROC model. As this case study was not simulated, the number of data points was limited due to the prospective design, being notably lower than that of the other datasets.

Table 21: Quasibinomial model for AUROC

methodber_p	0.271^{**} (0.057, 0.486)
methodmc	0.291^{***} (0.079, 0.503)
methodz	0.560^{***} (0.342, 0.779)
methodpagerank	-0.891^{***} (-1.100, -0.681)
methodrandom	-0.558^{***} (-0.758, -0.358)
methodraw:path_var_ref	-1.387^{***} (-1.648, -1.127)
methodber_p:path_var_ref	-1.030^{***} (-1.279, -0.782)
methodmc:path_var_ref	-0.635^{***} (-0.854, -0.417)
methodz:path_var_ref	-0.484^{***} (-0.710, -0.258)
methodpagerank:path_var_ref	-1.473^{***} (-1.695, -1.251)
methodrandom:path_var_ref	0.035 (-0.129, 0.199)
Constant	0.710^{***} (0.559, 0.861)
Observations	834
Note:	*p<0.1; **p<0.05; ***p<0.01

The model in table 21 supported the claim that `raw` was more affected than `z` by the reference variance. Figure 66 reflects this fact along the values of `path_var_ref`, whereas the contrast between the interaction terms (i.e. of the form `method:path_var_ref`) of `raw` and `z` was significant:

```
##  contrast             estimate        SE  df z.ratio p.value
##  raw - ber_p       -0.35722993 0.1837676 Inf  -1.944  0.3752
##  raw - mc          -0.75203314 0.1735438 Inf  -4.333  0.0002
##  raw - z           -0.90326861 0.1761406 Inf  -5.128  <.0001
##  raw - pagerank     0.08536864 0.1747622 Inf   0.488  0.9966
##  raw - random      -1.42200492 0.1570979 Inf  -9.052  <.0001
##  ber_p - mc        -0.39480321 0.1688678 Inf  -2.338  0.1789
##  ber_p - z         -0.54603868 0.1715353 Inf  -3.183  0.0182
##  ber_p - pagerank   0.44259857 0.1701197 Inf   2.602  0.0968
##  ber_p - random    -1.06477498 0.1519165 Inf  -7.009  <.0001
##  mc - z            -0.15123547 0.1605344 Inf  -0.942  0.9356
##  mc - pagerank      0.83740178 0.1590209 Inf   5.266  <.0001
##  mc - random       -0.66997178 0.1393756 Inf  -4.807  <.0001
##  z - pagerank       0.98863725 0.1618508 Inf   6.108  <.0001
##  z - random        -0.51873631 0.1425959 Inf  -3.638  0.0037
##  pagerank - random -1.50737356 0.1408898 Inf -10.699  <.0001
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

Predictions with confidence intervals in the mean value of `path_var_ref` are shown in figure 67, whereas their raw values can be found in figure 68 –

	raw	ber_p	mc	z	pagerank	random
raw		-0.038(-0.06,-0.02)	-0.043(-0.077,-0.016)	-0.084(-0.13,-0.045)	0.16(0.13,0.19)	0.16(0.1,0.21)
ber_p	3.49e-07		-0.0011(-0.01,0.0058)	-0.026(-0.046,-0.011)	0.22(0.18,0.26)	0.22(0.17,0.26)
mc	4.74e-04	7.73e-01		-0.035(-0.053,-0.02)	0.23(0.19,0.28)	0.22(0.18,0.26)
z	5.39e-09	5.16e-04	1.01e-05		0.29(0.24,0.33)	0.26(0.22,0.3)
pagerank	1.72e-17	7.14e-19	3.91e-17	7.14e-19		-0.026(-0.083,0.034)
random	6.66e-07	1.96e-11	8.36e-13	1.71e-17	4.40e-01	

Table 22: Paired two-sided Wilcoxon test between AUROCs, corrected by FDR. Above diagonal: differences with 0.95 confidence interval. Below diagonal: FDR.

both figures depict similar trends. Testing overall differences (averaging over `path_var_ref` and using Tukey's test), `z` significantly outperformed `raw`:

```
##  contrast          odds.ratio         SE  df z.ratio p.value
##  raw / ber_p        0.8158250 0.09847996 Inf  -1.686  0.5409
##  raw / mc           0.8623415 0.10037696 Inf  -1.272  0.8002
##  raw / z            0.6781553 0.08086040 Inf  -3.257  0.0143
##  raw / pagerank     2.3974532 0.27356640 Inf   7.663  <.0001
##  raw / random       2.2891461 0.24679547 Inf   7.682  <.0001
##  ber_p / mc         1.0570178 0.12225956 Inf   0.479  0.9969
##  ber_p / z          0.8312510 0.09851785 Inf  -1.559  0.6254
##  ber_p / pagerank   2.9386857 0.33311868 Inf   9.510  <.0001
##  ber_p / random     2.8059279 0.30027989 Inf   9.641  <.0001
##  mc / z             0.7864116 0.08974768 Inf  -2.105  0.2843
##  mc / pagerank      2.7801668 0.30235276 Inf   9.402  <.0001
##  mc / random        2.6545702 0.27110599 Inf   9.559  <.0001
##  z / pagerank       3.5352568 0.39518139 Inf  11.297  <.0001
##  z / random         3.3755483 0.35560792 Inf  11.548  <.0001
##  pagerank / random  0.9548241 0.09501094 Inf  -0.465  0.9973
##
## P value adjustment: tukey method for comparing a family of 6 estimates
## Tests are performed on the log odds ratio scale
```

A paired non-parametric test outside the model yielded stronger evidence of such differences, see table 22.

Note how by its own definition, the `pagerank` centrality baseline was noticeably affected by the bias, in a similar way to the `raw` scores (figure 66). This was expected because node degree, the most basic measure of centrality, showed collinearity with the reference variance (figure 61). Provided that pathway biases were found in both directions, i.e. genes with either more or less variance than most genes (figure 64), `pagerank` had a close-to-random AUROC (figure 68). On the other hand, the `random` baseline behaved as expected, with an AUROC close to 0.5 and independent of the reference pathway variance (figure 66).

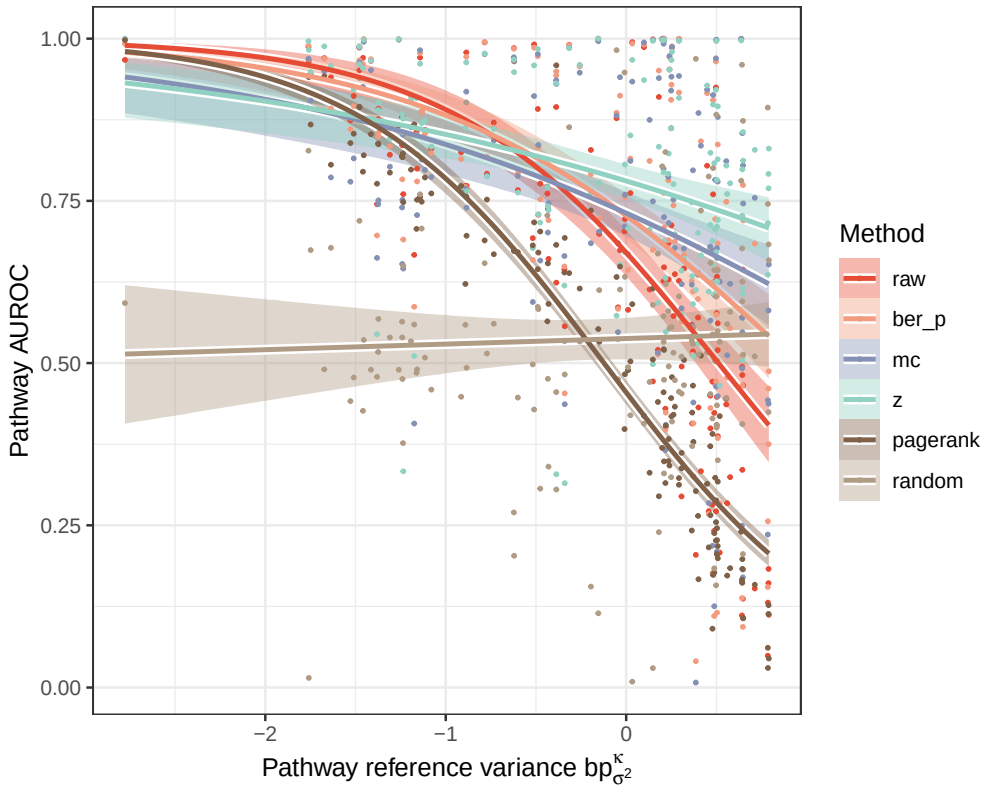


Figure 66: Prediction of the AUROC model by method along the reference pathway variance, represented by path_var_ref. Shaded are the 0.95 confidence intervals for the predicted mean AUROC.

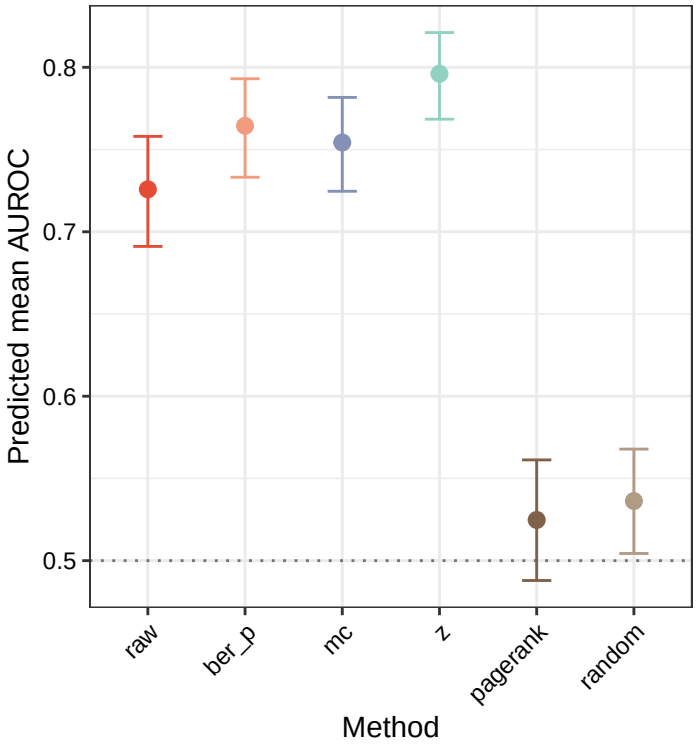


Figure 67: Predictions using the AUROC model (0.95 confidence intervals). Predictions were averaged over the path_var_ref covariate.

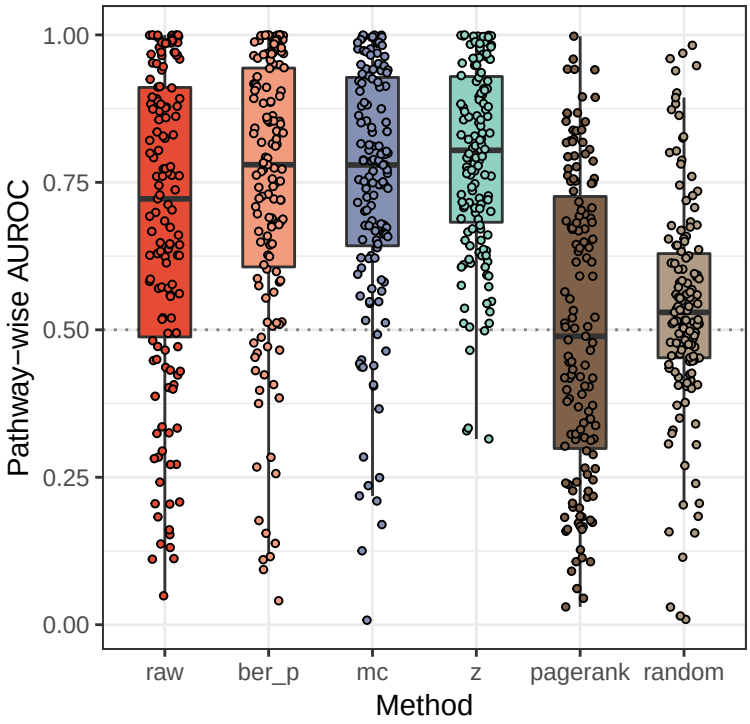


Figure 68: AUROC for all the pathways, by method.

AUPRC

The model on AUPRC pointed out that early retrieval was challenging in this prospective study. Even though proper methods did outperform the baselines, performances were low, also due to the heavy class imbalance.

Table 23 describes the `quasilogistic` model – differences were minimal between methods, lacking statistical support. Furthermore, figure 70 proves how the 0.95 confidence intervals on the mean value of `path_var_ref` are overlapping.

Besides, AUPRC is affected by the class imbalance, meaning that pathways with few new genes were expected to yield low values of AUPRC.

Due to the two reasons above, AUPRC was not useful to describe differences between methods, but to highlight the difficult nature of this prospective analysis. The fact that an old network was used rules out possible circularities, i.e. the new genes being included in the pathways and in new interactions, based on the same data source.

Table 23: Quasibinomial model for AUPRC

methodber_p	0.021 (−0.488, 0.531)
methodmc	-1.054^{***} (−1.783, −0.326)
methodz	-0.554^{*} (−1.169, 0.062)
methodpagerank	-3.246^{***} (−5.171, −1.320)
methodrandom	-3.656^{***} (−5.917, −1.395)
methodraw:path_var_ref	0.002 (−0.450, 0.454)
methodber_p:path_var_ref	0.008 (−0.441, 0.456)
methodmc:path_var_ref	-0.651^{**} (−1.239, −0.063)
methodz:path_var_ref	-0.673^{***} (−1.137, −0.209)
methodpagerank:path_var_ref	−1.062 (−2.507, 0.383)
methodrandom:path_var_ref	−0.332 (−2.709, 2.045)
Constant	-3.239^{***} (−3.601, −2.877)
Observations	834
Note:	*p<0.1; **p<0.05; ***p<0.01

Performing a paired Wilcoxon test yielded no evidence that `raw` and `z` had different AUPRCs (Table 24). The fact that `mc` was actually performing slighly worse than the rest of methods was deemed uninformative, given the actual magnitude of the effect and the overall low AUPRCs.

For completeness, we checked for significant differences between `raw` and `z` in the interaction term, because table 23 may suggest that `z` could be more

	raw	ber_p	mc	z	pagerank	random
raw		-8e-05(-0.00025,1.1e-05)	0.0011(4e-04,0.0029)	8.2e-05(-0.00023,0.00072)	0.0066(0.0035,0.012)	0.008(0.0047,0.017)
ber_p	4.65e-02		0.0013(0.00052,0.0028)	0.00017(-0.00011,0.00075)	0.0067(0.0035,0.013)	0.0083(0.0048,0.018)
mc	1.59e-03	1.45e-05		-8e-04(-0.0026,-0.00014)	0.0046(0.0021,0.0065)	0.0055(0.0032,0.0089)
z	7.01e-01	3.06e-01	3.21e-03		0.0064(0.0035,0.01)	0.0069(0.0039,0.012)
pagerank	2.50e-17	8.34e-18	3.79e-15	2.56e-19		5.3e-05(-4.1e-05,4e-04)
random	1.08e-17	8.82e-19	2.81e-18	7.87e-19	2.34e-01	

Table 24: Paired two-sided Wilcoxon test between AUPRCs, corrected by FDR. Above diagonal: differences with 0.95 confidence interval. Below diagonal: FDR.

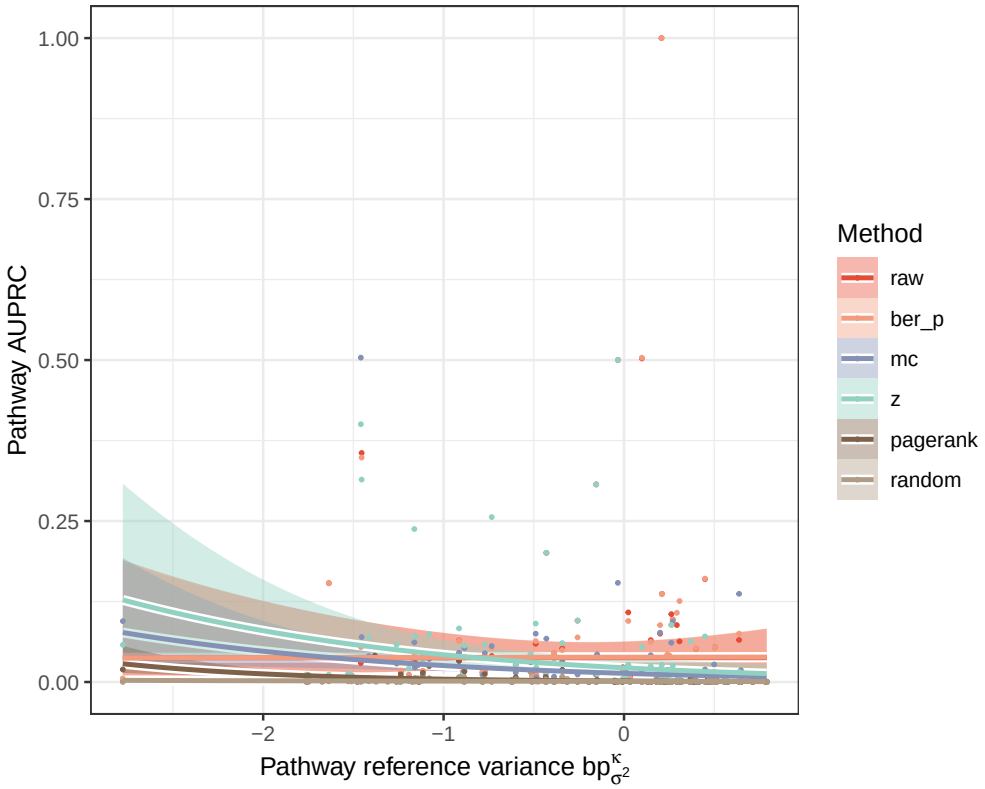


Figure 69: Prediction of the AUPRC model by method along the reference pathway variance, represented by path_var_ref. Shaded are the 0.95 confidence intervals for the predicted mean AUPRC.

affected than `raw` by the reference variances. In line with the other results, this counterintuitive claim could not be proven after a contrast on the interaction term `method:path_var_ref`:

```
##  contrast              estimate        SE  df z.ratio p.value
##  raw - ber_p       -0.005667778 0.3248737 Inf  -0.017  1.0000
##  raw - mc           0.652724459 0.3784352 Inf   1.725  0.5152
##  raw - z            0.674948860 0.3303838 Inf   2.043  0.3179
##  raw - pagerank     1.064137045 0.7723649 Inf   1.378  0.7405
##  raw - random       0.333723339 1.2343521 Inf   0.270  0.9998
##  ber_p - mc         0.658392237 0.3773436 Inf   1.745  0.5019
##  ber_p - z          0.680616638 0.3291329 Inf   2.068  0.3042
##  ber_p - pagerank   1.069804823 0.7718306 Inf   1.386  0.7355
##  ber_p - random     0.339391117 1.2340179 Inf   0.275  0.9998
##  mc - z             0.022224401 0.3820977 Inf   0.058  1.0000
##  mc - pagerank      0.411412586 0.7958597 Inf   0.517  0.9955
##  mc - random       -0.319001120 1.2491879 Inf  -0.255  0.9999
##  z - pagerank       0.389188185 0.7741660 Inf   0.503  0.9961
##  z - random        -0.341225521 1.2354800 Inf  -0.276  0.9998
##  pagerank - random -0.730413706 1.4190859 Inf  -0.515  0.9956
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

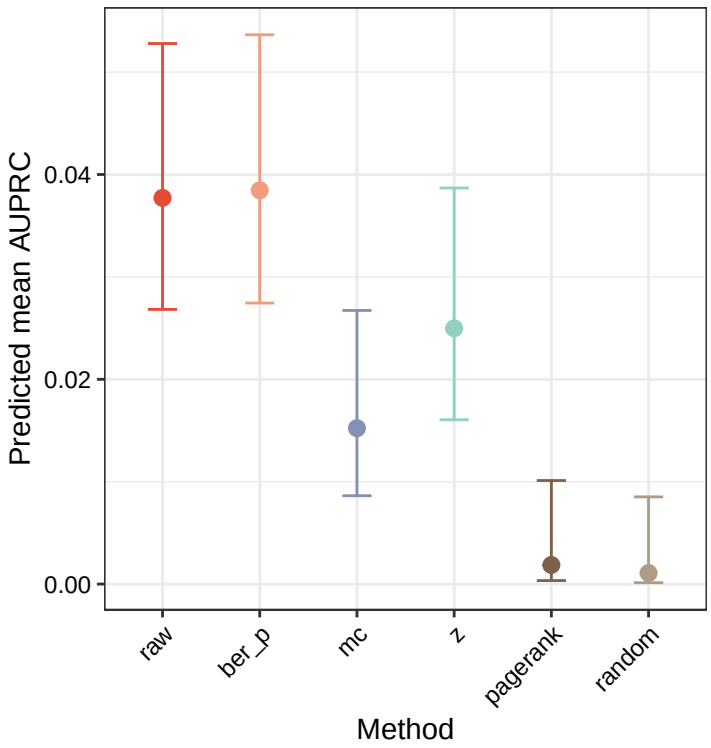


Figure 70: Predictions using the AUPRC model (0.95 confidence intervals). Predictions were averaged over the path_var_ref covariate.

Other remarks

The present case study served as an illustrative example of variance-related bias in diffusion scores.

- The effect of the bias correction was not as straightforward as the mean value-related bias. We hypothesised that `z` would have more power on low-variance nodes compared to `raw`, but our findings support the opposite. The counterintuitive nature of this bias encourages an additional layer of caution.
- Normalising the diffusion scores led to a more bias-independent power for AUROC, in line with our hypothesis.
- AUROC was more informative than AUPRC and helped identify bias-related trends in predictive power.
- Again, the overall performance, and therefore the decision on normalising, relied on the distribution of the positives with respect to the reference variance. In this particular instance, `z` outperformed `raw`.
- For all the methods, new positives with higher variances were harder to recover, although this was less pronounced in `z`. High variance nodes tended to have a low degree, so we speculate that the network was incomplete when describing their biology, thus limiting the performance in their respective pathways.

A.4.4 Reproducibility

```
##  [1] "R version 3.5.3 (2019-03-11)"
##  [2] "Platform: x86_64-pc-linux-gnu (64-bit)"
##  [3] "Running under: Ubuntu 16.04.6 LTS"
##  [4] ""
##  [5] "Matrix products: default"
##  [6] "BLAS: /usr/lib/atlas-base/atlas/libblas.so.3.0"
##  [7] "LAPACK: /usr/lib/atlas-base/atlas/liblapack.so.3.0"
##  [8] ""
##  [9] "locale:"
## [10] " [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C              "
## [11] " [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8    "
## [12] " [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8   "
## [13] " [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                 "
## [14] " [9] LC_ADDRESS=C               LC_TELEPHONE=C            "
## [15] "[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C       "
## [16] ""
## [17] "attached base packages:"
## [18] "[1] grid      stats     graphics  grDevices utils     datasets  methods
## [19] "[8] base     "
## [20] ""
## [21] "other attached packages:"
## [22] " [1] stargazer_5.2.2   emmeans_1.3.0     bindrcpp_0.2.2   "
## [23] " [4] xtable_1.8-3      data.table_1.11.8 extrafont_0.17   "
## [24] " [7] gtable_0.2.0      GGally_1.4.0      ggsci_2.9        "
## [25] "[10] ggplot2_3.1.0     tidyr_0.8.2       dplyr_0.7.8      "
## [26] "[13] plyr_1.8.4        reshape2_1.4.3    magrittr_1.5     "
## [27] "[16] diffuStats_1.2.0  igraphdata_1.0.1  igraph_1.2.2     "
## [28] "[19] rmarkdown_1.10   "
## [29] ""
## [30] "loaded via a namespace (and not attached):"
## [31] " [1] Rcpp_1.0.0                mvtnorm_1.0-8            "
## [32] " [3] lattice_0.20-38           zoo_1.8-4                "
## [33] " [5] assertthat_0.2.0          rprojroot_1.3-2          "
## [34] " [7] digest_0.6.18             packrat_0.5.0            "
## [35] " [9] R6_2.3.0                  backports_1.1.2          "
## [36] "[11] evaluate_0.12             pillar_1.3.0             "
## [37] "[13] rlang_0.3.0.1             lazyeval_0.2.1           "
## [38] "[15] multcomp_1.4-8            extrafontdb_1.0          "
## [39] "[17] Matrix_1.2-15             labeling_0.3             "
## [40] "[19] splines_3.5.3             stringr_1.3.1            "
## [41] "[21] munsell_0.5.0             compiler_3.5.3           "
## [42] "[23] pkgconfig_2.0.2           mgcv_1.8-27              "
## [43] "[25] htmltools_0.3.6           tidyselect_0.2.5         "
## [44] "[27] tibble_1.4.2              expm_0.999-3             "
## [45] "[29] codetools_0.2-16          reshape_0.8.8            "
## [46] "[31] crayon_1.3.4              withr_2.1.2              "
## [47] "[33] MASS_7.3-51.1             nlme_3.1-137             "
```

```
## [48] "[35] Rttf2pt1_1.3.7           scales_1.0.0             "
## [49] "[37] RcppParallel_4.4.1       estimability_1.3         "
## [50] "[39] stringi_1.2.4            RcppArmadillo_0.9.200.4.0"
## [51] "[41] sandwich_2.5-0           TH.data_1.0-9            "
## [52] "[43] RColorBrewer_1.1-2       tools_3.5.3              "
## [53] "[45] glue_1.3.0               purrr_0.2.5              "
## [54] "[47] survival_2.43-3          yaml_2.2.0               "
## [55] "[49] colorspace_1.3-2         corrplot_0.84            "
## [56] "[51] knitr_1.20               bindr_0.1.1              "
## [57] "[53] precrec_0.9.1           "
```

REFERENCES

Benjamini, Yoav and Yosef Hochberg

1995 "Controlling the false discovery rate: a practical and powerful approach to multiple testing", *Journal of the royal statistical society. Series B (Methodological)*, pp. 289-300.

Cao, Mengfei, Christopher M Pietras, Xian Feng, Kathryn J Doroschak, Thomas Schaffner, Jisoo Park, Hao Zhang, Lenore J Cowen, and Benjamin J Hescott

2014 "New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence", *Bioinformatics*, 30, 12, pp. i219-i227.

Carlson, Marc

2016 *KEGG.db: A set of annotation maps for KEGG*, R package version 3.2.3.

Chatr-aryamontri, Andrew, Rose Oughtred, Lorrie Boucher, Jennifer Rust, Christie Chang, Nadine K Kolas, Lara O'Donnell, Sara Oster, Chandra Theesfeld, Adnane Sellam, et al.

2017 "The BioGRID interaction database: 2017 update", *Nucleic acids research*, 45, D1, pp. D369-D379.

Chiaretti, Sabina, Xiaochun Li, Robert Gentleman, Antonella Vitale, Marco Vignetti, Franco Mandelli, Jerome Ritz, and Robin Foa

2004 "Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival", *Blood*, 103, 7, pp. 2771-2778.

Csardi, Gabor

2015 *igraphdata: A Collection of Network Data Sets for the 'igraph' Package*, R package version 1.0.1, https://CRAN.R-project.org/package=igraphdata.

Dittrich, Marcus T, Gunnar W Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Müller

2008 "Identifying functional modules in protein–protein interaction networks: an integrated exact approach", *Bioinformatics*, 24, 13, pp. i223-i231.

Dittrich, Marcus and Daniela Beisser

2010 *DLBCL: Diffuse large B-cell lymphoma expression data*, R package version 1.16.0, http://bionet.bioapps.biozentrum.uni-wuerzburg.de/.

Kanehisa, Minoru, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima

2017 "KEGG: new perspectives on genomes, pathways, diseases and drugs", *Nucleic acids research*, 45, D1, pp. D353-D361.

Li, Xiaochun

2009 *ALL: A data package*, R package version 1.20.0.

Mishra, Gopa R, M Suresh, K Kumaran, N Kannabiran, Shubha Suresh, P Bala, K Shivakumar, N Anuradha, Raghunath Reddy, T Madhan Raghavan, et al.

2006 "Human protein reference database—2006 update", *Nucleic acids research*, 34, suppl_1, pp. D411-D414.

Picart-Armada, Sergio and Thompson, Wesley K and Buil, Alfonso and Perera-Lluna, Alexandre

2017 "diffuStats: an R package to compute diffusion-based scores on biological networks", *Bioinformatics*, 34, 3, pp. 533-534.

Rosenwald, Andreas, George Wright, Wing C Chan, Joseph M Connors, Elias Campo, Richard I Fisher, Randy D Gascoyne, H Konrad Muller-Hermelink, Erlend B Smeland, Jena M Giltnane, et al.

2002 "The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma", *New England Journal of Medicine*, 346, 25, pp. 1937-1947.

Smola, Alexander J and Risi Kondor

2003 "Kernels and regularization on graphs", in *Learning theory and kernel machines*, Springer, pp. 144-158.

Von Mering, Christian, Roland Krause, Berend Snel, Michael Cornell, Stephen G Oliver, Stanley Fields, and Peer Bork

2002 "Comparative assessment of large-scale data sets of protein–protein interactions", *Nature*, 417, 6887, p. 399.

B | THE R PACKAGE DIFFUSTATS

DIFFUSTATS: AN R PACKAGE TO COMPUTE DIFFUSION-BASED SCORES ON BIOLOGICAL NETWORKS

B.1 ABSTRACT

Label propagation approaches are a standard and ubiquitous procedure in computational biology for giving context to molecular entities. Node labels, which can derive from gene expression, genome-wide association studies, protein domains or metabolomics profiling, are propagated to their neighbours, effectively smoothing the scores through prior annotated knowledge and prioritising novel candidates. However, there are several settings to tune when defining the diffusion process, including the diffusion kernel, the numeric codification of the labels and a choice of statistical normalisation of the scores. These settings can have a large impact on results, and there is currently no software implementing many of them in one place to screen their performance in the application of interest. This vignette presents *diffuStats*, an R package with a collection of diffusion kernels and scores, as well as a parallel permutation analysis for the normalised scores, that eases the analysis of several sets of molecular entities at once.

B.2 INTRODUCTION

The application of label propagation algorithms (Zoidi et al., 2015) is based on the guilt by association principle (Oliver, 2000), which can be rephrased in the protein-protein interaction context as "proteins that interact are more likely to share biological functions". However, this principle is extremely general and has experienced success in numerous applications in bioinformatics.

HotNet (Vandin et al., 2010) uses a diffusion process with mutated genes as seed nodes to find modules with a statistically high number of mutated genes in cancer. Another attempt to find relevant modules from gene expression and mutation data can be found in (Bersanelli et al., 2016), where

This appendix is based on the main vignette of the diffuStats package (https://doi.org/doi:10.18129/B9.bioc.diffuStats, accessed 31/12/2019), supplementary data of: Picart-Armada, Sergio, Wesley K. Thompson, Alfonso Buil, and Alexandre Perera-Lluna. "diffuStats: an R package to compute diffusion-based scores on biological networks". *Bioinformatics 34, no. 3 (2018): 533-534.*

the authors propose a diffusion process followed by a statistical normalisation and an automatic process to extract a subnetwork. TieDIE (Paull et al., 2013) runs two diffusion processes to link perturbation in the genome with changes in the transcriptome, effectively linking two sets of genes. GeneMANIA (Mostafavi et al., 2008) is a web server that predicts gene function using label propagation with a bias on the unlabelled nodes. Network-based learning sharing a background with diffusion has been also applied to protein classification using multiple networks (Tsuda et al., 2005). Label propagation using graph kernels has been proven successful in gene-disease association (Lee et al., 2011; Valentini et al., 2014). Equivalent formulations can be found under different terminology, like the electrical model applied to prioritise candidate genes in eQTL in (Suthram et al., 2008).

The heterogeneity of applications hinders comparisons among approaches, therefore tools gathering the state of the art are highly needed. An existing solution is *RANKS*, an R package that contains a variety of diffusion kernels and kernelised scores for label propagation using a binary input vector. *RANKS* eases kernelised scores benchmarking and models it as a "one-class" classification semi-supervised learning problem, in which only some members of the class (positives) are known. Another possibility is to divide nodes into labelled positive, negative and unlabelled, like in (Mostafavi et al., 2008), which poses questions like the effect of unlabelled nodes and possible numeric codifications of the labels, or the option to include quantitative data in the labels. In addition, statistical normalisations such as in (Bersanelli et al., 2016) remove the effect of network structures like hubs and should be taken into account when choosing a diffusion scoring method. This motivates the introduction of our *diffuStats* R package, which collects widely adopted input codifications and explicitly accounts for unlabelled nodes. It also includes three statistically normalised scores, which can be obtained through Monte Carlo trials or a parametric alternative. The *diffuStats* package uses existent classes and provides high-level functions to screen the performance of several diffusion scores, in order to facilitate their integration in any computational biology study.

B.3 METHODOLOGY

One of the main purposes of *diffuStats* is to offer a battery of approaches to compute and compare diffusion scores. The diffusion scores f using an input vector y and a diffusion kernel K are generally computed as

$$f = K \cdot y$$

possibly followed by further adjustments or a statistical normalisation.

The decisions taken in the definition of K, y and the posterior normalisation generally give rise to different priorisations due to a different treatment of the balance between positive and negative examples, the unlabelled data and the network structure. The following sections cover the implemented choices for the kernel K and the initial labels y.

Kernel	Function
Regularised Laplacian	$r(\lambda) = 1 + \sigma^2\lambda$
Diffusion process	$r(\lambda) = \exp(\frac{\sigma^2}{2}\lambda)$
p-Step random walk	$r(\lambda) = (a - \lambda)^{-p}$ with $a \geqslant 2$, $p \geqslant 1$
Inverse cosine	$r(\lambda) = (\cos(\lambda\frac{\pi}{4}))^{-1}$

Table 25: Implemented diffusion kernels from (Smola and Kondor, 2003)

B.3.1 Diffusion kernels and regularisation

The representation of any kind of data in a network model allows the definition of notions like distance or similarity based on the links in the network. This section will follow the notation in (Smola and Kondor, 2003) and summarise the kernels proposed by the authors. In general, an undirected graph $G = (V, E)$ consists of a set of n nodes V and a set of edges E of unordered pairs of nodes. This can be extended to weighted, undirected graphs, where each edge $i \sim j$ has a weight attribute $W_{ij} \in [0, \infty)$. The degree matrix of G is defined as the $n \times n$ diagonal matrix so that $D_{ii} = \sum_{j=1}^{n} W_{ij}$. The (unnormalised) Laplacian of G is defined as the $n \times n$ matrix $L = D - W$, whereas its normalised version is $\tilde{L} = D^{-\frac{1}{2}} \cdot L \cdot D^{-\frac{1}{2}}$.

The graph Laplacian is diagonalisable and can be written in terms of its eigenvalues ν_j and eigenvectors λ_j, as $L = \sum_{j=1}^{n} \lambda_j \nu_j \nu_j^T$. The proposed kernels stem from a family of regularisation functions $r(\lambda)$ on the spectrum of the graph Laplacian:

$$K = \sum_{j=1}^{n} r^{-1}(\lambda_j)\nu_j\nu_j^T$$

Well known graph kernels belong to this family because they can be written as transformations on the Laplacian spectrum. Table 25 summarises them, assuming the usage of the normalised Laplacian - the unnormalised Laplacian can also be used as long as the resulting kernel is still positive semidefinite. Further details about this family of kernels, all available in our package *diffuStats*, can be found in the original manuscript (Smola and Kondor, 2003).

Additionally, the *diffuStats* package includes the commute time kernel, introduced in (Yen et al., 2007). This kernel, also writable in terms of a regularisation function, is simply the pseudoinverse of the graph Laplacian, $K = L^+$.

The default option in the *diffuStats* package is the regularised Laplacian kernel, as it is widely adopted and describes many physical models, for instance in (Paull et al., 2013; Suthram et al., 2008; Vandin et al., 2010).

B.3.2 Diffusion scores

Besides choosing a graph kernel, the codification of the input and the presence of a statistical normalisation can lead to important differences in the results. Table 26 gives an overview of the implemented scores, which will be detailed in the following sections. The argument `method` in the

Score	y^+	y^-	y^u	Normalised	Stochastic	Quantitative	Reference
raw	1	0	0	No	No	Yes	(Vandin et al., 2010)
ml	1	-1	0	No	No	No	(Tsuda et al., 2005; Zoidi et al., 2015)
gm	1	-1	k	No	No	No	(Mostafavi et al., 2008)
ber_s	1	0	0	No	No	Yes	(Bersanelli et al., 2016)
ber_p	1	0	0*	Yes	Yes	Yes	(Bersanelli et al., 2016)
mc	1	0	0*	Yes	Yes	Yes	(Bersanelli et al., 2016)
z	1	0	0*	Yes	No	Yes	(Harchaoui et al., 2013)

Table 26: Implemented diffusion scores

function `diffuse` can be set to the desired scores in table 26, which are described in the following sections. The numeric values of the positive, negative and unlabelled examples are respectively y^+, y^- and y^u. Column "normalised" refers to the application of a statistical model involving permutations, whereas "stochastic" enumerates the normalised scores that need actual Monte Carlo permutations. The scores that also accept quantitative inputs instead of binary labels are listed in the "quantitative" column.

Scores without statistical normalisation

The base diffusion score `raw`, which has been used in algorithms like HotNet (Vandin et al., 2010) and TieDIE (Paull et al., 2013), solves a diffusion problem in terms of the regularised Laplacian kernel (Smola and Kondor, 2003).

$$f_{raw} = K \cdot y_{raw}$$

K is the kernel matrix and y_{raw} the vector of codified inputs. In general, the i-th component of y equals y^+ if node i is a positive, y^- if i is a negative and y^u if i is unlabelled. In the particular case of y_{raw}, the positively labelled nodes introduce one flow unit in the network ($y^+_{raw} = 1$), whereas the negative and unlabelled nodes are treated equivalently and do not introduce anything ($y^-_{raw} = y^u_{raw} = 0$). In the physical model using the regularised Laplacian kernel, the flow can be evacuated from the graph due to the presence of first-order leaking in every node, see (Vandin et al., 2010) for further details on this. This formulation is proportional up to a scaling factor to the average score in *RANKS*.

On the other hand, the classical label propagation (Zoidi et al., 2015) treats positives as $y^+_{ml} = 1$ and negatives as $y^-_{ml} = -1$, while unlabelled nodes remain as $y^u_{ml} = 0$, thus making a distinction between the last two. A biological example can be found in protein classification (Tsuda et al., 2005). This option is available as `ml`, and intuitively scores a node by counting if the majority of its neighbours vote positive or negative:

$$f_{ml} = K \cdot y_{ml}$$

The authors of GeneMANIA (Mostafavi et al., 2008) propose a modification on the `ml` input - they adhere to $y^+_{gm} = 1$ and $y^-_{gm} = -1$, but introduce a bias term in the unlabelled nodes

$$y^u_{gm} = \frac{n^+ - n^-}{n^+ + n^- + n^u}$$

being n^+, n^- and n^u the number of positives, negatives and unlabelled entities. The `gm` score is then computed through

$$f_{gm} = K \cdot y_{gm}$$

The last option in this part, named `ber_s` (Bersanelli et al., 2016), is a quantification of the relative change in the node score before and after the network smoothing. The score for a particular node i can be written as

$$f_{ber_s,i} = \frac{f_{raw,i}}{y_{raw,i} + \epsilon}$$

where $\epsilon > 0$ is a parameter that regulates the importance of the relative change.

Scores with statistical normalisation

Recently, the combination of a permutation analysis with diffusion processes has been suggested (Bersanelli et al., 2016). This is a way to quantify how the diffusion score of a certain node compares to its score if the input was randomised - nodes that might have systematically high or low scores regardless of the input are normalised accordingly.

The cornerstone of normalised scores is the empirical p-value (North et al., 2002) that indicates, for a node i, the proportion of input permutations that led to a diffusion score as high or higher than the original diffusion score. Specifically, f_{raw} is compared to scores from random trials j, $f_{raw}^{null,j} = K \cdot \pi_j(y_{raw})$, where $\pi_j(y_{raw})$ is a permutation of y_{raw} on the labelled entities. The empirical p-value for node i is therefore defined as in (North et al., 2002):

$$p_i = \frac{r_i + 1}{n + 1}$$

being r_i the number of trials j in which $f_{raw,i}^{null,j} \geqslant f_{raw,i}$, and n the total number of trials.

To be consistent with the increasing direction of the scores, the `mc` scores are defined as

$$f_{mc,i} = 1 - p_i$$

Importantly, the permutation has been applied only to the observed nodes. Therefore, any node outside the labelled background cannot receive a different input score in a permuted input. This implies that even if both negatives and unlabelled nodes are assigned the same input value ($y^- = y^u = 0$), the negatives are actually permuted and eventually exchanged for a 1 in some permutations provided that the number of runs is large enough. For this reason, negatives and unlabelled nodes are not equivalent in these scores.

A parametric alternative to f_{mc}, are z scores, where each node i is scored as:

$$f_{z,i} = \frac{f_{raw,i} - E(f_{raw,i}^{null,j})}{\sqrt{V(f_{raw,i}^{null,j})}}$$

The expectation and variance are computed in a closed form and these scores do not require running actual permutations, therefore saving on computational time and avoiding stochastic models. The analytical expression proven below also works for the general case when the input has quantitative labels.

First of all, note that all the unlabelled nodes will cancel out when substracting the expected value, so they can be excluded without loss of generality. Thus, let the row vector $\tilde{k}_i$ be the i-th row of the submatrix from K including the columns indexed by the labelled nodes only, and let n_{lab} be the number of labelled nodes. Analogously, let $\tilde{y}$ be the (quantitative or binary) inputs for the observed nodes, with the same indexing as $\tilde{k}_i$. Consider the following scalar quantities:

$$Sk_i^I = \sum_{j=1}^{n_{lab}} (\tilde{k}_i)_j$$

$$Sk_i^{II} = \sum_{j=1}^{n_{lab}} (\tilde{k}_i)_j^2$$

$$Sy^I = \sum_{j=1}^{n_{lab}} \tilde{y}_j$$

$$Sy^{II} = \sum_{j=1}^{n_{lab}} \tilde{y}_j^2$$

Using these definitions and notation, the raw score for node i is

$$f_i = \tilde{k}_i \cdot \tilde{y}$$

Its null score using a permuted input score is the random variable

$$f_i^{null} = \tilde{k}_i \cdot X$$

where X is the random variable that arises from permuting $\tilde{y}$. In order to compute the z-scores, the expected value and variance of f_i^{null} are needed. Starting with its expected value,

$$E_X(f_i^{null}) = E_X(\tilde{k}_i \cdot X) = \tilde{k}_i \cdot E_X(X) = \tilde{k}_i \cdot \frac{\sum_{j=1}^{n_{lab}} \tilde{y}_j}{n_{lab}} \cdot \mathbb{1} = \frac{Sk_i^I \cdot Sy^I}{n_{lab}}$$

where $\mathbb{1}$ is the n_{lab}-th dimensional column vector full of ones. Regarding its variance,

$$V_X(f_i^{null}) = V_X(\tilde{k}_i \cdot X) = \tilde{k}_i \cdot V_X(X) \cdot \tilde{k}_i^T$$

The covariance of X can be written as

$$V_X(X) = \left[\frac{\sum_{j=1}^{n_{lab}} \tilde{y}_j^2}{n_{lab}} - \left(\frac{\sum_{j=1}^{n_{lab}} \tilde{y}_j}{n_{lab}} \right)^2 \right] \left[\frac{n_{lab}}{n_{lab}-1} Id - \frac{1}{n_{lab}-1} \mathbb{1}\mathbb{1}^T \right]$$

being Id the $n_{lab} \times n_{lab}$ identity matrix. Operating, the variance of f_i^{null} can be finally computed as

$$V_X(f_i^{null}) = \frac{1}{(n_{lab}-1)n_{lab}^2}\left[n_{lab}Sy^{II} - (Sy^{I})^2\right]\left[n_{lab}Sk_i^{II} - (Sk_i^{I})^2\right]$$

The z-score for node i is, in terms of the new notation:

$$f_{z,i} = \frac{f_i - E_X(f_i^{null})}{\sqrt{V_X(f_i^{null})}}$$

This closes the `z` scoring option, which clearly does not require any actual permutations but normalises through the theoretical first and second statistical moments of the null distribution of the diffusion scores.

Finally, the authors in (Bersanelli et al., 2016) also suggest a combination of a classical score with an statistically normalised one. Available as `ber_p`, the score of node i is defined as

$$f_{ber_p,i} = -\log_{10}(p_i) \cdot f_{raw,i}$$

This approach corrects the original diffusion scores by the effect of the network, in order to mitigate the effect of structures like hubs.

Quantitative inputs

In its current release, *diffuStats* also accepts quantitative labels as input in the following scores: `raw`, `ber_s`, `z`, `ber_p` and `mc`. The scores `ml` and `gm` are naturally excluded from non-binary inputs due to their definitions. Currently `mc` scores (and therefore `ber_p` scores as well) accept quantitative inputs that are treated as sparse. Dense continuous inputs might take more time to compute, but this will be extended in future versions. Beware that quantitative inputs should be meaningful and one-tailed (monotonic) - the nature of diffusion processes involves averaging and strong effects with opposing signs cancel out instead of adding up.

B.3.3 Implementation, functions and classes

The package *diffuStats* is mainly implemented in R (R Core Team, 2017), but takes advantage of existing classes in *igraph* (Csardi and Nepusz, 2006) and basic data types, thus not introducing any new data structure - this minimises the learning effort by the final user. Inputs and outputs are conceived to require minimal formatting effort in the whole analysis. The computationally intense stochastic part of the permutation analysis is implemented in C++ through the packages *Rcpp* (Eddelbuettel, 2013), *RcppArmadillo* (Eddelbuettel and Sanderson, 2014) and parallelised through *RcppParallel* (Allaire et al., 2016). Package *diffuStats* also contains documented functions and unit testing with small cases to spot potential bugs. Two vignettes facilitate the user experience: an introductory vignette showing the basic usage of the functions on a synthetic example and this vignette, which further describes the contents of the package and shows an application to real data.

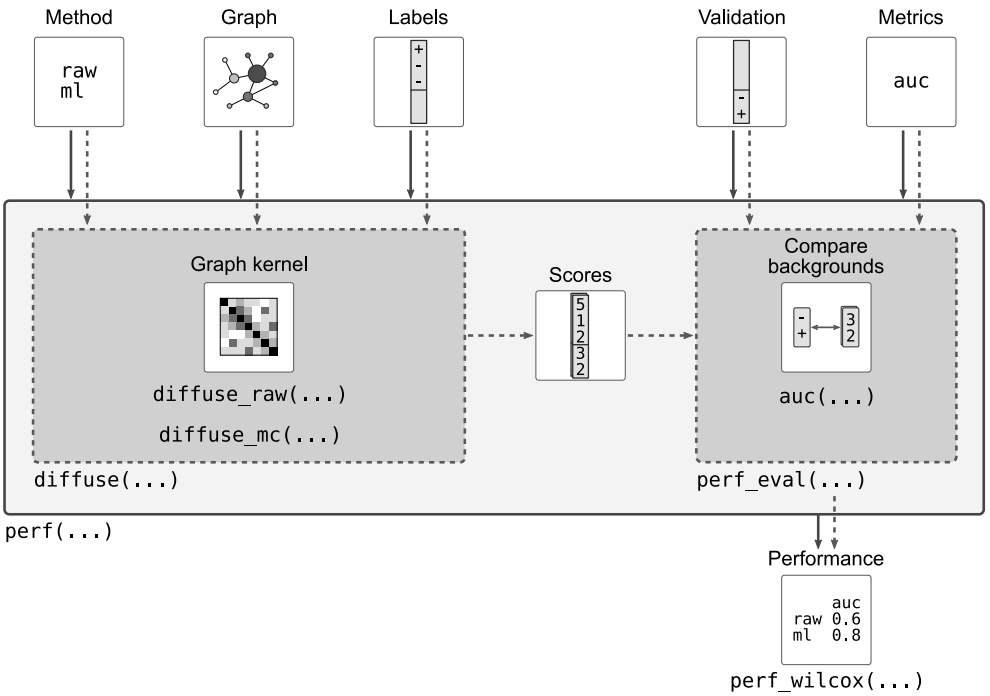


Figure 71: Overview of the main package functions.

A diagram containing the main functions in the R package *diffuStats* can be found in (Fig. 71). The main funcion is `diffuse`, a wrapper for computing diffusion scores from several categories at once, stemming from possibly different observed backgrounds. Function `diffuse` makes use of the deterministic `diffuse_raw` or the stochastic `diffuse_mc` implementations and combines them to give the desired scores for all the nodes in each of the observed backgrounds. The second wrapper `perf` compares the result of the diffusion scores to target validation scores. Validation scores might include only part of the nodes of the network and these nodes can be background-specific.

Regarding memory and processing power requirements, the analysis of networks with more than 10,000 nodes might need additional RAM memory and processing power. The main reason is the manipulation of dense graph kernel matrices that scale quadratically with the network order. To give a reference, a dense matrix of double-precision real numbers with 10,000 rows and columns uses roughly 800MB of memory. Computing a graph kernel on a large network can require -depending on the kernel- matrix diagonalisation, matrix products and matrix inversion operations that are likely to use a considerable amount of memory and time.

B.3.4 Limitations

The kernel framework is known to scale poorly with the number of nodes of the network when the kernel is explicitly computed and dense. Therefore, *diffuStats* is best suited for manipulating biological networks of a medium size - few tenths of thousands of nodes. Protein-protein interaction networks can have around 20,000 nodes, which is also the limit of the capabilities

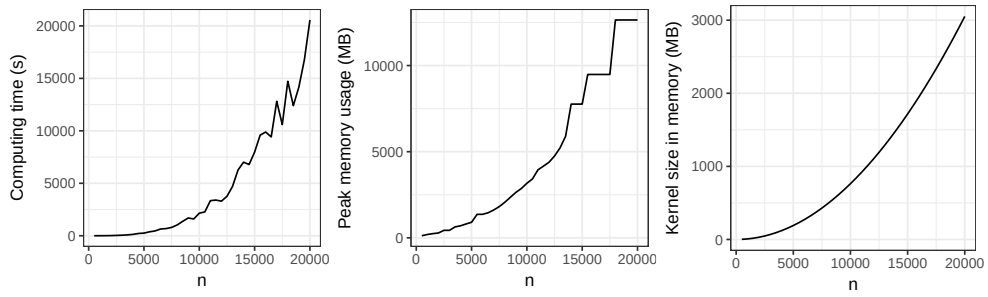


Figure 72: Profiling of the computation of the regularised Laplacian kernel on a synthetic n-th order network. Undirected networks are generated through `barabasi.game` in *igraph* using default parameters and `m = 6` (each node adds 6 edges).

diffuStats, as it is now, in a standard workstation with 16GB of physical memory.

In particular, the explicit kernel computation is a demanding step, although it only has to be performed once with a given network. Figure 72 contains a profiling on the kernel computation using a Compaq q8100 workstation (intel core i5 650 @3.20GHz, 16GB physical memory). This should give an approximation of what to expect in terms of time and memory consumption. The same figure contains the memory usage of the kernel matrix per se.

On the other hand, the parallel implementation of the stochastic permutation analysis is another demanding task. It has been optimised assuming that the number of positives is usually low. Lists of scores with a very high amount of positives might slow down the permutation analysis, but not the parametric z.

B.4 GETTING STARTED

This vignette contains a classical example of label propagation on a biological network. The core tools for this analysis are the *igraph* R package (Csardi and Nepusz, 2006) and the *diffuStats* package, whereas *ggplot2* (Wickham, 2009) is a convenient tool to plot the results.

The data for this example is the `yeast` interactome with functional annotations, as found in the data package `igraphdata` (Csardi, 2015).

```
# Core
library(igraph)
library(diffuStats)

# Plotting
library(ggplot2)
library(ggsci)

# Data
library(igraphdata)
```

```
data(yeast)
set.seed(1)
```

B.4.1 Data description

A summary of the network object can be obtained by just showing the object:

```
summary(yeast)

## IGRAPH 65c41bb UN-- 2617 11855 -- Yeast protein interactions, von Mering et
## + attr: name (g/c), Citation (g/c), Author (g/c), URL (g/c), Classes
## | (g/x), name (v/c), Class (v/c), Description (v/c), Confidence
## | (e/c)
```

For this analysis, only the largest connected component of this graph will be used, although the algorithms can handle graphs with several connected components.

```
yeast <- diffuStats::largest_cc(yeast)
```

This yields to a graph with 2375 nodes and 11693 edges. There are several attributes that can be of interest. First of all, the `name` of the protein nodes:

```
head(V(yeast)$name)

## [1] "YLR197W" "YOR039W" "YDR473C" "YOR332W" "YER090W" "YDR394W"
```

Furthermore, the corresponding aliases and complete names can be found in `Description`

```
head(V(yeast)$Description)

## [1] "SIK1 involved in pre-rRNA processing"
## [2] "CKB2 casein kinase II beta' chain"
## [3] "PRP3 essential splicing factor"
## [4] "VMA4 H+-ATPase V1 domain 27 KD subunit, vacuolar"
## [5] "TRP2 anthranilate synthase component I"
## [6] "RPT3 26S proteasome regulatory subunit"
```

The labels to perform network propagation are MIPS categories (Mewes et al., 2000), which provide means to classify proteins regarding their function. These functions are coded as characters in the `yeast` object, in the node attribute `Class`

```
table_classes <- table(V(yeast)$Class, useNA = "always")
table_classes

##
##    A    B    C    D    E    F    G    M    O    P    R    T    U <NA>
##   51   98  122  238   95  171   96  278  171  248   45  240  483   39
```

The graph attribute `Classes` maps these abbreviations to the actual category:

```
head(yeast$Classes)

##   Category             Description
## 1        E       energy production
## 2        G    aminoacid metabolism
## 3        M        other metabolism
## 4        P             translation
## 5        T           transcription
## 6        B transcriptional control
##                                          Original.MIPS.category
## 1                                                        energy
## 2                                          aminoacid metabolism
## 3                              all remaining metabolism categories
## 4                                                protein synthesis
## 5 transcription, but without subcategory 'transcriptional control'
## 6                            subcategory 'transcriptional control'
```

Finally, the graph edges have a `Confidence` attribute that assesses the amount of evidence supporting the interaction. All the edges will be kept in this analysis, but different confidences can be weighted to favour diffusion in high confidence edges.

```
table(E(yeast)$Confidence)

##
##   high medium
##   2395   9298
```

More on the yeast object can be found through `?yeast`.

B.4.2 First analysis: protein ranking

In this first case, the diffusion scores will be applied to the prediction of a single protein function. Let's assume that 50% of the labelled proteins in the graph as `transport and sensing` (category `A`) are actually unlabelled. Now, using the labels of the known positive and negative examples for `transport and sensing`, can we correctly label the remaining 50%? First of all, the list of known and unknown positives is generated. The function `diffuse` uses (row)names in the input scores so that unlabelled nodes are accounted as so.

```
perc <- .5

# Transport and sensing is class A
nodes_A <- V(yeast)[Class %in% "A"]$name
nodes_unlabelled <- V(yeast)[Class %in% c(NA, "U")]$name
nodes_notA <- setdiff(V(yeast)$name, c(nodes_A, nodes_unlabelled))
```

```
# Known labels
known_A <- sample(nodes_A, perc*length(nodes_A))
known_notA <- sample(nodes_notA, perc*length(nodes_notA))
known <- c(known_A, known_notA)

# Unknown target nodes
target_A <- setdiff(nodes_A, known_A)
target_notA <- setdiff(nodes_notA, known_notA)
target <- c(target_A, target_notA)
target_id <- V(yeast)$name %in% target

# True scores
scores_true <- V(yeast)$Class %in% "A"
```

Now that the input is ready, the diffusion algorithm can be applied to rank all the proteins. As a first approach, the vanilla diffusion scores will be computed through the `raw` method and the default regularised Laplacian kernel, which is calculated on the fly.

```
# Vector of scores
scores_A <- setNames((known %in% known_A)*1, known)

# Diffusion
diff <- diffuStats::diffuse(
    yeast,
    scores = scores_A,
    method = "raw"
)

## Kernel not supplied.  Computing regularised Laplacian kernel ...
## Done
## All done
```

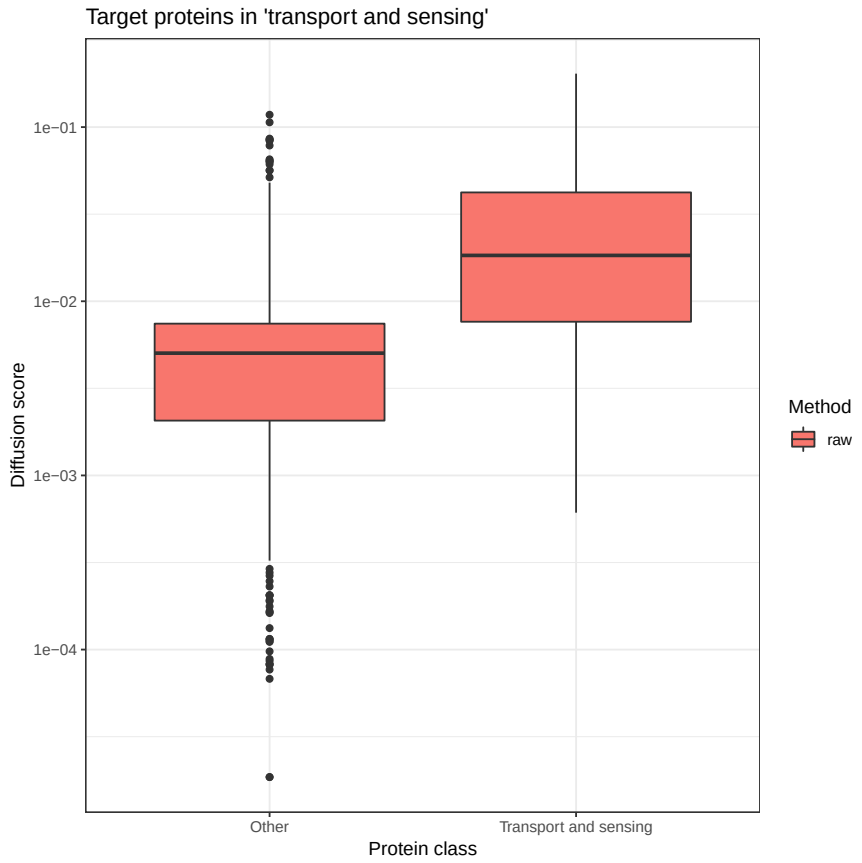
Diffusion scores are ready and in the same format they were introduced:

```
head(diff)

##     YLR197W     YOR039W     YDR473C     YOR332W     YER090W     YDR394W
## 0.004622066 0.003721601 0.003274780 0.085080780 0.009089522 0.006101765
```

Now, the scores obtained by the proteins actually belonging to `transport and sensing` can be compared to proteins with other labels.

```
# Compare scores
df_plot <- data.frame(
    Protein = V(yeast)$name,
    Class = ifelse(scores_true, "Transport and sensing", "Other"),
    DiffusionScore = diff,
    Target = target_id,
    Method = "raw",
```

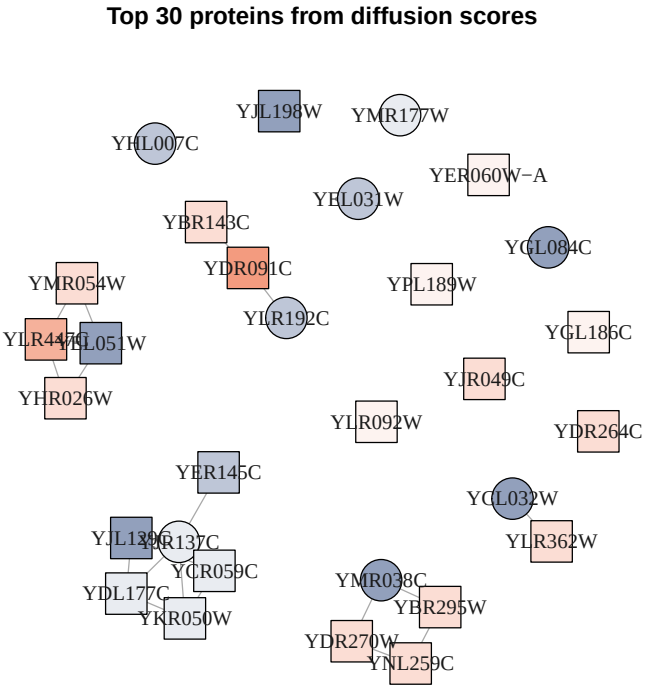
```
    stringsAsFactors = FALSE
)

ggplot(subset(df_plot, Target), aes(x = Class, y = DiffusionScore)) +
    geom_boxplot(aes(fill = Method)) +
    theme_bw() +
    scale_y_log10() +
    xlab("Protein class") +
    ylab("Diffusion score") +
    ggtitle("Target proteins in 'transport and sensing'")
```



The last plot justifies the usefulness of label propagation, as proteins in `transport and sensing` obtain higher diffusion scores than the rest. The network analysis can be deepened by examining, for instance, the subnetwork containing the proteins with the top 30 diffusion scores, highlighting with squares the ones that were positive labels in the input. Notice the small clusters of proteins:

```
# Top scores subnetwork
vertex_ids <- head(order(df_plot$DiffusionScore, decreasing = TRUE), 30)
yeast_top <- igraph::induced.subgraph(yeast, vertex_ids)

# Overlay desired properties
# use tkplot for interactive plotting
igraph::plot.igraph(
    yeast_top,
```

```
    vertex.color = diffuStats::scores2colours(
        df_plot$DiffusionScore[vertex_ids]),
    vertex.shape = diffuStats::scores2shapes(
        df_plot$Protein[vertex_ids] %in% known_A),
    vertex.label.color = "gray10",
    main = "Top 30 proteins from diffusion scores"
)
```



B.4.3 Comparing scores with single protein ranking

The proposed diffusion scores can be easily applied and compared. The regularised Laplacian kernel will be used to compute all the implemented scores for the target nodes in `transport and sensing`.

```
K_rl <- diffuStats::regularisedLaplacianKernel(yeast)
```

Functions `diffuse` and `perf` do accept, however, an `igraph` object as well, and compute the kernel automatically. For medium networks (10,000 nodes or more) the kernel computation can be computationally expensive in memory and time, so precomputing it avoids unnecessary recalculations.

The diffusion scores can be computed over a list of methods or sets of parameters. This can be achieved with instructions like `lapply`, but *diffuStats* contains a wrapper to facilitate this task. The function `diffuse_grid` takes the specified combinations of parameters -which can include the scoring

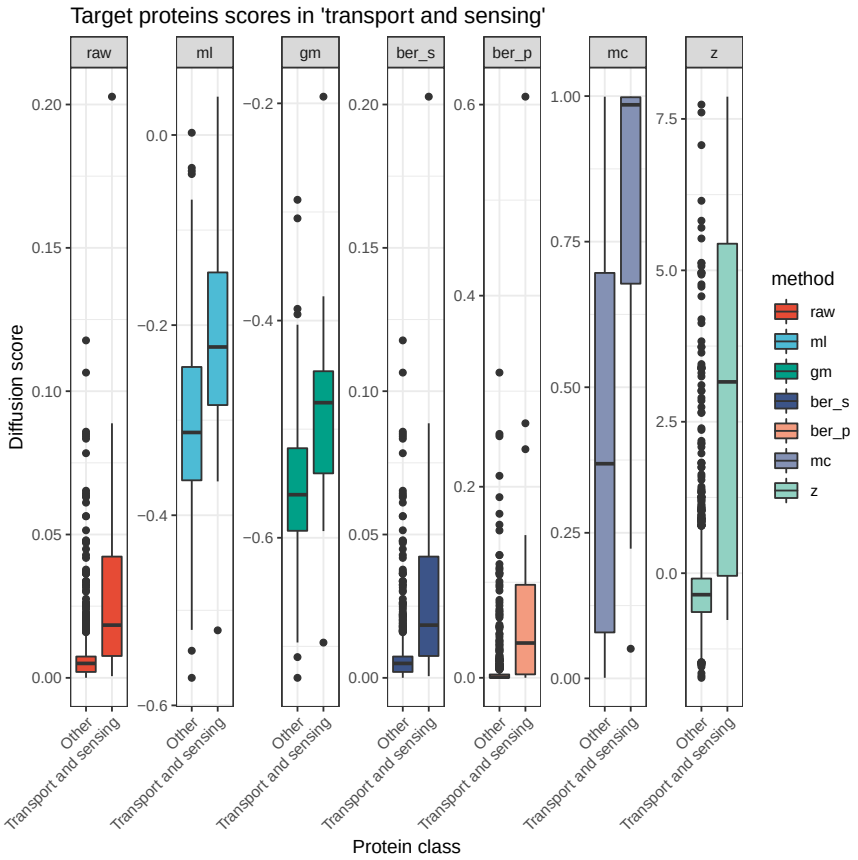
method as well- and computes the diffusion scores for each one. The results are concatenated in a data frame that can be easily plotted:

```
list_methods <- c("raw", "ml", "gm", "ber_s", "ber_p", "mc", "z")

df_diff <- diffuse_grid(
    K = K_rl,
    scores = scores_A,
    grid_param = expand.grid(method = list_methods),
    n.perm = 1000
)

## Using supplied kernel matrix...
## All done
## Using supplied kernel matrix...
## All done
## Using supplied kernel matrix...
## All done
## Using supplied kernel matrix...
## All done
## Using supplied kernel matrix...
## Using supplied kernel matrix...
## X1:  permuting scores...
## Permuting...
## X1:  computing heatRank...
## All done
## Using supplied kernel matrix...
## X1:  permuting scores...
## Permuting...
## X1:  computing heatRank...
## All done
## Using supplied kernel matrix...
## All done

df_diff$transport <- ifelse(
    df_diff$node_id %in% nodes_A,
    "Transport and sensing",
    "Other"
)
```

The results can be directly plotted:

```
df_plot <- subset(df_diff, node_id %in% target)
ggplot(df_plot, aes(x = transport, y = node_score)) +
    geom_boxplot(aes(fill = method)) +
    scale_fill_npg() +
    theme_bw() +
    theme(axis.text.x = element_text(
        angle = 45, vjust = 1, hjust = 1)) +
    facet_wrap( ~ method, nrow = 1, scales = "free") +
```

```
    xlab("Protein class") +
    ylab("Diffusion score") +
    ggtitle("Target proteins scores in 'transport and sensing'")
```



As expected, all the diffusion scores qualitatively show differences between positive and negative labels, but the quality of class separation will generally depend on the dataset and scoring method.

B.4.4 Benchmarking scores with multiple protein functions

The package *diffuStats* is able to perform several screenings at once. To show its usefulness, we will generalise the procedure in the last section but screening all the categories in the yeast graph.

First of all, the input data must meet an adequate format - a straightforward approach is to populate a matrix with the input labels (one category per column).

```
# All classes except NA and unlabelled
names_classes <- setdiff(names(table_classes), c("U", NA))

# matrix format
mat_classes <- sapply(
    names_classes,
    function(class) {
        V(yeast)$Class %in% class
```

```
    }
)*1
rownames(mat_classes) <- V(yeast)$name
colnames(mat_classes) <- names_classes
```

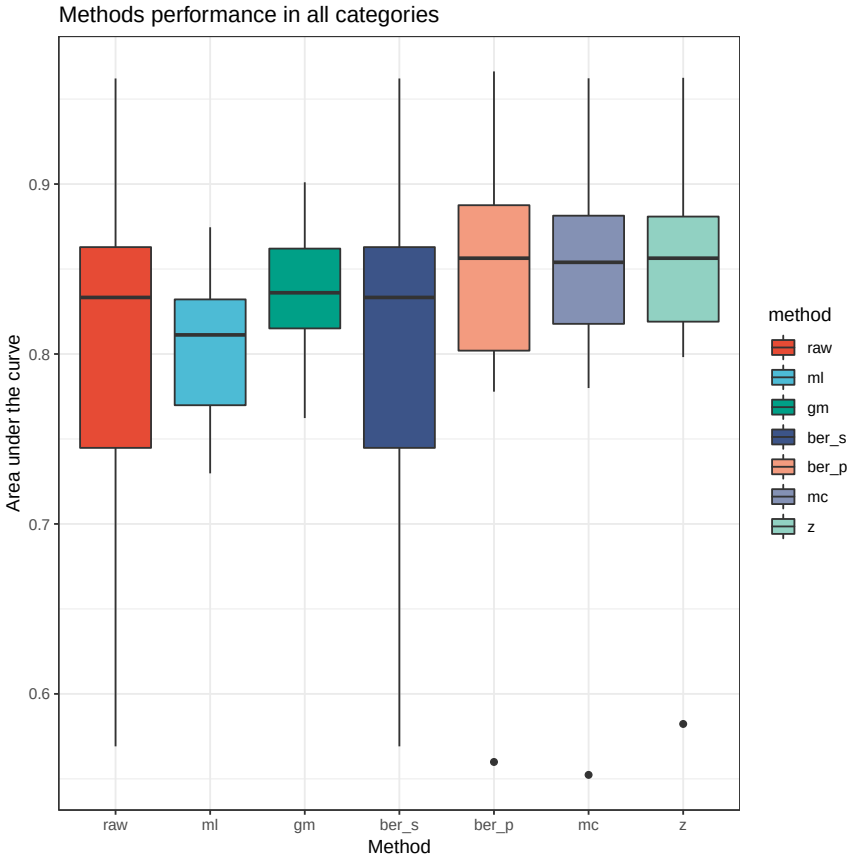
The former 50% known / 50% unknown approach will be kept with the same split, although not all the 12 categories will be totally balanced in the splits now. All the methods will be compared using the area under the ROC curve (AUROC) as a performance index.

Please note that *diffuStats* is equipped with basic performance measures for rankers: the AUROC, the area under the Precision-Recall curve, or AUPRC, and their partial versions. These are available through the helper function `metric_fun` and can be passed in list format to `perf`. These measures are based on the *precrec* R package - further detail can be found in the original manuscript (Saito and Rehmsmeier, 2017).

```
list_methods <- c("raw", "ml", "gm", "ber_s", "ber_p", "mc", "z")

df_methods <- perf(
    K = K_rl,
    scores = mat_classes[known, ],
    validation = mat_classes[target, ],
    grid_param = expand.grid(
        method = list_methods,
        stringsAsFactors = FALSE),
    n.perm = 1000
)

## Using supplied kernel matrix...
## All done
## Using supplied kernel matrix...
## All done
## Using supplied kernel matrix...
## All done
## Using supplied kernel matrix...
## All done
## Using supplied kernel matrix...
## Using supplied kernel matrix...
## X1:  permuting scores...
## Permuting...
## X1:  computing heatRank...
## All done
## Using supplied kernel matrix...
## X1:  permuting scores...
## Permuting...
## X1:  computing heatRank...
## All done
## Using supplied kernel matrix...
## All done
```

This allows plotting of the AUCs over the categories for each method in one step:

```
ggplot(df_methods, aes(x = method, y = auc)) +
    geom_boxplot(aes(fill = method)) +
    scale_fill_npg() +
    theme_bw() +
    xlab("Method") +
    ylab("Area under the curve") +
    ggtitle("Methods performance in all categories")
```



Scaling up the analysis can be useful for assessing how adequate a diffusion score is in the dataset of interest. These results suggest that, for the current `yeast` interactome and protein functions, the best priorisations are those obtained through a statistical normalisation, which might motivate its usage in other biological networks.

The user can also statistically compare the performance metrics through the function `perf_wilcox`. This generates a table with (i) the estimates on the differences on performance between the methods in rows and columns, with their confidence intervals and (ii) their associated p-value (Wilcoxon test). Positive and negative estimates respectively favour the method in the row and the column.

```
# Format the data
df_perf <- reshape2::acast(df_methods, Column~method, value.var = "auc")
# Compute the comparison matrix
```

```
df_test <- perf_wilcox(
    df_perf,
    digits_p = 1,
    adjust = function(p) p.adjust(p, method = "fdr"),
    scientific = FALSE)

## Warning in wilcox.test.default(x = perf_mat[, met1], y = perf_mat[,
met2], :  cannot compute exact p-value with zeroes
## Warning in wilcox.test.default(x = perf_mat[, met1], y = perf_mat[,
met2], :  cannot compute exact confidence interval with zeroes

knitr::kable(df_test, format = "latex")
```

	raw	ml	gm	ber_s	ber_p
raw	NA	0.018(-0.075,0.068)	-0.0084(-0.082,0.025)	NA	-0.023(-0.036,-0.0097)
ml	0.55	NA	-0.025(-0.047,-0.013)	-0.018(-0.068,0.075)	-0.042(-0.095,0.047)
gm	0.71	0.01	NA	0.0084(-0.025,0.082)	-0.02(-0.049,0.054)
ber_s	NA	0.55	0.71	NA	-0.023(-0.036,-0.0097)
ber_p	0.03	0.46	0.54	0.03	NA
mc	0.12	0.35	0.55	0.12	0.73
z	0.03	0.32	0.46	0.03	0.46

B.5 CONCLUSIONS

The *diffuStats* package is a new computational tool to compute and compare single-network diffusion scores that are object of active research in several bioinformatics areas. It is an effort to gather a collection of settings in the diffusion process like the graph kernel, the label codification and the choice of a statistical normalisation. The *diffuStats* package will help the end user in choosing and computing the best performing diffusion scores in the application of interest.

B.6 FUNDING

This work was supported by the Spanish Ministry of Economy and Competitiveness (MINECO) [TEC2014-60337-R to A.P.] and the National Institutes of Health (NIH) [R01GM104400 to W.T.]. AP. and S.P. thank for funding the Spanish Biomedical Research Centre in Diabetes and Associated Metabolic Disorders (CIBERDEM) and the Networking Biomedical Research Centre in the subject area of Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), both initiatives of Instituto de Investigación Carlos III (ISCIII). SP. thanks the AGAUR FI-scholarship programme.

B.7 SESSION INFO

Here is the output of `sessionInfo()` on the system that compiled this vignette:

- R version 3.6.2 (2019-12-12), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=es_ES.UTF-8`, `LC_COLLATE=en_US.UTF-8`, `LC_MONETARY=es_ES.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=es_ES.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=es_ES.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 16.04.6 LTS`
- Matrix products: default
- BLAS: `/usr/lib/atlas-base/atlas/libblas.so.3.0`
- LAPACK: `/usr/lib/atlas-base/atlas/liblapack.so.3.0`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: diffuStats 1.4.0, ggplot2 3.1.1, ggsci 2.9, igraph 1.2.4.1, igraphdata 1.0.1, knitr 1.22
- Loaded via a namespace (and not attached): assertthat 0.2.1, backports 1.1.4, BiocManager 1.30.4, BiocStyle 2.12.0, colorspace 1.4-1, compiler 3.6.2, crayon 1.3.4, data.table 1.12.2, digest 0.6.18, dplyr 0.8.3, evaluate 0.13, expm 0.999-4, glue 1.3.1, grid 3.6.2, gtable 0.3.0, highr 0.8, htmltools 0.3.6, labeling 0.3, lattice 0.20-38, lazyeval 0.2.2, magrittr 1.5, MASS 7.3-51.5, Matrix 1.2-18, munsell 0.5.0, pillar 1.4.0, pkgconfig 2.0.2, plyr 1.8.4, precrec 0.10.1, purrr 0.3.2, R6 2.4.0, Rcpp 1.0.1, RcppArmadillo 0.9.800.3.0, RcppParallel 4.4.4, reshape2 1.4.3, rlang 0.4.0, rmarkdown 1.12, scales 1.0.0, stringi 1.4.3, stringr 1.4.0, tcltk 3.6.2, tibble 2.1.1, tidyselect 0.2.5, tools 3.6.2, vctrs 0.2.0, withr 2.1.2, xfun 0.6, yaml 2.2.0, zeallot 0.1.0

REFERENCES

Allaire, JJ, Romain Francois, Kevin Ushey, Gregory Vandenbrouck, Marcus Geelnard, and Intel

2016 *RcppParallel: Parallel Programming Tools for 'Rcpp'*, R package version 4.3.20, https://CRAN.R-project.org/package=RcppParallel.

Bersanelli, Matteo, Ettore Mosca, Daniel Remondini, Gastone Castellani, and Luciano Milanesi

2016 "Network diffusion-based analysis of high-throughput data for the detection of differentially enriched modules." *Scientific Reports*, 6, August, p. 34841, ISSN: 2045-2322, DOI: 10.1038/srep34841.

Csardi, Gabor

2015 "igraphdata: A Collection of Network Data Sets for the 'igraph' Package", R package version 1.0.1, https://CRAN.R-project.org/package=igraphdata.

Csardi, Gabor and Tamas Nepusz

2006 "The igraph software package for complex network research", *InterJournal*, Complex Systems, p. 1695, http://igraph.org.

Eddelbuettel, Dirk

2013 *Seamless R and C++ integration with Rcpp*, Springer.

Eddelbuettel, Dirk and Conrad Sanderson

2014 "RcppArmadillo: Accelerating R with high-performance C++ linear algebra", *Computational Statistics and Data Analysis*, 71 (Mar. 2014), pp. 1054-1063, DOI: 10.1016/j.csda.2013.02.005.

Harchaoui, Zaid, Francis Bach, Olivier Cappe, and Eric Moulines

2013 "Kernel-based methods for hypothesis testing: A unified view", *IEEE Signal Processing Magazine*, 30, 4, pp. 87-97.

Lee, Insuk, U Martin Blom, Peggy I Wang, Jung Eun Shim, and Edward M Marcotte

2011 "Prioritizing candidate disease genes by network-based boosting of genome-wide association data", *Genome Research*, 21, 7, pp. 1109-1121, DOI: 10.1101/gr.118992.110.

Mewes, Hans-Werner, Dmitrij Frishman, Christian Gruber, Birgitta Geier, Dirk Haase, Andreas Kaps, Kai Lemcke, Gertrud Mannhaupt, Friedhelm Pfeiffer, C Schüller, et al.

2000 "MIPS: a database for genomes and protein sequences", *Nucleic acids research*, 28, 1, pp. 37-40.

Mostafavi, Sara, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris

2008 "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function." *Genome Biology*, 9 Suppl 1, S4, ISSN: 1474-760X, DOI: 10.1186/gb-2008-9-s1-s4.

North, Bernard V, David Curtis, and Pak C Sham

2002 "A note on the calculation of empirical P values from Monte Carlo procedures", *The American Journal of Human Genetics*, 71, 2, pp. 439-441.

Oliver, Stephen

2000 "Guilt-by-association goes global", *Nature*, 403, February, pp. 601-603.

Paull, Evan O., Daniel E. Carlin, Mario Niepel, Peter K. Sorger, David Haussler, and Joshua M. Stuart

2013 "Discovering causal pathways linking genomic events to transcriptional states using Tied Diffusion Through Interacting Events (TieDIE)", *Bioinformatics*, 29, 21, pp. 2757-2764, ISSN: 13674803, DOI: 10.1093/bioinformatics/btt471.

R Core Team

2017 *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Saito, Takaya and Marc Rehmsmeier

2017 "Precrec: fast and accurate precision–recall and ROC curve calculations in R", *Bioinformatics*, 33, 1, pp. 145-147.

Smola, Alexander J and Risi Kondor

2003 "Kernels and regularization on graphs", pp. 144-158, DOI: 10.1007/978-3-540-45167-9_12.

Suthram, Silpa, Andreas Beyer, Richard M Karp, Yonina Eldar, and Trey Ideker

2008 "eQED: an efficient method for interpreting eQTL associations using protein networks", *Molecular systems biology*, 4, 1, p. 162.

Tsuda, Koji, HyunJung J. Shin, and Bernhard Schölkopf

2005 "Fast protein classification with multiple networks", *Bioinformatics*, 21, SUPPL. 2, pp. 59-65, ISSN: 13674803, DOI: 10.1093/bioinformatics/bti1110.

Valentini, Giorgio, Alberto Paccanaro, Horacio Caniza, Alfonso E. Romero, and Matteo Re

2014 "An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods", *Artificial Intelligence in Medicine*, 61, 2, pp. 63-78, ISSN: 18732860, DOI: 10.1016/j.artmed.2014.03.003.

Vandin, Fabio, Eli Upfal, and Benjamin J. Raphael

2010 "Algorithms for detecting significantly mutated pathways in cancer", *Lect. Notes Comput. Sci.*, 6044 LNBI, 3, pp. 506-521, ISSN: 03029743, DOI: 10.1007/978-3-642-12683-3_33.

Wickham, Hadley

2009 *ggplot2: Elegant Graphics for Data Analysis*, Springer-Verlag New York, ISBN: 978-0-387-98140-6, http://ggplot2.org.

Yen, Luh, Francois Fouss, and Christine Decaestecker

2007 "Graph nodes clustering based on the commute-time kernel", *Advances in Knowledge Discovery and Data Mining*, pp. 1037-1045, ISSN: 03029743, DOI: 10.1007/978-3-540-71701-0_117.

Zoidi, Olga, Eftychia Fotiadou, Nikos Nikolaidis, and Ioannis Pitas

2015 "Graph-Based Label Propagation in Digital Media: A Review", *ACM Computing Surveys*, 47, 3, 48:1-48:35, ISSN: 0360-0300, DOI: 10.1145/2700381.

C METABOLOMICS ENRICHMENT

C.1 APPENDIX S1 – GRAPH STRUCTURE AND CURATION

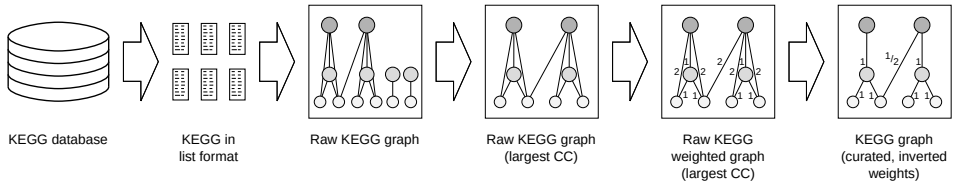


Figure 73: Procedure to obtain the KEGG graph. The KEGG database is read as a collection of lists that contain the annotations. The raw KEGG graph is built through these annotations, where the vertices are KEGG entries from categories in Fig. 74a. We only work with the largest CC of the raw KEGG graph, to which weights are assigned, enabling the curation step that gives place to the KEGG graph. Note that the weights in the definitive KEGG graph are the inverse of the former dissimilarity weights, to be consistent with the diffusive methods.

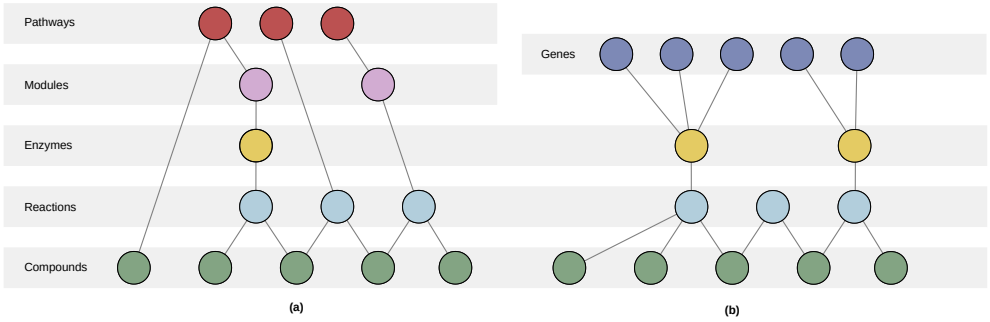


Figure 74: **(a)** Structure of our KEGG graph. Each entry belongs to a level, ranging from 1 (pathways) to 5 (compounds). **(b)** MetScape concept of compound-reaction-enzyme-gene network. Their construction is similar in the three lowest levels, while in the upper level they include KEGG genes.

The first step to depict current knowledge is to build a graph object from KEGG that enables data enrichment (Fig. 73). The KEGG graph contains various categories (Fig. 74a) and keeps similarities with the networks built through MetScape (Karnovsky et al., 2012), see (Fig. 74b), although our structure is conceived to include biological pathways and modules to obtain

This appendix reproduces the supplementary data (Appendices S1 to S5) of: Picart-Armada, Sergio, Francesc Fernández-Albert, Maria Vinaixa, Miguel A. Rodríguez, Suvi Aivio, Travis H. Stracker, Oscar Yanes, and Alexandre Perera-Lluna. "Null diffusion-based enrichment for metabolomics data". *PloS one 12, no. 12 (2017)*.

a pathway enrichment procedure. The lists relating these categories can be retrieved through the KEGGREST package (Tenenbaum, 2016).

In order to restrict our KEGG graph to nodes related to Homo sapiens, only human pathways and modules were considered. Pathway hsa01100 (Metabolic pathways) was discarded for being too general. Enzymes were included only if at least one human gene was related to them, as enzyme-module and enzyme-pathway connections were inferred through genes. Reactions and compounds were drawn only if they belonged to a human pathway or module. Finally, for completeness purposes, any reactant or product of the already kept reactions was added. These steps resulted in the raw KEGG graph (Fig. 73).

We have conceived a curation algorithm that assigns edge weights and removes redundant edges from the graph. The requirements for the graph that enable the curation and the diffusion processes are: **(i)** the chosen categories allow a hierarchical arrangement, **(ii)** none of the links relates nodes belonging to the same category and **(iii)** affected nodes lie only on the bottom level (lowest category).

In the first place, our five KEGG categories conform a hierarchy, from top to bottom: biological pathway, module, enzyme, reaction and compound. This choice mimics the transition from the smaller parts (compounds) to the larger units (pathways) and facilitates the tracking of the biological perturbation, suggesting paths and entities by which the affected compounds translate into altered pathways. In the second place, KEGG does not contain any link between entries within the same category.

After building the unweighted graph from KEGG annotations and working with its largest CC, we begin the curation by proposing edge weights (Fig. 73) that reflect the specificity of the link between the two entries i and j within the hierarchy, as described in equation (48):

$$w_{ij} = w_{ji} = \begin{cases} |l_i - l_j| \text{ if } i \text{ and } j \text{ are linked through an edge} \\ \infty \text{ otherwise (equivalently, not adjacent vertices)} \end{cases} \tag{48}$$

In equation (48), l_i stands for the level of node i; note that the specified requirements ensure that l_i is defined for each node (hierarchical structure) and that $w_{ij} \neq 0$ (no edges between nodes within the same level). For instance, an edge between a compound and a reaction weights 1, meaning that it describes a close relationship in metabolic terms. Instead, if the link involves a compound and a pathway this weight becomes 4, meaning the lack of known intermediate implications involving reactions, enzymes and modules.

The next step in the curation process discards any edge that can be explained using more informative edges (Fig. 73), therefore avoiding any data loss. Specifically, any triangle in the graph is removed by dropping the edge with the largest weight. For example, a link between a compound and a pathway will drop if there are known intermediate levels that explain this connection.

The algorithm to achieve this from the original weighted graph $G = (V, E)$ of order n and size m is the following. Note that the algorithm is still valid in the presence of multi-edges (edges that are incident to the same pair of

vertices), but as a proof of concept we assume that the graph does not contain them.

1. Sort the edges in E with increasing w_{ij}: $L = (e_{(1)}, \ldots, e_{(m)})$. The criterion to break ties is irrelevant.
2. Initialise a graph $G_{new} = (V_{new}, E_{new}) = (V, \varnothing)$ with the same node set as the original graph, but with no edges
3. For each edge e_{ij} in L, which links vertices i and j in G, add e_{ij} to G_{new} only if $d_{G_{new}}(i, j) > w_{ij}$.
4. Return G_{new}

In other words, only edges that contribute with new data in the biological graph are added. Distances must use the weights provided by w_{ij}. If an edge e_{ij} is discarded, that means that there is already a connection between i and j with the same level of detail, and because of the construction of G_{new} this connection is through two or more edges, all of them having strictly less weight than e_{ij}. Hence, e_{ij} is redundant in that situation. A small example is shown (Fig. 75) to justify the curation process.

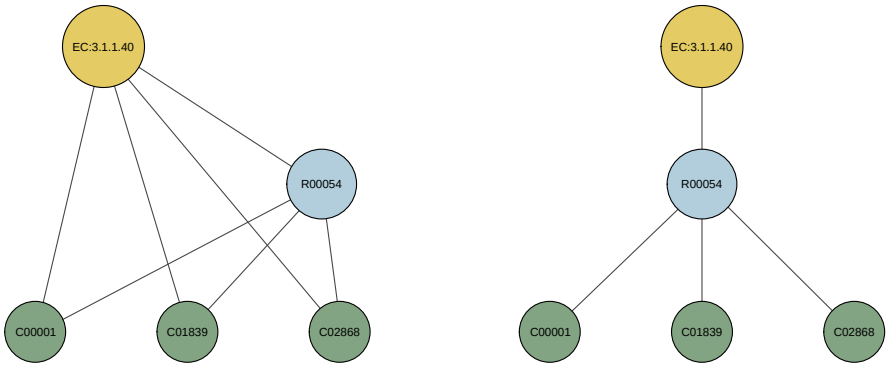


Figure 75: Example of the curation process applied to a small subgraph from KEGG graph. The graph in the left contains all the original edges. Likewise, the graph in the right contains a neater explanation of the biology: three compounds that participate in a reaction, catalysed by one enzyme. We capture the essence of the data through 4 edges instead of 7, while easing the posterior visual interpretation.

After the curation process, we obtain the KEGG graph (Fig. 73). The final weights are inverted to be consistent with the graph Laplacian matrix and the diffusive methods. KEGG graph contains a total of 10,183 nodes and 31,539 edges. The nodes are stratified in 288 pathways, 178 modules, 1,149 enzymes, 4,699 reactions and 3,869 compounds. The degree distribution (Fig. 76) follows a scale-free model.

The third requisite about the graph (the measured nodes should lie on the lower level) ensures that the boundary setup is meaningful and it eases the traceability of the biological perturbation, which follows a bottom-up tendency. Introducing flow on intermediate levels can nullify the structure inheritance from the whole graph when selecting the subnetwork, thus undermining the quality of the resulting biological interpretation.

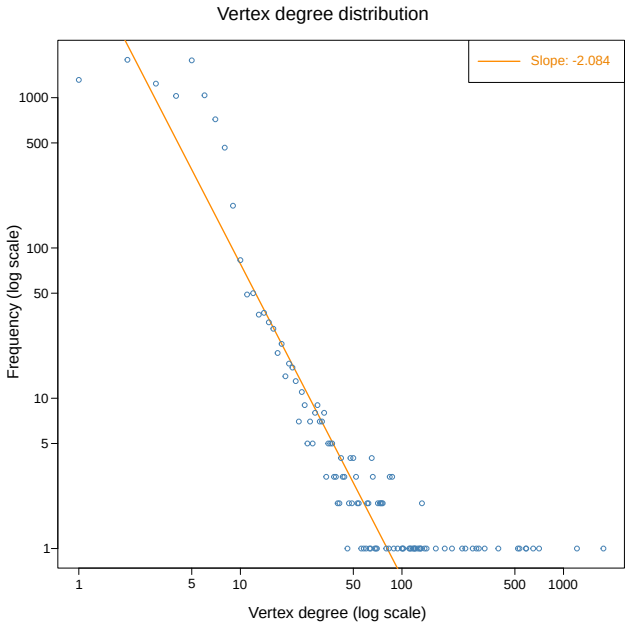


Figure 76: Our KEGG graph degree distribution follows the scale-free network pattern $P(k) \sim k^{-\gamma}$, with $\gamma = 2.084 \in [2, 3]$. The heavy tail of this distribution confirms the existence of hubs, which are nodes with an extremely high degree and a major role in the biology.

The application of our diffusion processes with their null models is aimed at reporting a relevant subgraph of our KEGG graph. This subgraph can be examined through the order (amount of nodes) and amount of connected components (CC), an indicator of its structure and quality. A large CC is likely to give a global explanation in terms of all the levels in the graph while several small CCs will only highlight very specific relationships between small sets of nodes.

C.2 APPENDIX S2 – HEAT DIFFUSION PROCESS

The heat diffusion process is a model to quantify the propagation of flow in a network; this flow represents a biological perturbation when the experimental conditions change. However, note that this design is neither a functional model of biology nor a simulation of heat diffusion on biological molecules.

Using the explicit method for the finite difference formulation of the heat diffusion problem (Eq. 49), we can relate the temperatures between contiguous time instants in a meshed object containing n nodes. A graph can be naturally regarded as a meshed object, thus allowing the heat diffusion on our KEGG graph. The formulation (Bonals, 2005) is:

$$T^{k+1} = T^k + DTC \cdot [KI \cdot T^k + KC \cdot TC + G] \tag{49}$$

where

$$T^k = \begin{bmatrix} T_1^k \\ T_2^k \\ \vdots \\ T_n^k \end{bmatrix} {}^\circ C \tag{50a}$$

are the temperatures of the n nodes in the graph at the k-th instant, T_i^k. Also,

$$DTC = \begin{bmatrix} \frac{\Delta t}{C_1^k} & 0 & \dots & 0 \\ 0 & \frac{\Delta t}{C_2^k} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\Delta t}{C_n^k} \end{bmatrix} \frac{{}^\circ C}{W} \tag{50b}$$

is the diagonal matrix containing the quotient between the time step Δt, in seconds, and the heat capacity of node n at the k-th instant, C_n^k, in $\frac{J}{{}^\circ C}$. Following,

$$KI = \begin{bmatrix} -\sum_j K_{1j}^k & K_{12}^k & \dots & K_{1n}^k \\ K_{21}^k & -\sum_j K_{2j}^k & \dots & K_{2n}^k \\ \vdots & \vdots & \ddots & \vdots \\ K_{n1}^k & K_{n2}^k & \dots & -\sum_j K_{nj}^k \end{bmatrix} \frac{W}{{}^\circ C} \tag{50c}$$

contains the heat conductance between nodes v_i and v_j in the k-th instant, K_{ij}^k. The sums in the diagonal also account for the conductance to the boundary nodes if present. Next,

$$KC = \begin{bmatrix} K_{1,n+1}^k & K_{1,n+2}^k & \dots & K_{1,n+c}^k \\ K_{2,n+1}^k & K_{2,n+2}^k & \dots & K_{2,n+c}^k \\ \vdots & \vdots & \ddots & \vdots \\ K_{n,n+1}^k & K_{n,n+2}^k & \dots & K_{n,n+c}^k \end{bmatrix} \frac{W}{{}^\circ C} \tag{50d}$$

is the matrix containing the conductances from the node v_i to the l-th boundary node (which does not belong to the graph) in the k-th instant, $K_{i,n+l}^k$. As for these boundary nodes,

$$TC = \begin{bmatrix} T_{n+1}^k \\ T_{n+2}^k \\ \vdots \\ T_{n+c}^k \end{bmatrix} {}^\circ C \tag{50e}$$

is the vector that contains temperatures of the boundary node l in the k-th instant, T_{n+l}^k (note that there are c boundary nodes in total and that they are not in V). Finally,

$$G = \begin{bmatrix} G_1^k \\ G_2^k \\ \vdots \\ G_n^k \end{bmatrix} W \tag{50f}$$

contains the inner heat generation for node v_i in the k-th instant, G_i^k.

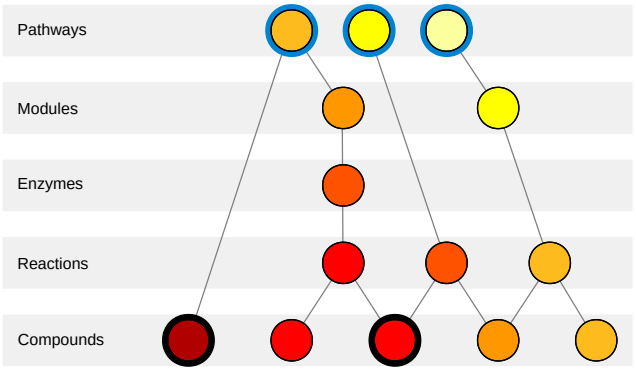


Figure 77: Nodes arrangement for heat diffusion. In this tiny example, the levels represent the hierarchy from pathways (top) to compounds (bottom). The affected compounds are highlighted with a black ring. Affected compounds are forced to generate unitary flow. To reach a stationary state, these two flow units must evacuate through pathways, located at the top level. Every pathway is highlighted with a blue ring, representing its connection to a cool boundary node at 0°C. In the stationary state, depicted through heat colours proportional to the final temperature, the warmest pathways will hold greatest heat flow, suggesting a relevant role in the experiment.

The finite difference expression (Eq. 49) takes a substantially simpler form when applied to our node arrangement (Fig. 77). First, imposing the stationary state $T^n = T^{n+1} = T$ and constant parameters for every time step:

$$T = -KI^{-1} \cdot [KC \cdot TC + G] \tag{51}$$

As shown in the proposed configuration (Fig. 77), the boundary nodes are at 0°C, therefore the equation simplifies into Eq. (52).

$$T = -KI^{-1} \cdot G = R_{HD} \cdot G \tag{52}$$

where $R_{HD} = -KI^{-1}$ is the linear mapping of the heat diffusion process. This is the expression shown in the main body; an equivalent formulation can be found at HotNet (Vandin et al., 2011). The terms in the conductance matrix KI are given by the inverse of the weights in the curation process presented in Appendix S1, whereas conductances to boundary nodes are unitary. Besides allowing the calculation of temperatures, Eq. 52 also describes the diffusion process. For example, the null diffusion correlation matrix between biological entities in KEGG, described in Appendix S4, can give insights about the nature of the network.

Further analyses can be achieved through the conductance matrix of the graph (Bapat, 2004). This perspective is usually regarded as the electrical problem of finding the equivalent resistance between any couple of nodes. The resistance distance is a metric that takes into account all the possible paths from one vertex to the other, and not only its shortest path, thus effectively including the graph topology.

C.3 APPENDIX S3 – PAGERANK

The PageRank algorithm (Page et al., 1999) scores every node in a graph using a web surfer model. The graph is typically directed because so are the hyperlinks between websites. Applying PageRank to an undirected graph is even more similar to the heat diffusion described in Appendix S3.

The web surfer model mimics the behaviour of real internet users. The surfer starts a random walk at a randomly chosen website, according to a prior probabilities vector p. In each step of the random walk, he or she decides whether to continue with the current random walk (probability d) or start a new one (probability $1 - d$). If the random walk is resumed, the probability of choosing an edge is proportional to its weight. Finally, the PageRank scores are the stationary probability distribution over the graph nodes for this surfer.

Despite the different formulation of the PageRank problem, the final calculation of the PageRank scores is similar to the stationary state of our heat diffusion process. The arrangement of the nodes is identical (Fig. 78), but the PageRank graph is directed upwards.

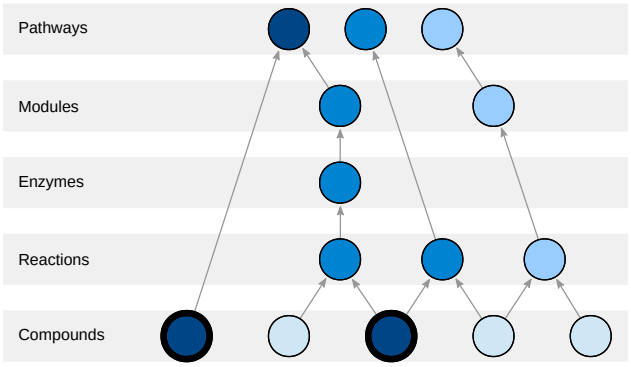


Figure 78: Nodes arrangement for PageRank. Affected compounds are the start of random walks, uniformly distributed among them. PageRank scores, represented by the intensity of the blue colour, will attain larger values in the nodes that are frequently reached through the random walks in a stationary state. Random walks resumed in dead ends start in a uniformly chosen node from the graph.

The mathematical expression to obtain the scores (Equation 54) can be derived by imposing a stationary probability in the random walk process, whose website transitions are governed by the matrix M:

$$PR = d \cdot M \cdot PR + (1 - d) \cdot p \tag{53}$$

obtaining

$$PR = R_{PR} \cdot p \tag{54}$$

p is the probability distribution for the source of new random walks and R_{PR} is the matrix:

$$R_{PR} = (1 - d) \cdot (Id - d \cdot M)^{-1} \tag{55}$$

where Id is the identity matrix, d is the damping factor and M is a matrix obtained from the weighted adjacency $n \times n$ matrix A from the directed KEGG graph (edges pointing upwards):

$$M = f(t(A)) \tag{56}$$

In this expression, t is the matrix transpose operator and f is the function that normalises each column to sum 1, except when it contains n zeroes; in the latter case it returns a column with n elements equal to $\frac{1}{n}$. We apply the function f with that particularity to be coherent with the R package igraph (Csardi and Nepusz, 2006), which considers that terminal nodes resume the random walk uniformly distributed in all the vertices.

In Equation 54, the calculation of the PageRank scores uses the binary vector of affected compounds normalised by the amount of affected compounds (Fig. 78):

$$p = \frac{G}{\sum G_i} \tag{57}$$

The similarity between the final expression for heat diffusion (see Appendix S2) and PageRank (Equation 54) is remarkable, given the common random walk background for these two methods. The differences between them include the forced upwards directionality of PageRank and the damping factor concept, which allows leaps in the diffusion. The rescaling of the p vector does not affect the null model, as the rescaling factor remains constant in the random trials.

All the PageRank scores in our approach have been computed using the standard $d = 0.85$ established in the original publication (Page et al., 1999), a range of damping factors has been swept, going from 0.1 (very frequent restarts) to 0.95 (almost no restarts), but results are consistent as a result of the application of our null model, described in Appendix S4.

C.4 APPENDIX S4 – NULL MODELS

We retrieve the formulation from Appendix S2 to compute the final temperatures in heat diffusion. The same procedure applies to the PageRank approach in Appendix S3, given the similarity between them, but as a proof of concept it will be developed for heat diffusion only.

$$T = R_{HD} \cdot G$$

By abuse of notation, R_{HD} will contain the columns of the original R_{HD} corresponding only to compounds, thus being a rectangular matrix from now on. Likewise, to have a well-defined matrix-vector product, G will only refer to compounds, as they are the only entities that can introduce heat. The vector G contains exactly n_{in} ones, corresponding to the affected compounds, and $n_{comp} - n_{in}$ zeroes, where n_{comp} is the amount of compounds in the graph.

When focusing on a node i, we want to assess whether its temperature T_i would be expected from a random selection of affected compounds or,

on the contrary, the affected compounds are more related to node i than expected. To that end, we define the null distribution of temperatures:

$$T_{null} = R_{HD} \cdot X \tag{58}$$

where X is the random variable obtained by permuting G. If we define $p = \frac{n_{in}}{n_{comp}}$, then every X_i is a Bernoulli trial with success probability p. X_i and X_j, for $i \neq j$, are slightly anticorrelated due to the permutation approach.

Exact statistical moments of T_{null} can be computed:

$$\mathbb{E}(T_{null}) = R_{HD} \cdot \mathbb{E}(X) \tag{59}$$

being

$$\mathbb{E}(X) = p \cdot \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{60}$$

and the same for the covariance matrix

$$\Sigma(T_{null}) = R_{HD} \cdot \Sigma(X) \cdot R_{HD}^{T} \tag{61}$$

where

$$\Sigma(X) = p(1-p) \cdot \begin{bmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & 1 \end{bmatrix} \tag{62}$$

being $\rho = -\frac{1}{n_{comp}-1}$. In terms of the elements r_{ij} in matrix R_{HD}, we can write

$$\mu_i = \frac{n_{in}}{n_{comp}} \left[\sum_{j=1}^{n_{comp}} r_{ij} \right] \tag{63}$$

$$\sigma_i^2 = \frac{n_{in}(n_{comp}-n_{in})}{n_{comp}(n_{comp}-1)} \left[\left(\sum_{j=1}^{n_{comp}} r_{ij}^2 \right) - \frac{1}{n_{comp}} \left(\sum_{j=1}^{n_{comp}} r_{ij} \right)^2 \right] \tag{64}$$

In fact, the correlation matrix of the null temperatures gives insights about the combination of the network structure and the null model. Focusing on pathways: if two pathways are strongly correlated, it suggests that both attain high or low temperatures with similar inputs. Thus, this couple of pathways are prone to overlap or to be nearby in the metabolism. Conversely, a strong anticorrelation suggests that warming up a pathway conditions the second pathway to become colder, suggesting that these pathways are dissimilar. (Fig. 79) depicts the correlations matrix for the pathways, where rows and columns have been reordered to illustrate clusters of KEGG pathways. Furthermore, the structure seems to be somehow related to the underlying biology: one of the clusters corresponds to the bulk of human

metabolic pathways, whereas the genetic information processing pathways also appear highly correlated. Besides these examples, pathway types do not appear totally shuffled, but as small blocks of pathways sharing a biological role.

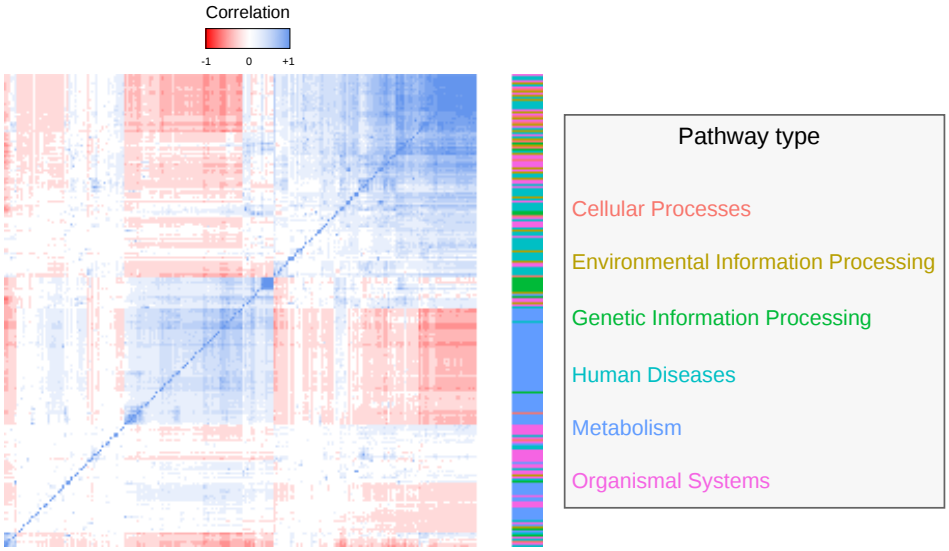


Figure 79: Correlation matrix between pathways for the heat diffusion process in the KEGG graph (identifiers omitted for clarity). Blue correlations are closer to 1, while red tend to -1. For the calculation of these correlations, a count of 33 compounds was assumed to perform the null model, like in the experimental data. In addition, the biological role of the pathways annotated in KEGG BRITE (Kanehisa et al., 2008) has been drawn in the rightmost bar.

(Eqs. 63, 64) provide a first approximation **(1)** to evaluate how high a temperature is. If T_i is remarkably greater than $\mu_i = \mathbb{E}(T_{null})_i$ in terms of its standard deviation, $\sigma_i = \sqrt{\Sigma(T_{null})_{ii}}$, then node i should be reported. The normalised score is

$$z_i = \frac{T_i - \mu_i}{\sigma_i} \tag{65}$$

Another approach **(2)** to evaluate the relevance of node i is to perform the permutation analysis through Monte Carlo trials. In that case, the random vector X is drawn n_{perm} times and, for node i, the p-value is approximated as $p_i = \frac{r_i+1}{n_{perm}+1}$, where r_i is the number of trials where $T_{null_i} \geqslant T_i$, see (North et al., 2002) for further details on this estimator. An ensemble solution can be obtained by repeating the procedure n_{vote} times and evaluating each node by majority vote, specifically including it only if it is reported at least $\lfloor \frac{n_{vote}}{2} \rfloor + 1$ times, also allowing a fuzzy representation of the consensus solution. This ensemble approach reduces the variability in the reported solution and also provides confidence measures for each included node. The input can also be subsampled in this approach, although this option has not been explored yet.

C.5 APPENDIX S5 – DETAILS ON REPORTED SOLUTIONS

The solutions reported in the main body encompass two different scoring functions (heat diffusion and PageRank) and two statistical approaches (z-scores and simulation). Monte Carlo permutations involve consensus solutions among the simulated runs to reduce the variability and enhance robustness and consistency between runs.

C.5.1 Solution stratification

(Fig. 80) depicts the stratification of all the reported graphs. Solutions tend to keep the same proportions as the original graph, allowing the discovery of relevant nodes in all the categories. This is not only a sign of agreement across the different solutions, but also a necessary behaviour to discover putative nodes in all the categories. The proportion of reported compounds seems lower than the one in the KEGG graph, probably due to the application of the null model, which tends to favour the metabolites in the input and penalise the rest.

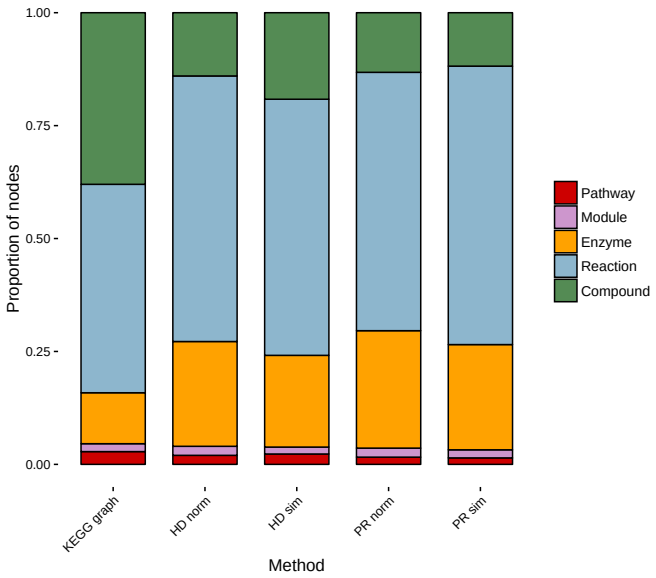


Figure 80: Subgraph stratification by method. Notice the tendency to keep the same distribution as the whole KEGG graph. Compared to the KEGG graph, the proportion of compounds decreases in all the solutions due to the inclination to recover the ones in the input and exclude the rest.

C.5.2 Connected component evolution

The choice of the number of desired nodes k affects the number and size of the reported connected components (CC). In general terms, the number of reported CCs seems to lower as the number of reported nodes grows (Fig. 81), as different CCs that contain seed nodes tend to merge. The number of nodes in the largest CC grows monotonically as k increases and it captures the majority of the nodes in the subgraphs (Fig. 82).

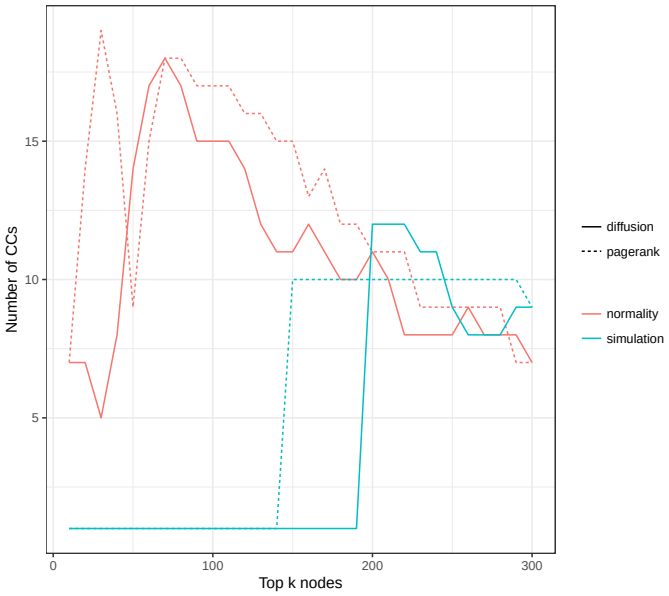


Figure 81: Number of reported CCs for varying k. The discrete nature of the majority vote approach in simulation trials leads to ties, which can be spotted as horizontal lines in the figure - more than a hundred nodes are tied with the best rank in both cases. In general, the number of CCs decreases as the reported subgraphs grow.

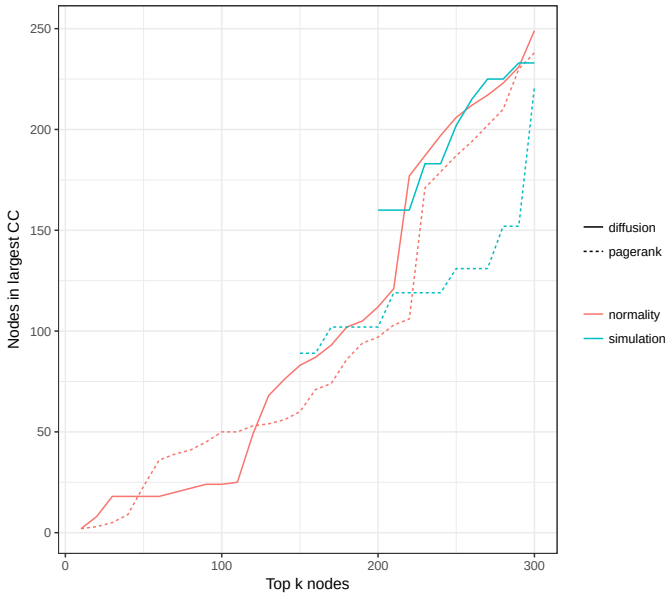


Figure 82: Number of nodes in the largest CC reported. If more nodes are reported, the largest CC grows accordingly. For $k = 300$ around 225-250 nodes are in the largest CC in every approach, meaning that approximately 75-83% nodes lie in it. As more than a hundred nodes are tied with maximum rank in the simulated version, lines start at these respective points.

C.5.3 Computational cost

In sight of further applications of these methodologies, we have performed a benchmark of several implementations of the Monte Carlo approach. For heat diffusion, the temperature calculation (Appendix S2) is achieved through

$$T = -KI^{-1} \cdot G = R_{HD} \cdot G \tag{66}$$

where G has n_{in} ones and $n_{comp} - n_{in}$ zeroes.

We define two strategies to permute the input G: **(a)** draw n_{in} elements without replacement from the set $[1, n_{comp}]$, and **(b)** shuffle the whole vector G. Furthermore, two possible calculations for T are: **(1)** explicitly compute R_{HD} and sum the columns indexed by the n_{in} ones, or **(2)** solve the linear system $T = -KI^{-1} \cdot G$

These strategies have been essayed with growing graph order, from 1,000 to 10,000 nodes. Graphs are randomly generated using the default Barabási-Albert model in igraph (Csardi and Nepusz, 2006). Then, 10% of the nodes are randomly selected to be pathways, so the heat flow can be dispelled.

We also consider two scenarios, depending on if **(I)** the input list has a fixed size of 30 compounds, or **(II)** it scales as 10% of the graph nodes. For each combination of parameters, a benchmark of 30 permutations is run 10 times and the trends are depicted in (Fig. 83). The differences between sampling strategies seem irrelevant compared to the solving method. Solving the linear system seems a good option for small inputs that do not scale with the graph order, but computing the inverse seems to scale better as the vector G becomes less sparse. However, the latter requires a vast amount of memory to store the matrix, so the best implementation will depend on the graph order and memory availability. All the benchmarks have been executed in a desktop workstation (Intel i5 650 at 3.2GHz, 16Gb RAM memory).

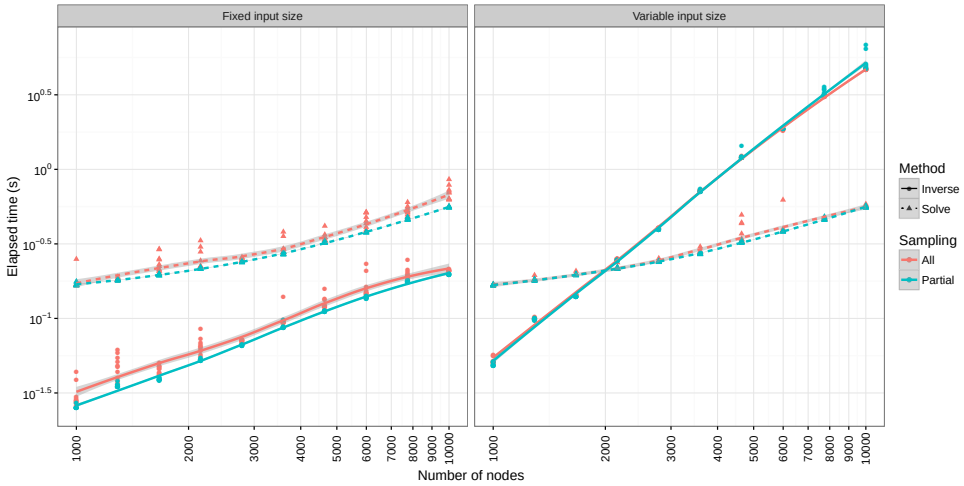


Figure 83: Computational cost of several strategies for computing 30 permutations, that is, 30 null temperatures for all the nodes. These simulations have been performed 10 times with each combination of parameters. In the left figure, the input size is kept constant and equal to 30 compounds, whereas in the right it scales with the graph nodes (10%). Two methods to compute the temperatures are compared: direct resolution (solve) and explicit computation of the R_{HD} matrix (inverse). Likewise, two sampling strategies are explored: permute the whole G vector (all) or just draw the n_{in} compounds (partial).

C.5.4 Damping factor influence

The damping factor in the PageRank setup (see Appendix S3) is a model parameter that could affect the results if set differently. Although the standard value $d = 0.85$ was used, we analysed the parameter sensibility by sweeping several values of d, computing the z-scores and reporting the top 250 nodes (Figs. 84, 85). The normalised scores seem stable in a wide range of choices of d, therefore its choice does not seem a critical issue.

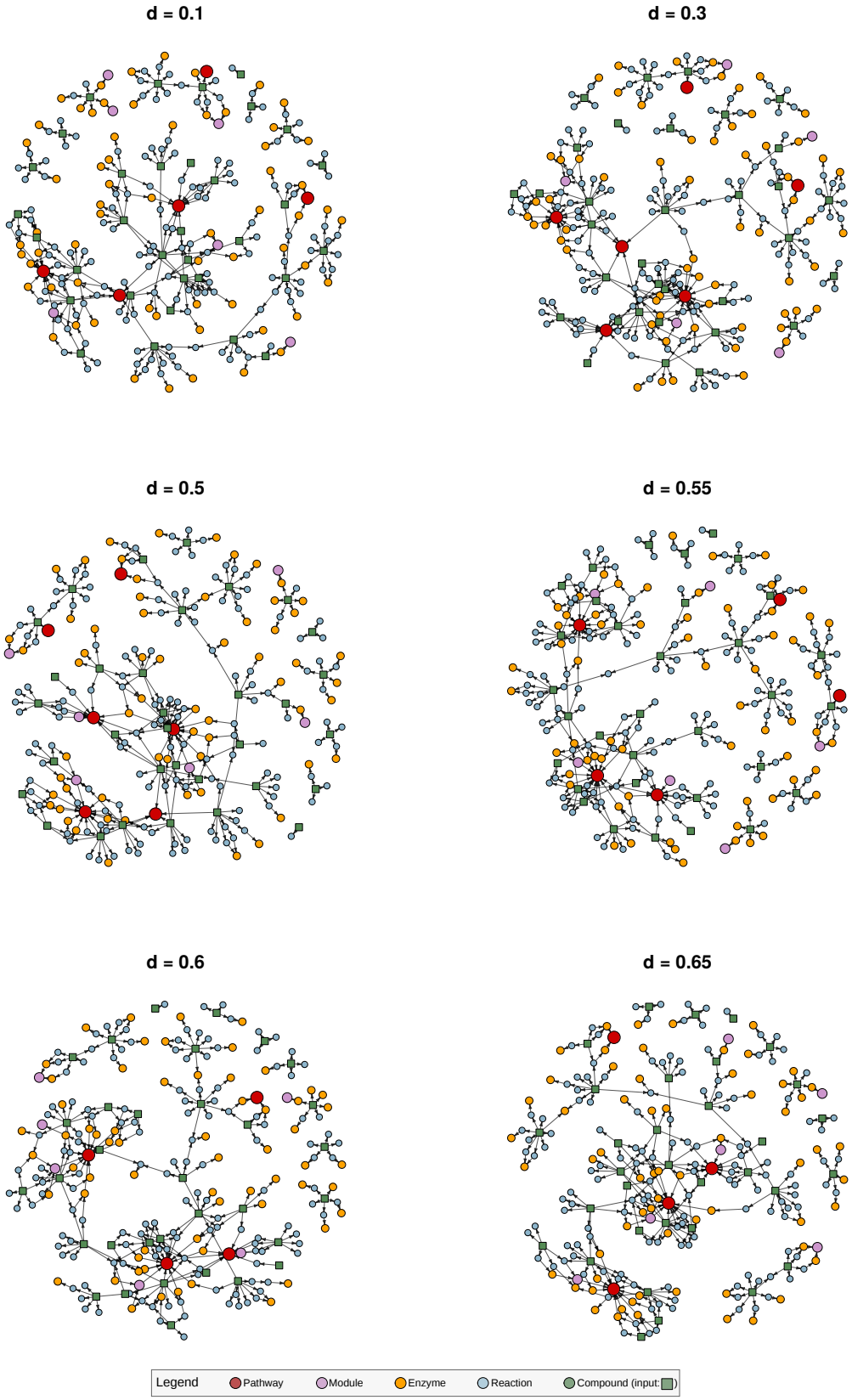


Figure 84: Damping factor impact. The normalised z-scores show consistent solutions for a range of damping factors.

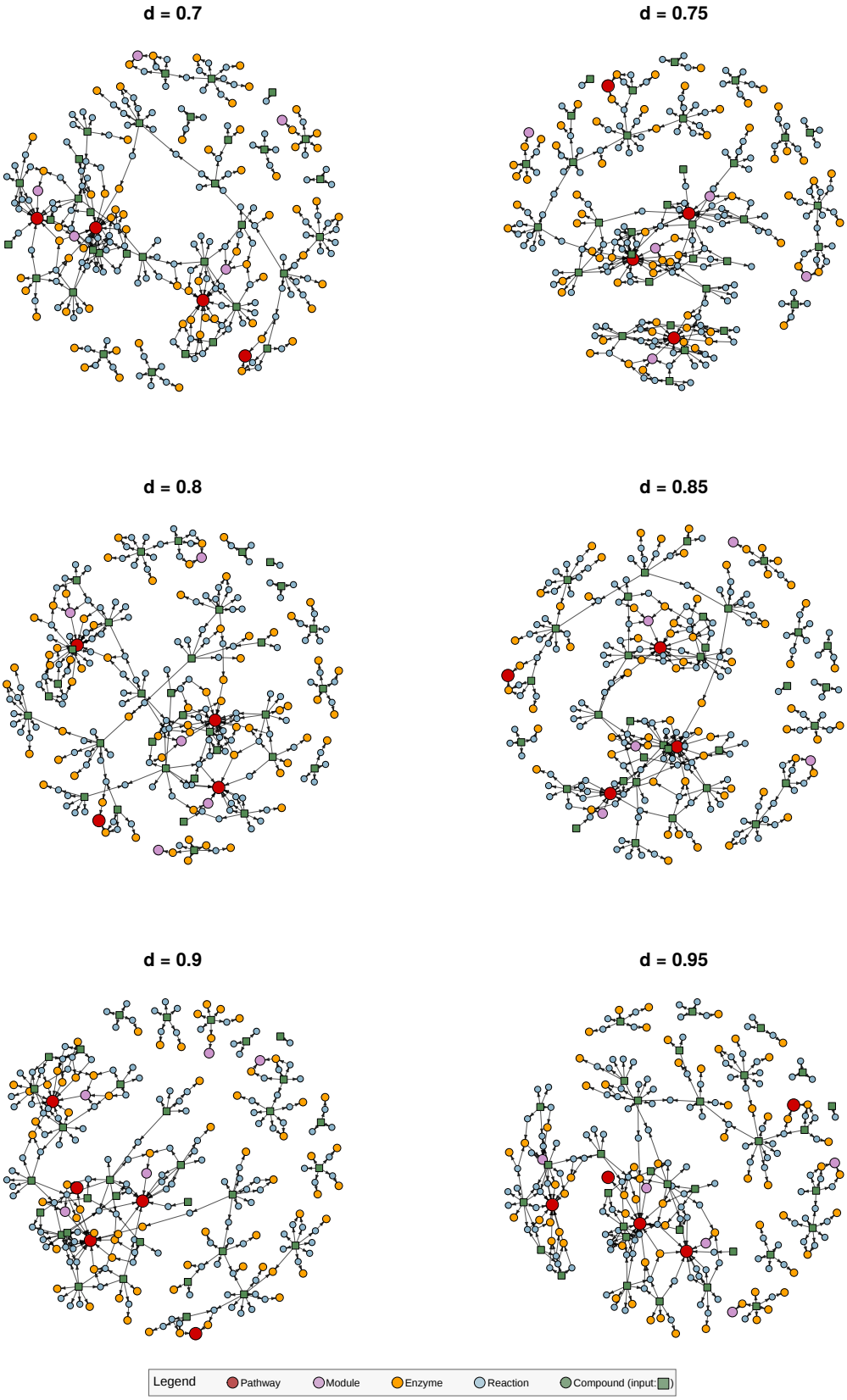


Figure 85: Damping factor impact (continued). The normalised z-scores show consistent solutions for a range of damping factors.

REFERENCES

Bapat, RB

2004 "Resistance matrix of a weighted graph", *Communications in Mathematical and in Computer Chemistry/MATCH*, 50, pp. 73-82.

Bonals, Lluís Albert

2005 *Transferència de calor: apunts de classe*, Publicacions d'Abast.

Csardi, Gabor and Tamas Nepusz

2006 "The igraph software package for complex network research", *InterJournal, Complex Systems*, 1695, 5, pp. 1-9.

Kanehisa, Minoru, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshiaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, and Yoshihiro Yamanishi

2008 "KEGG for linking genomes to life and the environment." *Nucleic Acids Research*, 36, Database-Issue, pp. 480-484, http://dblp.uni-trier.de/db/journals/nar/nar36.html#KanehisaAGHHIKKOTY08.

Karnovsky, Alla, Terry E. Weymouth, Tim Hull, V. Glenn Tarcea, Giovanni Scardoni, Carlo Laudanna, Maureen A. Sartor, Kathleen A. Stringer, H. V. Jagadish, Charles F. Burant, Brian D. Athey, and Gilbert S. Omenn

2012 "Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data." *Bioinformatics*, 28, 3, pp. 373-380, http://dblp.uni-trier.de/db/journals/bioinformatics/bioinformatics28.html#KarnovskyWHTSLSSJBAO12.

North, Bernard V, David Curtis, and Pak C Sham

2002 "A note on the calculation of empirical P values from Monte Carlo procedures", *American Journal of Human Genetics*, 71, 2, p. 439.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd

1999 "The PageRank citation ranking: bringing order to the Web."

Tenenbaum, Dan

2016 *KEGGREST: Client-side REST access to KEGG*.

Vandin, Fabio, Eli Upfal, and Benjamin J Raphael

2011 "Algorithms for detecting significantly mutated pathways in cancer", *Journal of Computational Biology*, 18, 3, pp. 507-522.

D THE R PACKAGE FELLA

D.1 ADDITIONAL FILE 1: QUICKSTART

D.1.1 Introduction

FELLA is an R package that brings a new concept for metabolomics data interpretation. The starting point of this data enrichment is a list of affected metabolites, which can stem from a contrast between experimental groups. This list, that may vary in size, encompasses key role players from different **biological pathways** that generate a biological perturbation.

The classical way to analyse this list is the **over representation analysis**. Each metabolic pathway has a statistic, the number of affected metabolites in it, that yields a p-value. After correcting for multiple testing, a list of prioritised pathways helps performing a quality check on the data and suggesting novel biological mechanisms related to the data. Subsequent generations of **pathway analysis** methods attempt to include quantitative and/or topological data in the statistics in order to improve power for subtle signals, but the interpretation of a prioritised pathway list remains a challenge.

Package FELLA, on the other hand, introduces a comprehensive output that encompasses other biological entities that coherently relate the top ranked pathways. The priorisation of the pathways and other entiteis stems from a diffusion process on a holistic **graph representation** of the **KEGG database**. FELLA needs:

1. The KEGG graph and other complementary data files. This is stored in a unique `FELLA.DATA` S4 object.
2. A list of affected metabolites (KEGG compounds). This is stored in a unique `FELLA.USER` S4 object, along with user analyses.

D.1.2 Loading the KEGG data

This vignette makes use of sample data that contains small subgraph of FELLA's KEGG graph (mid 2017 KEGG release). All the necessary contextual data is stored in an S4 data structure with class `FELLA.DATA`. Several functions need access to the contextual data, passed as an argument called `data`, being the enrichment itself among them.

This appendix is based on the four vignettes of the FELLA package (https://doi.org/doi:10.18129/B9.bioc.FELLA, accessed 31/12/2019), supplementary data (Additional files 1-4) of: Picart-Armada, Sergio, Francesc Fernández-Albert, Maria Vinaixa, Oscar Yanes, and Alexandre Perera-Lluna. "FELLA: an R package to enrich metabolomics data". *BMC bioinformatics 19, no. 1 (2018): 538.*

```
library(FELLA)

data("FELLA.sample")
class(FELLA.sample)
```

```
## [1] "FELLA.DATA"
## attr(,"package")
## [1] "FELLA"
```

```
show(FELLA.sample)
```

```
## General data:
## - KEGG graph:
##   * Nodes:  670
##   * Edges:  1677
##   * Density:  0.003741383
##   * Categories:
##     + pathway [2]
##     + module [6]
##     + enzyme [58]
##     + reaction [279]
##     + compound [325]
##   * Size:  366.9 Kb
## - KEGG names are ready.
## ----------------------------
## Hypergeometric test:
## - Matrix is ready
##   * Dim:  325 x 2
##   * Size:  25 Kb
## ----------------------------
## Heat diffusion:
## - Matrix not loaded.
## - RowSums are ready.
## ----------------------------
## PageRank:
## - Matrix not loaded.
## - RowSums are ready.
```

Keep in mind that `FELLA.DATA` objects need to be constructed only once by using `buildGraphFromKEGGREST` and `buildDataFromGraph`, in that precise order. This will store them in a local path and they should be loaded through `loadKEGGdata`. The user is disadvised from manually modifying the database internal files and the `FELLA.DATA` object slots not to corrupt the database.

D.1.3 Loading the metabolomics summary data

The second block of necessary data is a list of affected metabolites, which shoud be specified as KEGG compound IDs. Provided is a list of hypotheti-

cal affected metabolites belonging to the graph, to which some decoys that do not map to the graph are added.

```
data("input.sample")
input.full <- c(input.sample, paste0("intruder", 1:10))

show(input.full)
```

```
##  [1] "C00143"     "C00546"     "C04225"     "C16328"     "C00091"
##  [6] "C15979"     "C16333"     "C05264"     "C05258"     "C00011"
## [11] "C00083"     "C00044"     "C05266"     "C00479"     "C05280"
## [16] "C01352"     "C05268"     "C16329"     "C00334"     "C05275"
## [21] "C14145"     "C00081"     "C04253"     "C00027"     "C00111"
## [26] "C00332"     "C00003"     "C00288"     "C05467"     "C00164"
## [31] "intruder1"  "intruder2"  "intruder3"  "intruder4"  "intruder5"
## [36] "intruder6"  "intruder7"  "intruder8"  "intruder9"  "intruder10"
```

Compounds are introduced through the `defineCompounds` function and provide the first `FELLA.USER` user data object containing the mapped compounds and empty analyses slots. The user should always build `FELLA.USER` objects through `defineCompounds` instead of manipulating the slots of the object manually - this might skip quality checks.

```
myAnalysis <- defineCompounds(
    compounds = input.full,
    data = FELLA.sample)
```

```
## No background compounds specified. Default background will be used.

## Warning in defineCompounds(compounds = input.full, data = FELLA.sample):
## Some compounds were introduced as affected but they do not belong to
## the background. These compounds will be excluded from the analysis. Use
## 'getExcluded' to see them.
```

Note that a warning message informs the user that some compounds did not map to the KEGG compound collection. Compounds that successfully mapped can be obtained through `getInput`,

```
getInput(myAnalysis)
```

```
##  [1] "C00003" "C00011" "C00027" "C00044" "C00081" "C00083" "C00091"
##  [8] "C00111" "C00143" "C00164" "C00288" "C00332" "C00334" "C00479"
## [15] "C00546" "C01352" "C04225" "C04253" "C05258" "C05264" "C05266"
## [22] "C05268" "C05275" "C05280" "C05467" "C14145" "C15979" "C16328"
## [29] "C16329" "C16333"
```

while compounds that were excluded because of mismatch can be accessed through `getExcluded`:

```
getExcluded(myAnalysis)
```

```
##  [1] "intruder1"  "intruder2"  "intruder3"  "intruder4"  "intruder5"
##  [6] "intruder6"  "intruder7"  "intruder8"  "intruder9"  "intruder10"
```

Keep in mind that exact matching is sought, so be extremely careful with **whitespaces**, tabs or similar characters that might create mismatches. For example:

```
input.fail <- paste0(" ", input.full)
defineCompounds(
    compounds = input.fail,
    data = FELLA.sample)
```

```
## Error in defineCompounds(compounds = input.fail, data = FELLA.sample): None of
```

D.1.4 Enriching the data

Once the `FELLA.DATA` and the `FELLA.USER` with the affected metabolites are ready, the data can be easily enriched.

Enrichment methods

There are three methods to enrich:

1. **Hypergeometric test** (`method = "hypergeom"`): it performs the metabolite-sampling hypergeometric test using the connections in `FELLA`'s KEGG graph. This is included for completeness and does not include the contextual novelty of the diffusive methods.
2. **Diffusion** (`method = "diffusion"`): it performs sub-network analysis on the KEGG graph to extract a meaningful subgraph. This subgraph can be plotted an interpreted
3. **PageRank** (`method = "pagerank"`): analogous to `"diffusion"` but using the directed diffusion, which matches the PageRank algorithm for web ranking.

Statistical approximations

For methods `"diffusion"` and `"pagerank"`, two statistical approximations are proposed:

1. **Normal approximation** (`approx = "normality"`): scores are computed through z-scores based on analytical expected value and covariance matrix of the null model for diffusion. This approximation is deterministic and fast.
2. **Monte Carlo trials** (`approx = "simulation"`): scores are computed through Monte Carlo trials of the random variables. This approximation requires computing the random trials, governed by the `ntrials` argument.

Enrichment: methods, approximations and wrapper function

The function `enrich` wraps the functions `defineCompounds`, `runHypergeom`, `runDiffusion` and `runPagerank` in an easily usable manner, returning a `FELLA.USER` object with complete analyses.

```
myAnalysis <- enrich(
    compounds = input.full,
    method = "diffusion",
    approx = "normality",
    data = FELLA.sample)
```

```
## No background compounds specified. Default background will be used.

## Warning in defineCompounds(compounds = compounds, compoundsBackground =
## compoundsBackground, : Some compounds were introduced as affected but they
## do not belong to the background. These compounds will be excluded from the
## analysis. Use 'getExcluded' to see them.

## Running diffusion...

## Computing p-scores through the specified distribution.

## Done.
```

The output is quite informative and aggregates all the warnings. Let's compare an empty `FELLA.USER` object

```
show(new("FELLA.USER"))
```

```
## Compounds in the input: empty
## Background compounds: all available compounds (default)
## ----------------------------
## Hypergeometric test: not performed
## ----------------------------
## Heat diffusion: not performed
## ----------------------------
## PageRank: not performed
```

to the output of a processed one:

```
show(myAnalysis)
```

```
## Compounds in the input: 30
##  [1] "C00003" "C00011" "C00027" "C00044" "C00081" "C00083" "C00091"
##  [8] "C00111" "C00143" "C00164" "C00288" "C00332" "C00334" "C00479"
## [15] "C00546" "C01352" "C04225" "C04253" "C05258" "C05264" "C05266"
## [22] "C05268" "C05275" "C05280" "C05467" "C14145" "C15979" "C16328"
## [29] "C16329" "C16333"
## Background compounds: all available compounds (default)
## ----------------------------
```

```
## Hypergeometric test: not performed
## ----------------------------
## Heat diffusion: ready.
## P-scores under 0.05:  86
## ----------------------------
## PageRank: not performed
```

The wrapper function `enrich` can run the three analysis at once with the option `method = listMethods()`, or only the desired ones providing them as a character vector:

```
myAnalysis <- enrich(
    compounds = input.full,
    method = listMethods(),
    approx = "normality",
    data = FELLA.sample)

show(myAnalysis)
```

```
## Compounds in the input: 30
##  [1] "C00003" "C00011" "C00027" "C00044" "C00081" "C00083" "C00091"
##  [8] "C00111" "C00143" "C00164" "C00288" "C00332" "C00334" "C00479"
## [15] "C00546" "C01352" "C04225" "C04253" "C05258" "C05264" "C05266"
## [22] "C05268" "C05275" "C05280" "C05467" "C14145" "C15979" "C16328"
## [29] "C16329" "C16333"
## Background compounds: all available compounds (default)
## ----------------------------
## Hypergeometric test: ready.
## Top 2 p-values:
##     hsa00640     hsa00010
## 8.540386e-09 9.999888e-01
##
## ----------------------------
## Heat diffusion: ready.
## P-scores under 0.05:  86
## ----------------------------
## PageRank: ready.
## P-scores under 0.05:  70
```

The wrapped functions work in a similar way, here is an example with `runDiffusion`:

```
myAnalysis_bis <- runDiffusion(
    object = myAnalysis,
    approx = "normality",
    data = FELLA.sample)
```

```
## Running diffusion...

## Computing p-scores through the specified distribution.
```

```
## Done.
```

```
show(myAnalysis_bis)
```

```
## Compounds in the input: 30
##  [1] "C00003" "C00011" "C00027" "C00044" "C00081" "C00083" "C00091"
##  [8] "C00111" "C00143" "C00164" "C00288" "C00332" "C00334" "C00479"
## [15] "C00546" "C01352" "C04225" "C04253" "C05258" "C05264" "C05266"
## [22] "C05268" "C05275" "C05280" "C05467" "C14145" "C15979" "C16328"
## [29] "C16329" "C16333"
## Background compounds: all available compounds (default)
## ----------------------------
## Hypergeometric test: ready.
## Top 2 p-values:
##     hsa00640     hsa00010
## 8.540386e-09 9.999888e-01
##
## ----------------------------
## Heat diffusion: ready.
## P-scores under 0.05:  86
## ----------------------------
## PageRank: ready.
## P-scores under 0.05:  70
```

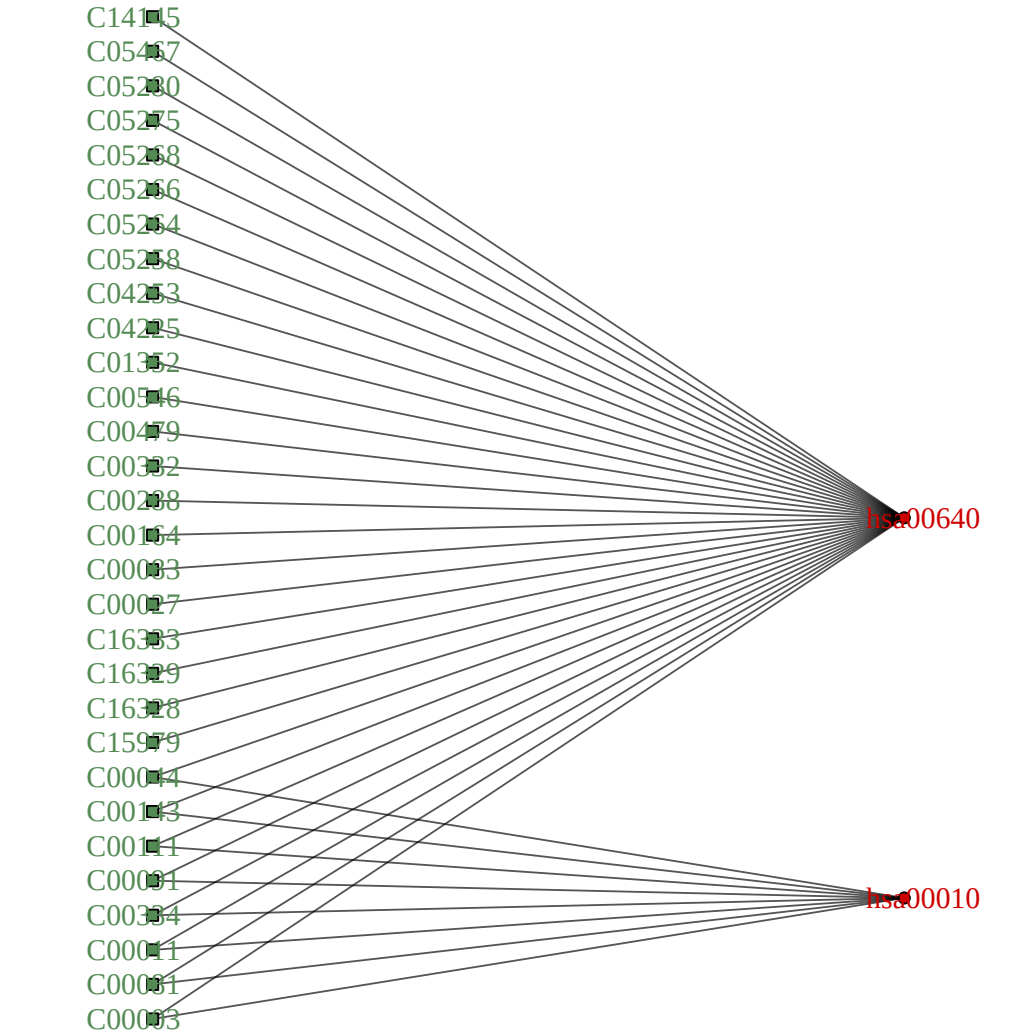
D.1.5 Visualising the results

The method `plot` for data from the package `FELLA` allows a friendly visualisation of the relevant part of the KEGG graph.

Hypergeom

In the case `method = "hypergeom"` the plot encompasses a bipartite graph that contains top pathways and affected compounds. In that case, `threshold = 1` allows the visualisation of both pathways; otherwise a plot with only one pathway would be quite uninformative.

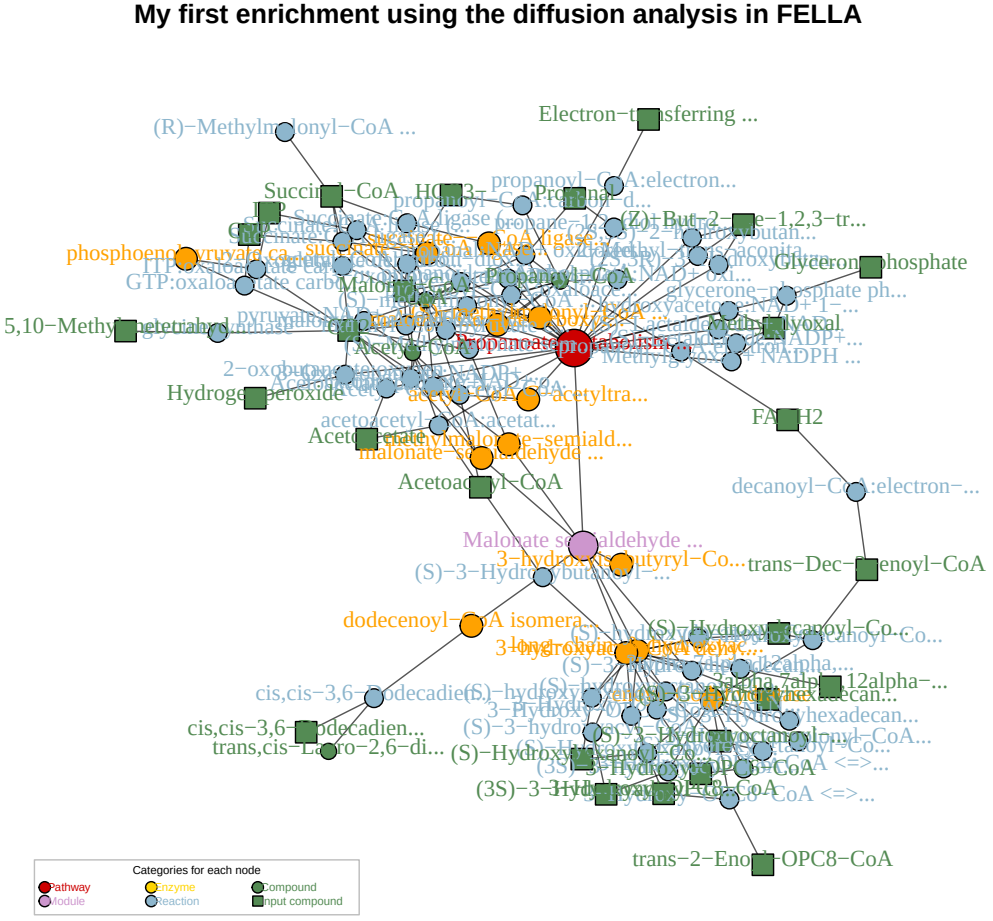
```
plot(
    x = myAnalysis,
    method = "hypergeom",
    main = "My first enrichment using the hypergeometric test in FELLA",
    threshold = 1,
    data = FELLA.sample)
```



Diffusion

For `method = "diffusion"` the graph contains a richer representations involving **modules, enzymes and reactions** that link affected pathways and compounds.

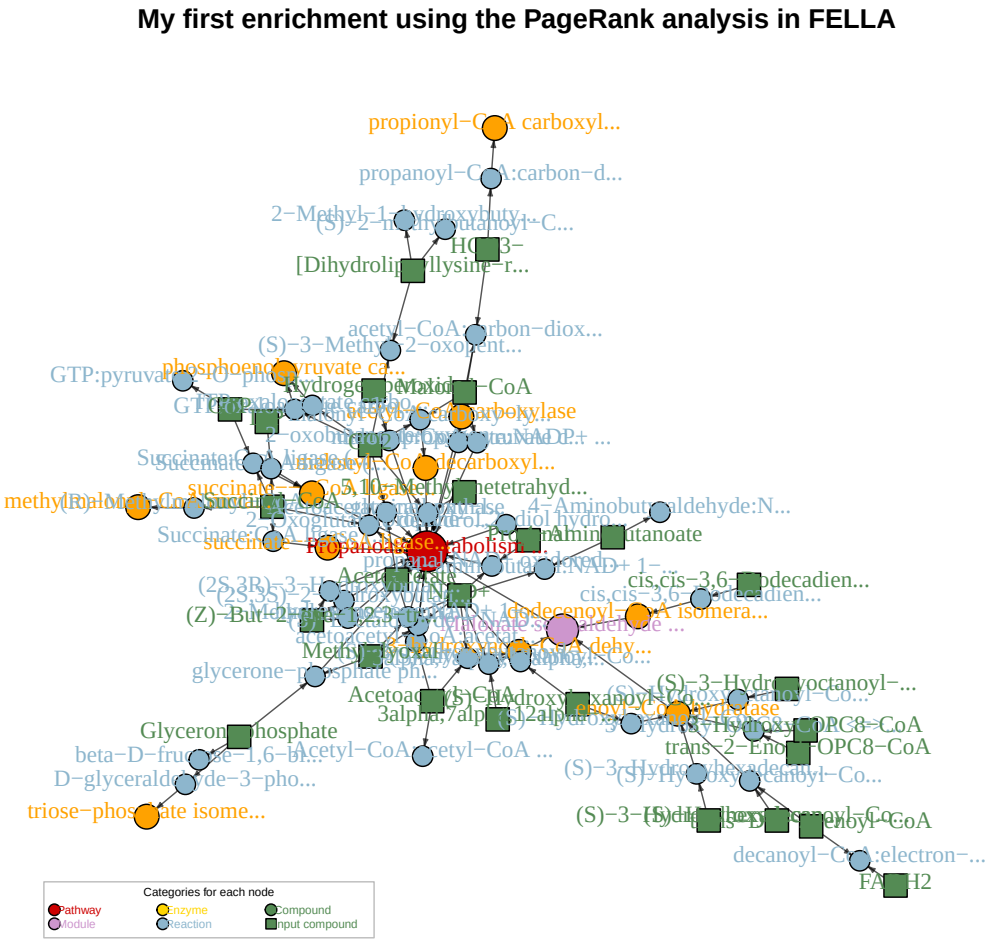
```
plot(
    x = myAnalysis,
    method = "diffusion",
    main = "My first enrichment using the diffusion analysis in FELLA",
    threshold = 0.1,
    data = FELLA.sample)
```



PageRank

For `method = "pagerank"` the concept is analogous to diffusion:

```
plot(
    x = myAnalysis,
    method = "pagerank",
    main = "My first enrichment using the PageRank analysis in FELLA",
    threshold = 0.1,
    data = FELLA.sample)
```



D.1.6 Exporting the results

FELLA offers several exporting alternatives, both for the R environment and for external software.

Exporting inside R

The appropriate functions to export the results inside R are `generateResultsTable` for a **data.frame** object:

```
myTable <- generateResultsTable(
    object = myAnalysis,
    method = "diffusion",
    threshold = 0.1,
    data = FELLA.sample)
```

```
## Writing diffusion results...

## Done.
```

```
knitr::kable(head(myTable, 20))
```

...and `generateResultsGraph` for a **graph** in igraph format:

KEGG.id	Entry.type	KEGG.name	p.score
hsa00640	pathway	Propanoate metabolism - Homo sapiens (human)	0.0036894
M00013	module	Malonate semialdehyde pathway, propanoyl-CoA ...	0.0044683
1.1.1.211	enzyme	long-chain-3-hydroxyacyl-CoA dehydrogenase	0.0371099
1.1.1.35	enzyme	3-hydroxyacyl-CoA dehydrogenase	0.0392511
1.2.1.18	enzyme	malonate-semialdehyde dehydrogenase (acetylat...	0.0069255
1.2.1.27	enzyme	methylmalonate-semialdehyde dehydrogenase (Co...	0.0165439
2.3.1.9	enzyme	acetyl-CoA C-acetyltransferase	0.0085923
3.1.2.4	enzyme	3-hydroxyisobutyryl-CoA hydrolase	0.0786804
4.1.1.32	enzyme	phosphoenolpyruvate carboxykinase (GTP)	0.0700429
4.1.1.41	enzyme	(S)-methylmalonyl-CoA decarboxylase	0.0223899
4.1.1.9	enzyme	malonyl-CoA decarboxylase	0.0002538
4.2.1.17	enzyme	enoyl-CoA hydratase	0.0015731
5.3.3.8	enzyme	dodecenoyl-CoA isomerase	0.0164255
6.2.1.4	enzyme	succinate—CoA ligase (GDP-forming)	0.0019142
6.2.1.5	enzyme	succinate—CoA ligase (ADP-forming)	0.0125330
R00209	reaction	pyruvate:NAD+ 2-oxidoreductase (CoA-acetylati...	0.0885938
R00233	reaction	malonyl-CoA carboxy-lyase (acetyl-CoA-forming...	0.0000698
R00238	reaction	Acetyl-CoA:acetyl-CoA C-acetyltransferase	0.0001037
R00353	reaction	malonyl-CoA:pyruvate carboxytransferase	0.0065794
R00405	reaction	Succinate:CoA ligase (ADP-forming)	0.0468613

```
myGraph <- generateResultsGraph(
    object = myAnalysis,
    method = "diffusion",
    threshold = 0.1,
    data = FELLA.sample)

show(myGraph)
```

```
## IGRAPH 6bf1c19 UNW- 102 166 --
## + attr: name (v/c), com (v/n), NAME (v/x), entrez (v/x), label
## | (v/c), input (v/l), weight (e/n)
## + edges from 6bf1c19 (vertex names):
##  [1] hsa00640--M00013    M00013  --1.1.1.211 M00013  --1.1.1.35
##  [4] M00013  --1.2.1.18  M00013  --1.2.1.27  hsa00640--2.3.1.9
##  [7] M00013  --3.1.2.4   hsa00640--4.1.1.41  hsa00640--4.1.1.9
## [10] M00013  --4.2.1.17  M00013  --5.3.3.8   hsa00640--6.2.1.4
## [13] hsa00640--6.2.1.5   4.1.1.9 --R00233    2.3.1.9 --R00238
## [16] hsa00640--R00353    6.2.1.5 --R00405    4.1.1.32--R00431
## [19] 6.2.1.4 --R00432    1.2.1.18--R00705    1.2.1.27--R00705
## + ... omitted several edges
```

Exporting outside R

Results can be saved as permanent files. The **data.frame** data format can be saved as a `.csv` file:

```
myTempDir <- tempdir()
myExp_csv <- paste0(myTempDir, "/table.csv")
exportResults(
```

```
    format = "csv",
    file = myExp_csv,
    method = "pagerank",
    threshold = 0.1,
    object = myAnalysis,
    data = FELLA.sample)
```

```
## Exporting to a csv file...

## Writing pagerank results...

## Done.

## Done
```

```
test <- read.csv(file = myExp_csv)
knitr::kable(head(test))
```

KEGG.id	Entry.type	KEGG.name	p.score
hsa00640	pathway	Propanoate metabolism - Homo sapiens (human)	0.0000085
M00013	module	Malonate semialdehyde pathway, propanoyl-CoA ...	0.0010330
1.1.1.35	enzyme	3-hydroxyacyl-CoA dehydrogenase	0.0422528
4.1.1.32	enzyme	phosphoenolpyruvate carboxykinase (GTP)	0.0088747
4.1.1.9	enzyme	malonyl-CoA decarboxylase	0.0005280
4.2.1.17	enzyme	enoyl-CoA hydratase	0.0003343

In the same line, the **graph** can be saved in `RData`:

```
myExp_graph <- paste0(myTempDir, "/graph.RData")
exportResults(
    format = "igraph",
    file = myExp_graph,
    method = "pagerank",
    threshold = 0.1,
    object = myAnalysis,
    data = FELLA.sample)
```

```
## Exporting to a RData file using 'igraph' object...

## Done
```

```
stopifnot("graph.RData" %in% list.files(myTempDir))
```

Other formats exported by igraph are also available, internally using their function `igraph::write.graph`. Check the **format** argument of `?igraph::write.graph` for a list of the supported formats. For example, using `"pajek"` format:

```
myExp_pajek <- paste0(myTempDir, "/graph.pajek")
exportResults(
    format = "pajek",
    file = myExp_pajek,
    method = "diffusion",
    threshold = 0.1,
    object = myAnalysis,
    data = FELLA.sample)
```

```
## Exporting to the format pajek using igraph...

## Done
```

```
stopifnot("graph.pajek" %in% list.files(myTempDir))
```

This option is toggled if the format does not match any other predefined export option.

D.1.7 Session info

For reproducibility purposes, below is the `sessionInfo()` output:

```
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/atlas-base/atlas/libblas.so.3.0
## LAPACK: /usr/lib/atlas-base/atlas/liblapack.so.3.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=es_ES.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=es_ES.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=es_ES.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] magrittr_1.5         igraph_1.2.4.1       KEGGREST_1.24.1
##  [4] org.Mm.eg.db_3.8.2   org.Hs.eg.db_3.8.2   AnnotationDbi_1.46.0
##  [7] IRanges_2.17.5       S4Vectors_0.21.24    Biobase_2.44.0
## [10] BiocGenerics_0.29.2  FELLA_1.5.3          knitr_1.22
```

```
##
## loaded via a namespace (and not attached):
##  [1] progress_1.2.2     xfun_0.6           lattice_0.20-38
##  [4] tcltk_3.6.2        vctrs_0.2.0        htmltools_0.3.6
##  [7] yaml_2.2.0         blob_1.2.0         XML_3.98-1.20
## [10] rlang_0.4.0        pillar_1.4.0       DBI_1.0.0
## [13] bit64_0.9-7        plyr_1.8.4         stringr_1.4.0
## [16] zlibbioc_1.29.0    Biostrings_2.51.5  GOSemSim_2.10.0
## [19] evaluate_0.13      memoise_1.1.0      biomaRt_2.40.1
## [22] curl_3.3           highr_0.8          Rcpp_1.0.1
## [25] backports_1.1.4    BiocManager_1.30.4 XVector_0.23.2
## [28] bit_1.1-14         BiocStyle_2.12.0   hms_0.5.0
## [31] png_0.1-7          digest_0.6.18      stringi_1.4.3
## [34] grid_3.6.2         tools_3.6.2        bitops_1.0-6
## [37] RCurl_1.95-4.12    RSQLite_2.1.1      tibble_2.1.1
## [40] GO.db_3.8.2        crayon_1.3.4       pkgconfig_2.0.2
## [43] zeallot_0.1.0      Matrix_1.2-18      prettyunits_1.0.2
## [46] assertthat_0.2.1   rmarkdown_1.12     httr_1.4.0
## [49] R6_2.4.0           compiler_3.6.2
```

D.2 ADDITIONAL FILE 2: MAIN VIGNETTE

D.2.1 Abstract

Pathway enrichment techniques are useful for giving context to experimental metabolomics data. The primary analysis of the raw metabolomics data leads to annotated metabolites with abundance measures. These metabolites are compared between experimental conditions, in order to find discriminative molecular signatures. The secondary analysis of the dataset aims at giving context to the affected metabolites in terms of the prior biological knowledge gathered in metabolic pathways. Several statistical approaches are available to derive a list of prioritised metabolic pathways that relate to the underlying changes in metabolite abundances. However, the interpretation of a prioritised pathway list remains challenging, as pathways are not disjoint and show overlap and cross talk effects. Furthermore, it is not straightforward to automatically propose novel enzymatic targets given a pathway enrichment.

We introduce *FELLA*, an R package to perform a network-based enrichment of a list of affected metabolites. *FELLA* builds a hierarchical network representation of the organism of choice using the Kyoto Encyclopedia of Genes and Genomes, which contains pathways, modules, enzymes, reactions and metabolites. The enrichment is accomplished by applying diffusion algorithms in the knowledge network. Flow is introduced in the metabolites from the input list and propagates to the rest of nodes, resulting in diffusion scores for all the nodes in the network. The top scoring nodes contain not only relevant pathways, but also the intermediate entities that build a plausible explanation on how the input metabolites translate into reported pathways. The highlighted sub-network can shed light on pathway cross talk under the experimental condition and potential enzymatic targets for further study.

The implementation and the programmatic use of *FELLA* is hereby described, along with a graphical user interface that wraps the package functionality. The algorithmic part in *FELLA* was previously validated on the study of an uncharacterised mitochondrial protein. The functionality of *FELLA* has been demonstrated on three public human metabolomics studies, respectively on (a) ovarian cancer cells, (b) dry eye and (c) malaria and other febrile illnesses. *FELLA* has been able to reproduce findings from the original publications and to report sub-network representations that can be manually handled.

D.2.2 Introduction

Metabolomics is the science that studies the chemical reactions in living organisms by quantifying their lightweight molecules, called metabolites. The utilities of metabolomics range from disease diagnosis through biomarkers and personalised medicine to the generation of biological knowledge (Madsen et al., 2010).

Metabolomics data is mainly acquired through technologies such as, but not limited to, Nuclear Magnetic Resonance (NMR) and Mass Spectrome-

try (MS). MS is usually preceded by Liquid Chromatography (LC) or Gas Chromatography (GC) (Weckwerth, 2003). The primary analysis of the raw metabolomics data can be achieved through publicly available tools: the R packages *xmcs* (Smith et al., 2006) for peak identification and *CAMERA* (Kuhl et al., 2011) for peak annotation. There are pipelines that cover the whole process, for example the online tool MeltDB (Kessler et al., 2013) or the R package *MAIT* (Fernández-Albert et al., 2014). Metabolites found in samples are mapped to specral databases such as the Human Metabolome Database (Wishart et al., 2012).

The secondary analysis, or data interpretation, starts when the metabolites are mapped to a database and their abundances are available (Chagoyen and Pazos, 2012). The existence of experimental conditions enables a statistical differential analysis that yields a set of metabolites that exhibit changes in the intervention. It is, however, increasingly important to understand the underlying biological perturbation by giving context to the affected metabolites rather than focusing on the ability to classify samples through them (Madsen et al., 2010). Pathway analysis is a fundamental methodology for data interpretation (Khatri et al., 2012) that enriches the affected metabolites with current knowledge on biology, available in pathway databases including the Kyoto Encyclopedia of Genes and Genomes or KEGG (Kanehisa, Goto, et al., 2011), Reactome (Fabregat et al., 2015) and WikiPathways (Kutmon et al., 2015). Enrichment techniques will be discussed in three categories or generations, according to the classification proposed in the review (Khatri et al., 2012). Commercial pathway analysis products such as *IPA* (QIAGEN Inc., `https://www.qiagenbioinformatics.com/products/ingenuitypathway-analysis`) are out of the scope of this work.

The first generation of methods, named over representation analysis (ORA), are based on testing if the proportion of affected metabolites within a pathway is statistically meaningful. ORA is based in statistical tests on probability distribution like the hypergeometric, binomial or chi-squared (Khatri et al., 2012). ORA is available in tools like the web servers MetaboAnalyst (Xia, Sinelnikov, et al., 2015) and IMPaLA (Kamburov et al., 2011) and the R package *clusterProfiler* (Yu, L.-G. Wang, et al., 2012). The online resource SubPathwayMiner identifies sub-pathways from KEGG pathways by mining k-cliques in each metabolic pathway prior to ORA. With this strategy, significant sub-regions can be spotted even if the whole pathway is not significant (C. Li et al., 2009).

The second generation of methods, functional class scoring (FCS), uses quantitative data instead and seeks subtle but coordinated changes in the metabolites belonging to a pathway. MSEA (Xia and Wishart, 2010) in MetaboAnalyst (Xia, Sinelnikov, et al., 2015) and IMPaLA (Kamburov et al., 2011) contain implementations of FCS for metabolomics. The R package *PAPi* calculates pathways activity scores per sample, based on the number of metabolites identified from each pathway and their relative abundances. Significantly affected pathways are found by applying an ANOVA or a t-test on those scores (Aggio et al., 2010). On the other hand, there is an ensemble approach relying on several pathway-based statistical tests (Alhamdoosh et al., 2017) and is available in the R pacakge *EGSEA*.

The third generation, known as pathway topology-based (PT) methods, further includes topological measures of the metabolites in the statistic, accounting for their inequivalence in the metabolic network. PT analyses can be performed using MetaboAnalyst (Xia, Sinelnikov, et al., 2015), where metabolites are weighted by their centrality within the pathway. The R package *MPINet* builds a pathway-level statistic that accounts for metabolite inequivalence in the global metabolic network and for bias in technical equipment (Feng Li et al., 2014).

Another perspective for understanding metabolomics data is through the construction and inquiry of metabolic networks. The *MetScape* plugin (Karnovsky et al., 2011) within the *Cytoscape* environment (Smoot et al., 2010) is useful for representing metabolite-reaction-enzyme-gene networks. *KEGGGraph* is an R package for constructing metabolic networks from the KEGG pathways (J. D. Zhang and Wiemann, 2009). *MetaboSignal* is an R package for building and examining the topology of gene-metabolite networks (Rodriguez-Martinez et al., 2017). The R package *MetaMapR* helps reduce sparsity in metabolic networks by integrating biochemical transformations, structural similarity, mass spectral similarity and empirical correlation information (Grapov et al., 2015).

Here, we introduce the R package *FELLA* for metabolomics data interpretation that combines concepts from pathway enrichment and network analysis. The main objective of *FELLA* is providing the user with a biological explanation involving biological pathways. *FELLA* starts from a single, comprehensive network consisting of metabolites, reactions, enzymes, modules and pathways as nodes. The list of affected metabolites and the pathways highlighted by *FELLA* are connected through intermediate entities -reactions, enzymes and KEGG modules- and returned as a sub-network. The intermediate entities suggest how the perturbation spreads from metabolites to pathways and how pathways cross talk. The provided enzymes are candidates for further examination, whereas new metabolites might be reported as well. *FELLA* is publicly available in `https://github.com/b2slab/FELLA` under the GPL-3 license.

D.2.3 Methodology

Implementation details

FELLA is written entirely in R (R Core Team, 2017) and relies on the *KEGGREST* R package (Tenenbaum, 2017) for retreiving KEGG, the *igraph* R package (Csardi and Nepusz, 2006) for network analysis and the *shiny* R package (Chang et al., 2017) for providing a graphical user interface.

FELLA defines two S4 classes for handling its main purposes: a *FELLA.DATA* object that encompasses the knowledge model from KEGG and a *FELLA.USER* object that contains the current analysis by the user. Table 27 contains further details about the slots and sub-slots in each one of these classes, whereas figure 86 depicts the package workflow and main functions.

FELLA contains two vignettes that illustrate its capabilities: (1) a quick-start example with the main functions applied to a toy dataset, and (2) this document, an in-depth demonstration on three real studies. This vignette

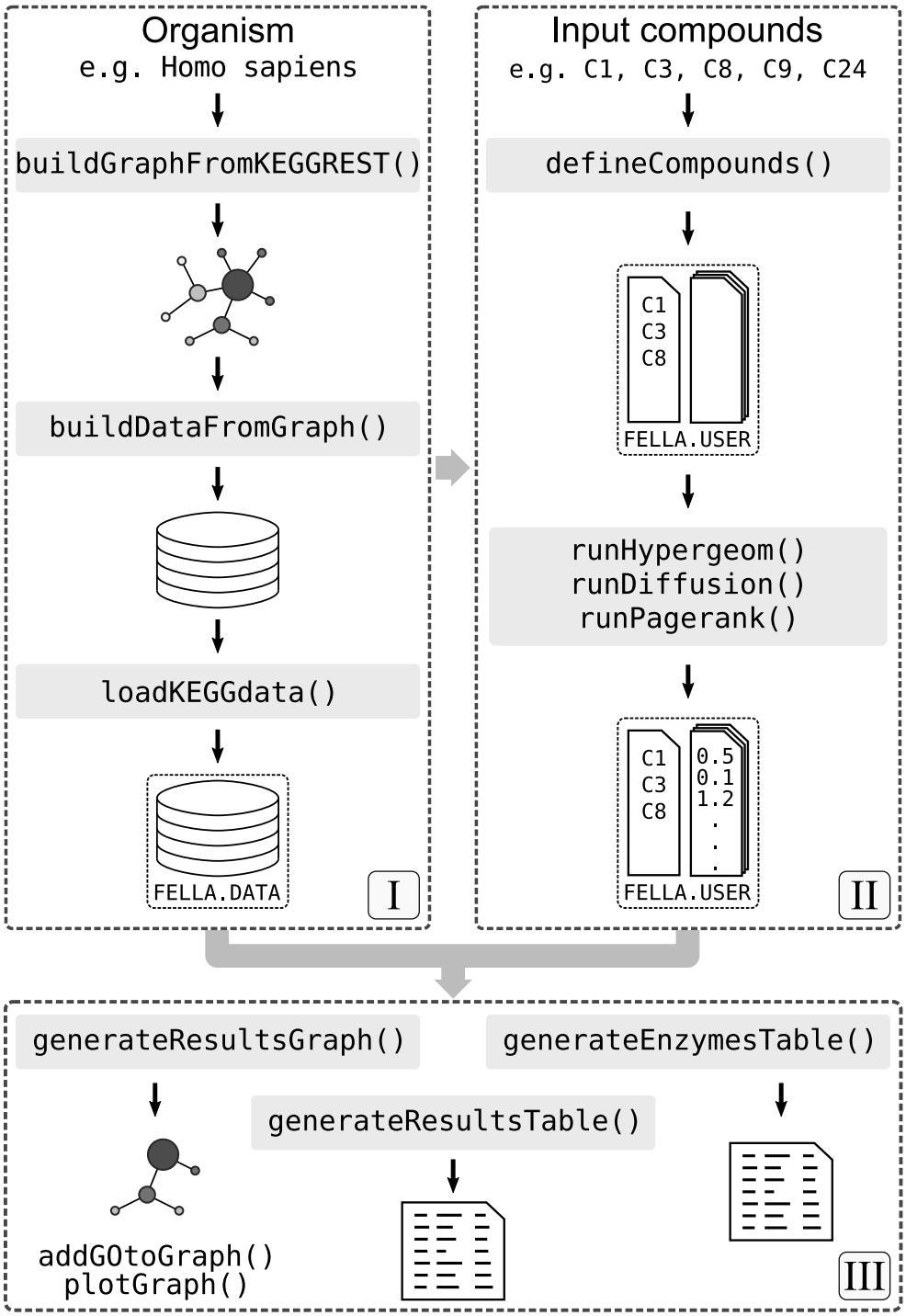


Figure 86: Design of the R package *FELLA*. Block **I** covers the creation of a graph object from an organism code and its database, which can be loaded into a *FELLA.DATA* object. This object is needed in all the following blocks. Block **II** requires block **I** and shows how to map the KEGG identifiers to the database in a *FELLA.USER* object and run the propagation algorithms (diffusion, PageRank) to score all the entities in the graph. Block **III** requires blocks **I** and **II** and exports the results as a sub-network or as a table.

Custom class	Slot	Sub-slot	Class	Description
FELLA.DATA	@keggdata	@graph	igraph	Knowledge graph object
		@id2name	list	Dictionary from KEGG ID to common name
		@pvalues.size	matrix	Matrix with largest CC size probabilities
		@id	list	Correspondence between IDs and category
		@status	character	Status indicator of the object
	@hypergeom	@matrix	Matrix	Metabolite-pathway binary relationship
	@diffusion	@matrix	matrix	Matrix to compute diffusion as a matrix-vector product
		@rowSums	vector	Internal data to compute the z-scores
		@squaredRowSums	vector	Internal data to compute the z-scores
	@pagerank	@matrix	matrix	Matrix to compute PageRank as a matrix-vector product
		@rowSums	vector	Internal data to compute the z-scores
		@squaredRowSums	vector	Internal data to compute the z-scores
FELLA.USER	@userinput	@metabolites	vector	KEGG IDs that map to the knowledge graph
		@metabolitesbackground	vector	Background KEGG IDs
		@excluded	vector	Input IDs not mapping to the knowledge graph
	@hypergeom	@valid	logical	Indicator of analysis validity
		@pvalues	vector	Pathway p-values
		@pathhits	vector	Number of hits in each pathway
		@pathbackground	vector	Number of metabolites in each pathway
		@nbackground	numeric	Number of compounds in the background
		@ninput	numeric	Number of compounds in the input
	@diffusion	@valid	logical	Indicator of analysis validity
		@pscores	vector	P-scores for each node in the network
		@approx	character	Chosen approximation
		@niter	numeric	Chosen iterations
	@parerank	@valid	logical	Indicator of analysis validity
		@pscores	vector	P-scores for each node in the network
		@approx	character	Chosen approximation
		@niter	numeric	Chosen iterations

Table 27: Summary of the S4 classes defined in *FELLA*.

requires an internet connection and can take up some time and memory to build, as it builds the internal KEGG representation for Homo sapiens on the fly.

Database and knowledge model

A distinctive feature of *FELLA* is its unique knowledge model. Instead of using individual pathway representations, either as a list of metabolites (ORA) or as a metabolic network (TP), *FELLA* builds a unique network that encompasses all the pathways at once: the KEGG graph. Figure 87 shows the hierarchical representation of the KEGG database, ranging from the small, specific molecular level (metabolite) to the large, complex unit (pathway). Intermediate levels contain, from bottom to top: reactions relating the metabolites, enzymes catalising the reactions and KEGG modules containing the enzymes. More details on the construction and curation of this structure, resemblant to the one used by MetScape (Karnovsky et al., 2011), can be found in (Picart-Armada et al., 2017). The enrichment is therefore achieved by finding a sub-network from the whole KEGG graph that is statistically relevant for a list of input metabolites.

As shown in the block (I) of figure 86, the first step is to build a KEGG graph from an organism in KEGG -Homo sapiens by default- using the `buildGraphFromKEGGREST` command. Afterwards, a local database can be built from the KEGG graph through the `buildDataFromGraph` command. The main purposes of `buildDataFromGraph` are to save (1) the matrices that allow computing diffusion and PageRank as a matrix-vector product, and (2) the null distribution of the largest connected component of a k-th order subgraph, with uniformly chosen nodes. Point (1) is required to compute the diffusion scores, whereas (2) is useful for filtering small connected components in the reported subgraphs.

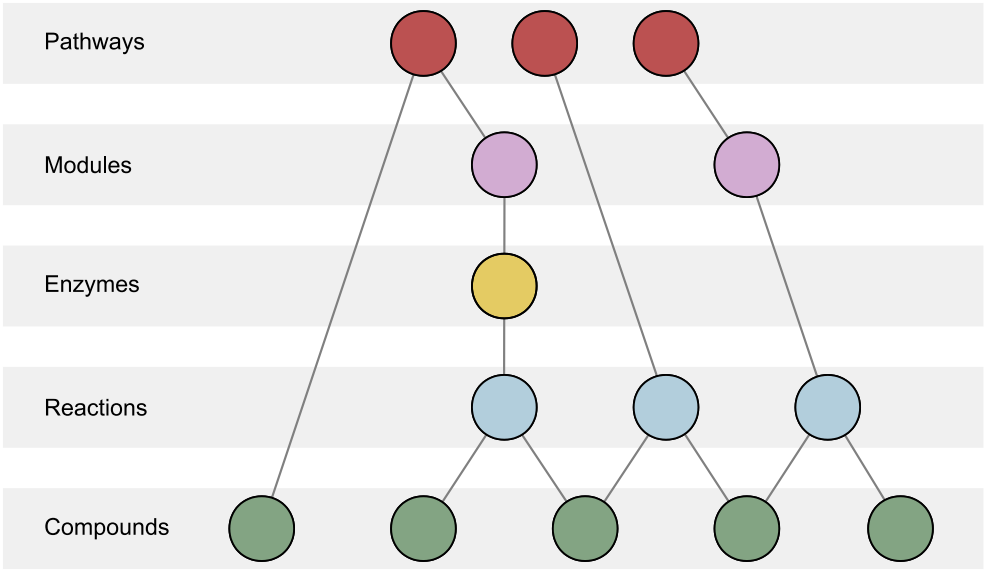


Figure 87: Internal knowledge representation from KEGG. The scheme outlines the KEGG graph, a heterogeneous network whose nodes belong to a category in KEGG: compound, reaction, enzyme, module or pathway. Lower levels are expected to be more specific entities, while top levels are broader concepts. The enrichment procedure starts from input metabolites and extracts a relevant sub-network from the KEGG graph. Figure extractred from (Picart-Armada et al., 2017)

The user should be aware that KEGG is frequently updated and therefore the derived KEGG graph can change between KEGG releases. The metadata from the KEGG version used to build a *FELLA.DATA* object can be retrieved through `getInfo`.

Enrichment analysis

Once the database is ready as a *FELLA.DATA* object and the input is formatted as a list of KEGG compounds, the enrichment can be performed. The results of the enrichment are stored in a *FELLA.USER* object, possibly using three methodologies described below.

Hypergeometric test

For completeness purposes, the hypergeometric test is included in *FELLA* in the function `runHypergeom`. As in several ORA implementations, the hypergeometric distribution is used to assess whether a biological pathway contains more hits within the input list than expected from chance given its size. Pathways are ranked according to their p-value after multiple testing correction.

Note that the results from this test will differ from a hypergeometric test using the original KEGG pathways, because metabolite-pathway connections are inferred from the KEGG graph. A metabolite is included in a pathway if the pathway can be reached from the metabolite in the upwards-directed KEGG graph, depicted in figure 89. In consequence, metabolites related to the enzymes within a pathway will belong to the pathway, even if they were not in the original definition of the KEGG pathway.

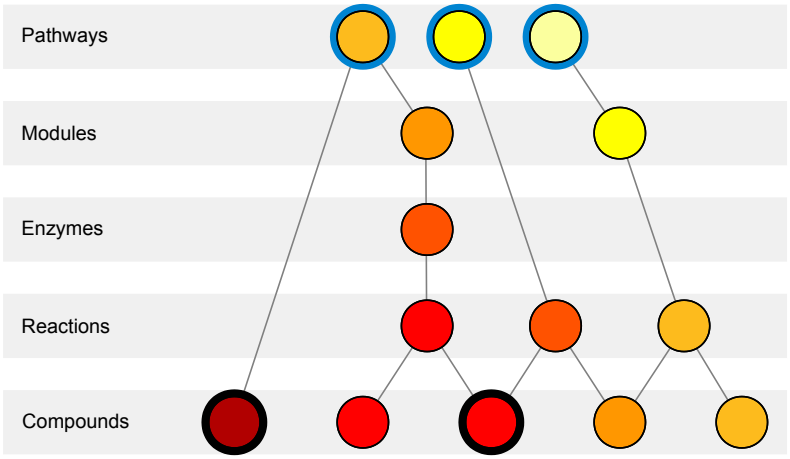


Figure 88: Network setup for the diffusion process. nput metabolites (in black rings) introduce a unitary flow in the network and only the pathway nodes (blue rings) can leak the flow. The final score of the nodes reflects the "temperature" of a stationary state. Figure extractred from (Picart-Armada et al., 2017).

DIFFUSION Diffusion algorithms have been extensively used in computational biology. For instance, HotNet is an algorithm for finding sub-networks with a large amount of mutated genes (Vandin et al., 2011), whereas TieDIE attemps to link a source set and a target set of molecular entities through two diffusion processes (Paull et al., 2013). Other applications include the prioritisation of disease genes (Lee et al., 2011) and the prediction of gene function (Mostafavi et al., 2008).

In *FELLA*, diffusion is a natural way to score all the nodes in the KEGG graph given an input list of metabolites, available using `method = "diffusion"` in the function `runDiffusion`. The input metabolites introduce unitary flow in the network. Flow can only leave the network through pathway nodes, forcing it to propagate through the intermediate entities as well (reactions, enzymes and modules), see figure 88. Further details can found in (Picart-Armada et al., 2017).

However, the diffusion scores are biased due to the network topology (Picart-Armada et al., 2017) and therefore a normalisation step is required. *FELLA* offers a normalisation through a z-score (`approx = "normality"`) or through an empirical p-value (`approx = "simulation"`), both assessing whether the diffusion score of a node is likely to be reached in a permutation analysis, i.e. if the input is random.

The normalisation through the z-scores leads to p-scores, defined as:

$$ps_i = 1 - \Phi(z_i)$$

Where ps_i is the p-score of node i, z_i is its z-score (Picart-Armada et al., 2017) and Φ is the cumulative distribution function of the standard gaussian distribution. Under this definition, nodes are ranked using increasing p-scores.

For completeness, two alternative parametric scores have been added. The heavier-tailed **t**-distribution can be used instead of the gaussian by choosing `approx = "t"` and supplying the desired degrees of freedom ν.

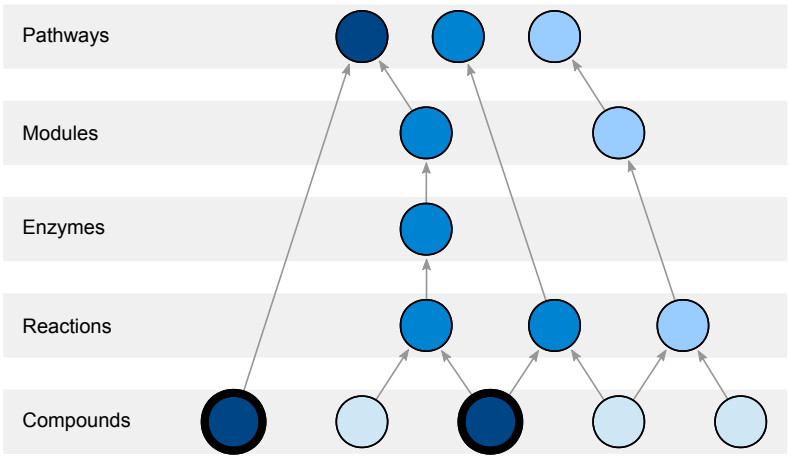


Figure 89: Network setup for PageRank. Input metabolites (in black rings) are the source of random walks that must climb through the graph levels, up to the pathway nodes. Figure extractred from (Picart-Armada et al., 2017).

Similarly, the **gamma** distribution can be used through `approx = "gamma"`. The p-score is obtained with

$$ps_i = 1 - F_i(T_i)$$

Being T_i the raw temperature of node i and F_i the cumulative distribution function of a gamma distribution, adjusted by its shape ($\frac{\mu_i^2}{\sigma_i^2}$) and scale ($\frac{\sigma_i^2}{\mu_i}$) parameters. The quantities μ_i and σ_i^2 are the mean and variance of the null temperatures and are analytically known from the null model formulation (Picart-Armada et al., 2017).

PAGERANK PageRank (Page et al., 1999) offers a scoring method for the nodes in the KEGG graph, based on a random walks approach. The random walks start at the input metabolites and are forced to explore their reachable nodes, see figure 89. As random walks take into account the direction of the edges, PageRank is applied to the upwards-directed KEGG graph (figure 87) in order to force the walks to reach pathway nodes. Nodes that are frequently visited by the random walks earn a higher PageRank, analogously to the diffusion scores. More details about this particular formulation, implemented in `runPagerank`, can be found in (Picart-Armada et al., 2017).

The PageRank scores are statistically normalised, providing the same options as in the diffusion scores in section D.2.3. Therefore, the argument `approx` can be set to `"simulation"` for the permutation analysis, or to `"normality"`, `"t"` or `"gamma"` for the parametric alternatives.

Enrichment wrapper

FELLA contains the wrapper `enrich` that maps the KEGG ids and runs the desired enrichment procedure with a single call. This can be convenient for producing compact scripts and running quick analyses.

Limitations

FELLA currently starts the statistical analysis from a list of affected metabolites. Therefore, it inherits a limitation from ORA methods: the need of choosing a cutoff to derive the list of affected metabolites, assuming that the metabolites stem from a differential abundance analysis.

Another limitation, shared among network-based models, is the incomplete biological knowledge from which the network is built. The knowledge model in *FELLA* might also constraint the complexity of the mechanisms that can be found through it. Processes such as genetic and epigenetic events, or the type and directionality of regulatory events, are not considered at the moment.

The user should be aware that *FELLA* neither builds a dynamic model of the biochemical reactions in the metabolism, nor relies on flux balance analysis. Conversely, *FELLA* is built on a knowledge representation from the biology in KEGG that focuses on offering interpretability to the final user.

D.2.4 Case studies

The functionalities of *FELLA* are demonstrated by (1) building a Homo sapiens database and (2) enriching summary metabolomics data from three public datasets.

Building the database

FELLA requires a database built from KEGG to perform any data enrichment. *FELLA* contains a small example database as a *FELLA.DATA* object, accessible via `data("FELLA.sample")`, but this is a toy example for demonstration purposes, not suited for regular analyses.

Therefore, the database for the corresponding organism has to be built before any analysis is run. The first step is to build the KEGG graph from the current KEGG release with the function `buildGraphFromKEGGREST`. Note that the user can force specific KEGG pathways to be excluded from the graph - the following code removes "overview" metabolic pathways based on KEGG brite.

```
library(FELLA)
set.seed(1)
# Filter overview pathways
graph <- buildGraphFromKEGGREST(
    organism = "hsa",
    filter.path = c("01100", "01200", "01210", "01212", "01230"))

## Building through KEGGREST...
## Available gene annotations:  ncbi-geneid, ncbi-proteinid.  Using
ncbi-geneid
## Done.
## Building graph...
```

```
## Filtering 5 pathways.
## Done.
## Pruning graph...
## Current weight:  1 out of 4...
## Current weight:  2 out of 4...
## Current weight:  3 out of 4...
## Current weight:  4 out of 4...
## Done.
```

Once the KEGG graph is ready, the database will be saved locally using `buildDataFromGraph`. The user can choose which matrices shall be stored using the `matrices` argument - saving both `"diffusion"` and `"pagerank"` might take up to 1GB of disk space.

If the user plans on using the z-score approximation, it is advisable to set the `normality` argument to `c("diffusion", "pagerank")` in order to speed up future computations. Using the z-scores with a custom metabolite background will require the matrices to be saved as well.

Finally, the argument `niter` controls how many random trials are performed in the estimation of the null distribution of the largest connected component of a k-th order random subgraph. As this is a property of the KEGG graph, it is performed once and reused in each analysis. This finds application when filtering small connected components from the reported sub-network, see section D.2.4.

```
tmpdir <- paste0(tempdir(), "/my_database")
# Mke sure the database does not exist from a former vignette build
# Otherwise the vignette will rise an error
# because FELLA will not overwrite an existing database
unlink(tmpdir, recursive = TRUE)
buildDataFromGraph(
    keggdata.graph = graph,
    databaseDir = tmpdir,
    internalDir = FALSE,
    matrices = "diffusion",
    normality = "diffusion",
    niter = 50)

## Computing probabilities for random subgraphs...  (this may take a
while)
## Directory /tmp/RtmpAOhDlw/my_database does not exist.  Creating it...
## Done.
## Done.
## Computing diffusion.matrix...  (this may take a while and use some
memory)
## Done
## Computing diffusion.rowSums...
## Done.
```

When the database is available in local, it can be loaded in an *R* session and assigned to a *FELLA.DATA* object using the function `loadKEGGdata`. This

should be the only procedure for creating any *FELLA.DATA* object. The user is given the choice of loading the diffusion and pagerank matrices to ease memory saving.

```
fella.data <- loadKEGGdata(
    databaseDir = tmpdir,
    internalDir = FALSE,
    loadMatrix = "diffusion"
)

## Loading KEGG graph data...
## Done.
## Loading hypergeom data...
## Loading matrix...
## 'hypergeom.matrix.RData' not present in:/tmp/RtmpAOhDlw/my_database/hypergeom.matrix.RDat
Hypergeometric test won't execute.
## Done.
## Loading diffusion data...
## Loading matrix...
## Done.
## Loading rowSums...
## Done.
## Loading pagerank data...
## Loading matrix...
## 'pagerank.matrix.RData' not loaded.  Simulated permutations may execute
slower for pagerank.
## Done.
## Loading rowSums...
## 'pagerank.rowSums.RData' not present in:/tmp/RtmpAOhDlw/my_database/pagerank.rowSums.RDat
Z-scores won't be available for pagerank.
## Done.
## Data successfully loaded.
```

The contents of the *FELLA.DATA* object can be summarised as well:

```
fella.data

## General data:
## - KEGG graph:
##   * Nodes:  11115
##   * Edges:  34787
##   * Density:  0.0002816029
##   * Categories:
##     + pathway [327]
##     + module [173]
##     + enzyme [1149]
##     + reaction [5467]
##     + compound [3999]
##   * Size:  6.2 Mb
## - KEGG names are ready.
```

```
## ----------------------------
## Hypergeometric test:
## - Matrix not loaded.
## ----------------------------
## Heat diffusion:
## - Matrix is ready
##   * Dim:  11115 x 3999
##   * Size:  340.1 Mb
## - RowSums are ready.
## ----------------------------
## PageRank:
## - Matrix not loaded.
## - RowSums not loaded.
```

The function `getInfo` provides the KEGG release and organism that generated a *FELLA.DATA* object:

```
cat(getInfo(fella.data))

## T01001           Homo sapiens (human) KEGG Genes Database
## hsa              Release 93.0+/02-22, Feb 20
##                  Kanehisa Laboratories
##                  22,498 entries
##
## linked db        pathway
##                  brite
##                  module
##                  ko
##                  genome
##                  enzyme
##                  network
##                  disease
##                  drug
##                  ncbi-geneid
##                  ncbi-proteinid
##                  uniprot
```

Please note that the database built for this vignette is stored in a temporary folder and will not be persistent. The user should build his or her own database and save it in a persistent location, either in the package installation directory (`internalDir = TRUE`) or in a custom folder (`internalDir = FALSE`). Internal databases can be listed using `listInternalDatabases`.

A cautionary note if the user is relying on the internal directory: reinstalling *FELLA* will wipe existent databases because its internal directory is overwritten. Also, if the database name already exists when saving a new database, the existing database will be renamed by appending `_old` in order to avoid overwriting.

Epithelial cells dataset

This example data is extracted from the epithelial cancer cells dataset (Chen et al., 2015), an in vitro model of dry eye in which the human epithelial cells IOBA-NHC are put under hyperosmotic stress. The original study files are deposited in the Metabolights repository (Haug et al., 2012) under the identifier MTBLS214: https://www.ebi.ac.uk/metabolights/MTBLS214. The list of metabolites hereby used reflects metabolic changes in "Treatment 1" (24 hours in serum-free media at 380 mOsm) against control (24 hours at 280 mOsm). The metabolites have been extracted from "Table 1" in the original manuscript and mapped to KEGG ids.

MAPPING THE INPUT METABOLITES The input metabolites should be provided as KEGG compound identifiers. If the user starts from another source (common names, HMDB identifiers), tools like the "compound ID conversor" from MetaboAnalyst can be useful for the ID conversion.

```
compounds.epithelial <- c(
    "C02862", "C00487", "C00025", "C00064",
    "C00670", "C00073", "C00588", "C00082", "C00043")
```

The first step is to map the input metabolites to the KEGG graph with `defineCompounds`. This step requires the *FELLA.DATA* object, loaded in section D.2.4. The user can impose a custom metabolite background with the `compoundsBackground` argument. By default, all the KEGG compounds in the graph are used.

```
analysis.epithelial <- defineCompounds(
    compounds = compounds.epithelial,
    data = fella.data)

## No background compounds specified.  Default background will be used.
## Warning in defineCompounds(compounds = compounds.epithelial, data
= fella.data):  Some compounds were introduced as affected but they
do not belong to the background.  These compounds will be excluded from
the analysis.  Use 'getExcluded' to see them.
```

Notice that `defineCompounds` throws a warning if any of the input metabolites does not map to the graph. The user can retrieve the mapped and unmapped identifiers through `getInput` and `getExcluded`, respectively.

```
getInput(analysis.epithelial)

## [1] "C00025" "C00043" "C00064" "C00073" "C00082" "C00487" "C00588" "C00670"

getExcluded(analysis.epithelial)

## [1] "C02862"
```

The status of a *FELLA.USER* object can be checked by printing the object.

```
analysis.epithelial

## Compounds in the input: 8
## [1] "C00025" "C00043" "C00064" "C00073" "C00082" "C00487" "C00588" "C00670"
## Background compounds: all available compounds (default)
## ----------------------------
## Hypergeometric test: not performed
## ----------------------------
## Heat diffusion: not performed
## ----------------------------
## PageRank: not performed
```

ENRICHING USING DIFFUSION Having mapped the compounds, the enrichment can be performed. In this vignette, only the diffusion method in `runDiffusion` will be applied, although PageRank has an almost identical usage in `runPagerank`.

If the user prefers an explicit permutation analysis, the option `approx = "simulation"` performs the amount of iterations specified in the `niter` argument.

Conversely, if the desired approximation is the z-score (`approx = "normality"`), the process does not require permutations. The z-scores are converted to `p.scores` using the `pnorm` routine. Likewise, `approx = "t"` and `approx = "gamma"` respectively rely on `pt` and `pgamma`. Section D.2.3 contains further details on the scores.

This example applies `approx = "normality"`, a fast option. For a comparison between prioritisations using Monte Carlo trials or the parametric z-score, the user can is referred to (Picart-Armada et al., 2017).

```
analysis.epithelial <- runDiffusion(
    object = analysis.epithelial,
    data = fella.data,
    approx = "normality")

## Running diffusion...
## Computing p-scores through the specified distribution.
## Done.
```

The *FELLA.USER* object has been updated with the `p.scores` from the diffusion results:

```
analysis.epithelial

## Compounds in the input: 8
## [1] "C00025" "C00043" "C00064" "C00073" "C00082" "C00487" "C00588" "C00670"
## Background compounds: all available compounds (default)
## ----------------------------
## Hypergeometric test: not performed
## ----------------------------
## Heat diffusion: ready.
```

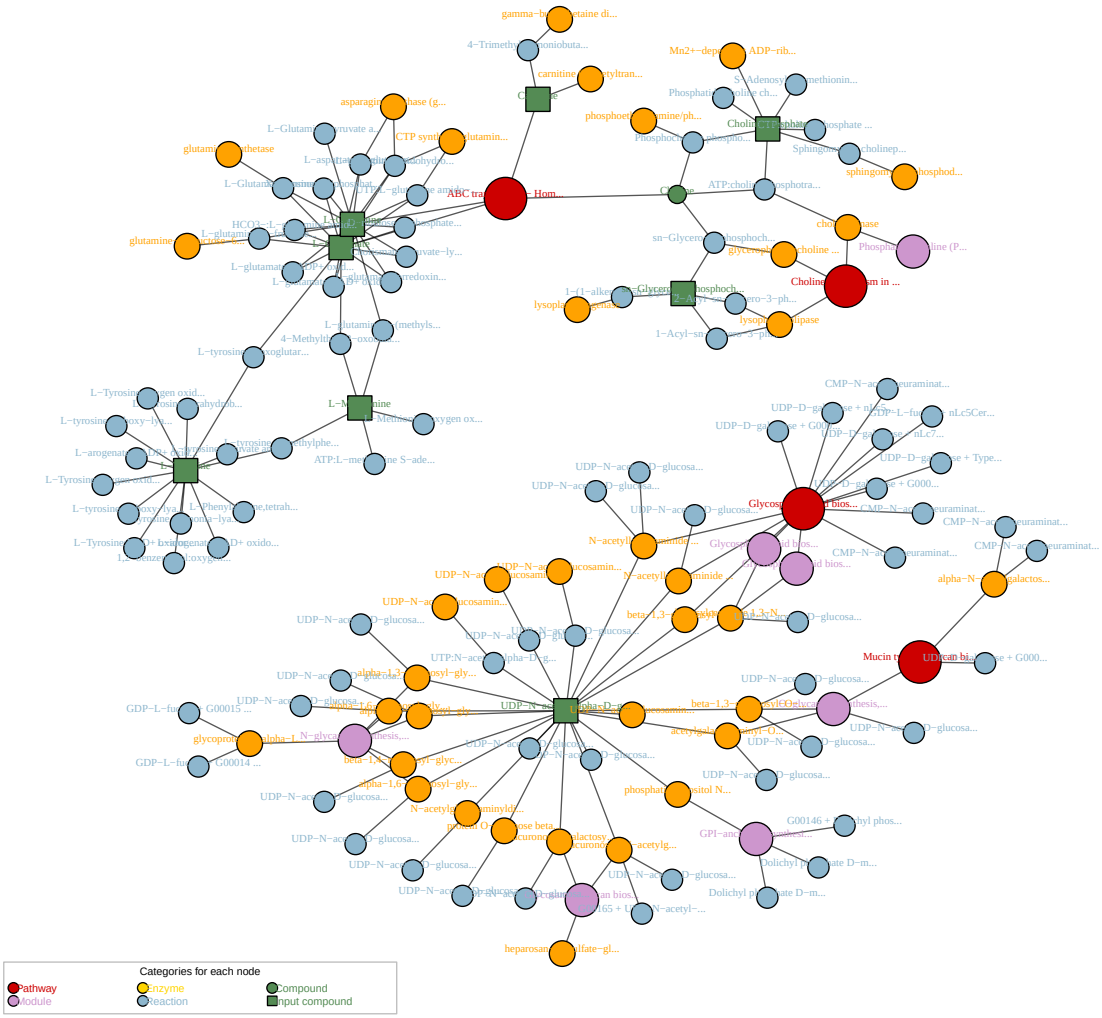
```
## P-scores under 0.05:  282
## ----------------------------
## PageRank: not performed
```

At this point, the subgraph consisting of top scoring nodes can be plotted in a heterogeneous network layout. In the presence of signal, this subgraph will exhibit large connected components and contain nodes from all the levels in the KEGG graph. It is also expected that the algorithm gives a high priority to the metabolites specified in the input, although not all of them must necessarily be top ranked.

Therefore, the user should expect to find the presence of intermediate entities (reactions, enzymes and modules) that connect the input to relevant KEGG pathways. Note that *FELLA* can also pinpoint new KEGG compounds as potentially relevant.

In this example, the plot is limited to 150 nodes using the `nlimit` argument from `plot`.

```
nlimit <- 150
vertex.label.cex <- .5
plot(
    analysis.epithelial,
    method = "diffusion",
    data = fella.data,
    nlimit = nlimit,
    vertex.label.cex = vertex.label.cex)

## 282 nodes below the threshold have been limited to 150 nodes.
```



In the original work (Chen et al., 2015), the activation of the **glycerophosphocholine synthesis** rather than the **carnitine** response is a main result. *FELLA* highlights[1] the related pathway *choline metabolism in cancer* and the *choline* metabolite as well. Another key process is the **O-linked glycosilation**, which is close to the KEGG module *O-glycan biosynthesis, mucin type core* and to the KEGG pathway *Mucin type O-glycan biosynthesis*. Finally, *FELLA* reproduces the finding of **UAP1** by reporting the enzyme *2.7.7.23*, named *UDP-N-acetylglucosamine diphosphorylase*. **UAP1** is a key protein in the study, pinpointed by iTRAQ and validated via western blot.

EXPORTING THE RESULTS After an initial exploration of the results, these can be exported using three functions that lead to network and tabular formats.

The top scoring nodes can be exported as a network in *igraph* with the function `generateResultsGraph`. The number `k` of nodes in the subgraph is controlled by the most stringent filter between `nlimit` (limit on the number of nodes) and `threshold` (limit on the `p.score`).

Once `k` is determined, the argument `thresholdConnectedComponent` further filters small connected components from the subgraph, implying that the resulting subgraph can have less than `k` nodes. A connected component

1 This analysis is subject to KEGG release 83.0, from August 17th, 2017. Posterior KEGG releases might alter the reported sub-network

of order r will be kept only if the probability that a random subgraph of order k contains a connected component of order at least r is smaller than the specified threshold. In other words, small connected components can arise from random sampling of the subgraph, whereas larger connected components are highly unlikely under a uniform sampling. The user can filter connected components that are too small to be meaningful in that sense.

Lastly, the argument `LabelLengthAtPlot` allows to truncate the KEGG names at the given number of characters for visualisation purposes.

```
g <- generateResultsGraph(
    object = analysis.epithelial,
    method = "diffusion",
    nlimit = nlimit,
    data = fella.data)

## 282 nodes below the threshold have been limited to 150 nodes.

g

## IGRAPH 54dfa2d UNW- 138 166 --
## + attr: organism (g/c), name (v/c), com (v/n), NAME (v/x), entrez
## | (v/x), label (v/c), input (v/l), weight (e/n)
## + edges from 54dfa2d (vertex names):
##  [1] hsa00512--M00056    hsa00601--M00070    hsa00601--M00071
##  [4] M00056  --2.4.1.102 M00075  --2.4.1.143 M00075  --2.4.1.144
##  [7] M00075  --2.4.1.145 hsa00601--2.4.1.146 M00056  --2.4.1.147
## [10] hsa00601--2.4.1.149 hsa00601--2.4.1.150 M00075  --2.4.1.155
## [13] M00065  --2.4.1.198 M00075  --2.4.1.201 M00070  --2.4.1.206
## [16] M00071  --2.4.1.206 M00059  --2.4.1.223 M00059  --2.4.1.224
## [19] M00075  --2.4.1.68  hsa00512--2.4.99.3  hsa05231--2.7.1.32
## + ... omitted several edges
```

The exported (sub)graph can be further complemented with data from GO, the Gene Ontology (Consortium, 2015). Specifically, the enzymes can be equipped with annotations from their underlying genes in any ontology from GO. Note that this requires additional packages: *biomaRt* and *org.Hs.eg.db*. The latter should be changed in case the analysis and the database are not from Homo sapiens.

The function `addGOToGraph` achieves this by accepting a query GO term and computing the semantic similarity of all the genes within each enzyme to the query GO term. The semantic similarity is detailed and implemented in the package *GOSemSim* (Yu, Fei Li, et al., 2010).

In the current example, enzymes are going to be compared to the GO cellular component term mitochondrion. Enzymes that contain genes whose cellular component is closer or coincident with the mitochondrion will be highlighted.

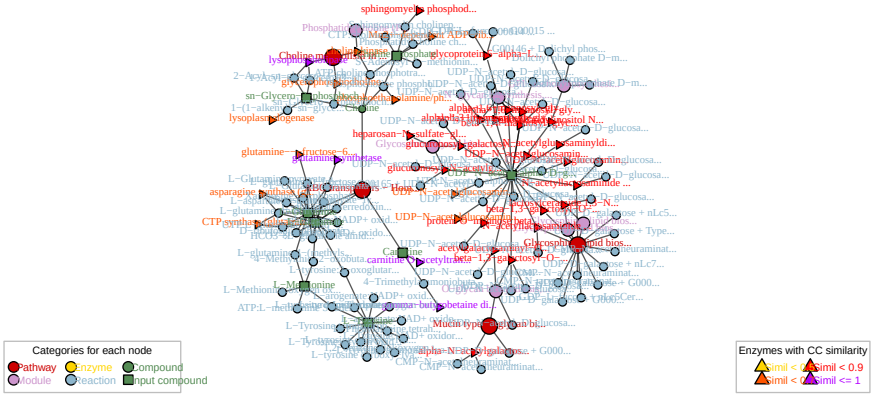
```
# GO:0005739 is the term for mitochondrion
g.go <- addGOToGraph(
    graph = g,
```

```
    GOterm = "GO:0005739",
    godata.options = list(
        OrgDb = "org.Hs.eg.db", ont = "CC"),
    mart.options = list(
        biomart = "ensembl", dataset = "hsapiens_gene_ensembl"))

##
## Loading required package:  org.Hs.eg.db
## Loading required package:  AnnotationDbi
## Loading required package:  stats4
## Loading required package:  BiocGenerics
## Loading required package:  parallel
##
## Attaching package:  'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##    clusterExport, clusterMap, parApply, parCapply, parLapply,
##    parLapplyLB, parRapply, parSapply, parSapplyLB
## The following objects are masked from 'package:stats':
##
##    IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##    anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##    dirname, do.call, duplicated, eval, evalq, Filter, Find, get,
##    grep, grepl, intersect, is.unsorted, lapply, Map, mapply, match,
##    mget, order, paste, pmax, pmax.int, pmin, pmin.int, Position,
##    rank, rbind, Reduce, rownames, sapply, setdiff, sort, table,
##    tapply, union, unique, unsplit, which, which.max, which.min
## Loading required package:  Biobase
## Welcome to Bioconductor
##
##    Vignettes contain introductory material; view with
##    'browseVignettes()'.  To cite Bioconductor, see
##    'citation("Biobase")', and for packages 'citation("pkgname")'.
## Loading required package:  IRanges
## Loading required package:  S4Vectors
##
## Attaching package:  'S4Vectors'
## The following object is masked from 'package:base':
##
##    expand.grid
##
## preparing gene to GO mapping data...
## preparing IC data...

g.go
```

```
## IGRAPH 54dfa2d UNW- 138 166 --
## + attr: organism (g/c), name (v/c), com (v/n), NAME (v/x), entrez
## | (v/x), label (v/c), input (v/l), GO (v/x), GO.simil (v/x), weight
## | (e/n)
## + edges from 54dfa2d (vertex names):
##  [1] hsa00512--M00056    hsa00601--M00070    hsa00601--M00071
##  [4] M00056  --2.4.1.102 M00075  --2.4.1.143 M00075  --2.4.1.144
##  [7] M00075  --2.4.1.145 hsa00601--2.4.1.146 M00056  --2.4.1.147
## [10] hsa00601--2.4.1.149 hsa00601--2.4.1.150 M00075  --2.4.1.155
## [13] M00065  --2.4.1.198 M00075  --2.4.1.201 M00070  --2.4.1.206
## [16] M00071  --2.4.1.206 M00059  --2.4.1.223 M00059  --2.4.1.224
## + ... omitted several edges
```

Plotting the graph with the function `plotGraph` reveals the addition of the GO term due to a slight change in the plotting legend. Enzyme nodes have a different shape and their colour scale reflects their degree of similarity to the queried GO term.

```
plotGraph(
    g.go,
    vertex.label.cex = vertex.label.cex)
```



The second way to export the enrichment results is to write the data from the KEGG entries in the top k p.scores using `generateResultsTable`. This function accepts arguments similar to those in `generateResultsTable`.

```
tab.all <- generateResultsTable(
    method = "diffusion",
    nlimit = 100,
    object = analysis.epithelial,
    data = fella.data)

## Writing diffusion results...
## Done.

# Show head of the table
knitr::kable(head(tab.all), format = "latex")
```

KEGG.id	Entry.type	KEGG.name	p.score
hsa00512	pathway	Mucin type O-glycan biosynthesis - Homo sapie...	3.7e-06
hsa05231	pathway	Choline metabolism in cancer - Homo sapiens (...	1.0e-06
M00056	module	O-glycan biosynthesis, mucin type core	1.0e-06
M00059	module	Glycosaminoglycan biosynthesis, heparan sulfa...	1.0e-06
M00075	module	N-glycan biosynthesis, complex type	1.0e-06
1.14.11.1	enzyme	gamma-butyrobetaine dioxygenase	1.0e-06

The last exporting option, `generateEnzymesTable`, is to a tabular format with details from the enzymes reported among the top k KEGG entries. In particular, the table contains the genes that belong to each enzyme family, separated by semicolons.

```
tab.enzyme <- generateEnzymesTable(
    method = "diffusion",
    nlimit = 100,
    object = analysis.epithelial,
    data = fella.data)

## Writing diffusion enzymes...
## Batch submitting query [=======>-------------]  29% eta:  21s Batch
submitting query [===========>-----------]  43% eta:  20s Batch submitting
query [===============>--------]  57% eta:  15s Batch submitting query
[===================>-----]  71% eta:  10s Batch submitting query [==============
86% eta:  5s Batch submitting query [============================] 100%
eta:  0s                                                                Null
GOterm provided to addGOToGraph.  Only the GO labels will be added.
To include similarity values as well, please specify a GOterm
## Done.

# Show head of the table
knitr::kable(head(tab.enzyme, 10), format = "latex")
```

EC_number	p.score	EC_name	Genes
2.3.1.7	1e-06	carnitine O-acetyltransferase	1384
1.14.11.1	1e-06	gamma-butyrobetaine dioxygenase	8424
3.1.4.2	1e-06	glycerophosphocholine phosphodiesterase	56261
3.1.3.75	1e-06	phosphoethanolamine/phosphocholine phosphatas...	162466
3.6.1.53	1e-06	Mn2+-dependent ADP-ribose/CDP-alcohol diphosp...	56985
3.1.4.12	1e-06	sphingomyelin phosphodiesterase	339221;55
2.7.1.32	1e-06	choline kinase	1119;1120
2.4.1.146	1e-06	beta-1,3-galactosyl-O-glycosyl-glycoprotein b...	10331
2.4.1.150	1e-06	N-acetyllactosaminide beta-1,6-N-acetylglucos...	2651
3.1.1.5	1e-06	lysophospholipase	10434;109

The three exporting options shown above are included in the wrapper function `exportResults`, using `format = "csv"` for the general tabular data, `format = "enzyme"` for the enzyme tabular data and `format = "igraph"` for saving an `.RData` object with the igraph sub-network object.

For instance, the general tabular data:

```
tmpfile <- tempfile()
exportResults(
    format = "csv",
    file = tmpfile,
    method = "diffusion",
    object = analysis.epithelial,
    data = fella.data)

## Exporting to a csv file...
## Writing diffusion results...
## Done.
## Done
```

If the argument `format` is none of the former, *FELLA* saves the sub-network using `write.graph` from the *igraph* package with the desired format.

```
tmpfile <- tempfile()
exportResults(
    format = "pajek",
    file = tmpfile,
    method = "diffusion",
    object = analysis.epithelial,
    data = fella.data)

## 282 nodes below the threshold have been limited to 250 nodes.
## Exporting to the format pajek using igraph...
## Done
```

DEPLOYING THE GRAPHICAL USER INTERFACE *FELLA* is equipped with a graphical user interface that eases data analysis without learning the package syntax. The app is divided in the following tabs:

- Compounds upload (figure 90): contains a general description of the tabs and a handle to submit the input metabolite list as a text file. Examples are provided as well. The right panel shows the mapped and the mismatching compounds with regard to the default database.
- Advanced options (figure 91): widgets that contain the main function arguments for customising the enrichment procedure. Allows database choice from the internal package directory, method and approximation choice and parameter tweaking. It also allows defining a GO label for the semantic similarity analysis on the reported enzymes.
- Results (figure 92): interactive plot with the sub-graph with the top `k` KEGG entries. Nodes can be selected, queried and link to the KEGG entries when hovered. Below the network lies an interactive table with the graph nodes, allowing the user to look into particular entries.
- Export (figure 93): several tabular and network exporting options.

The app is based on *shiny* (Chang et al., 2017) and can be launched through `launchApp`.

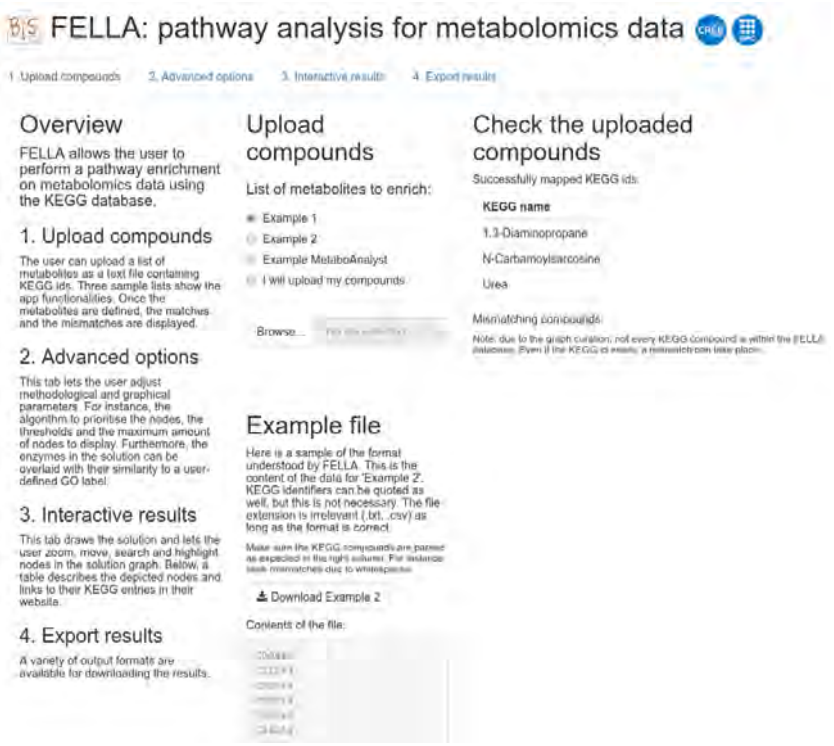


Figure 90: Graphical interface: compounds upload

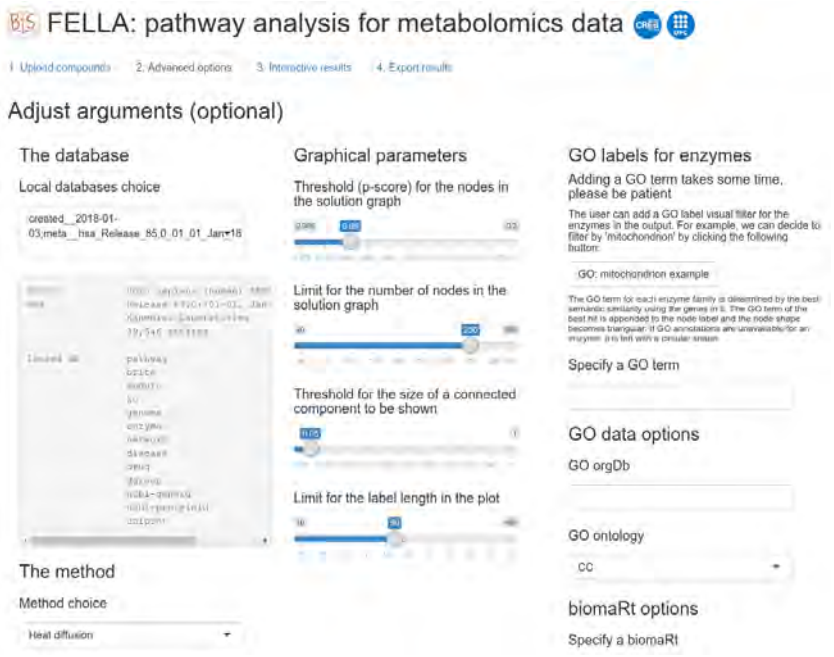


Figure 91: Graphical interface: advanced options

HELPER FUNCTIONS *FELLA* is equipped with helper functions that ease the user experience and avoid direct manipulation of the S4 classes. Some of them have been already introduced - a complete enumeration of the exported functions is hereby provided.

Functions of the type `get-` ease object and slot retrieval, with the following possibilities: `getBackground`, `getExcluded`, `getInfo`, `getInput`, `getName`, `getPscores`.

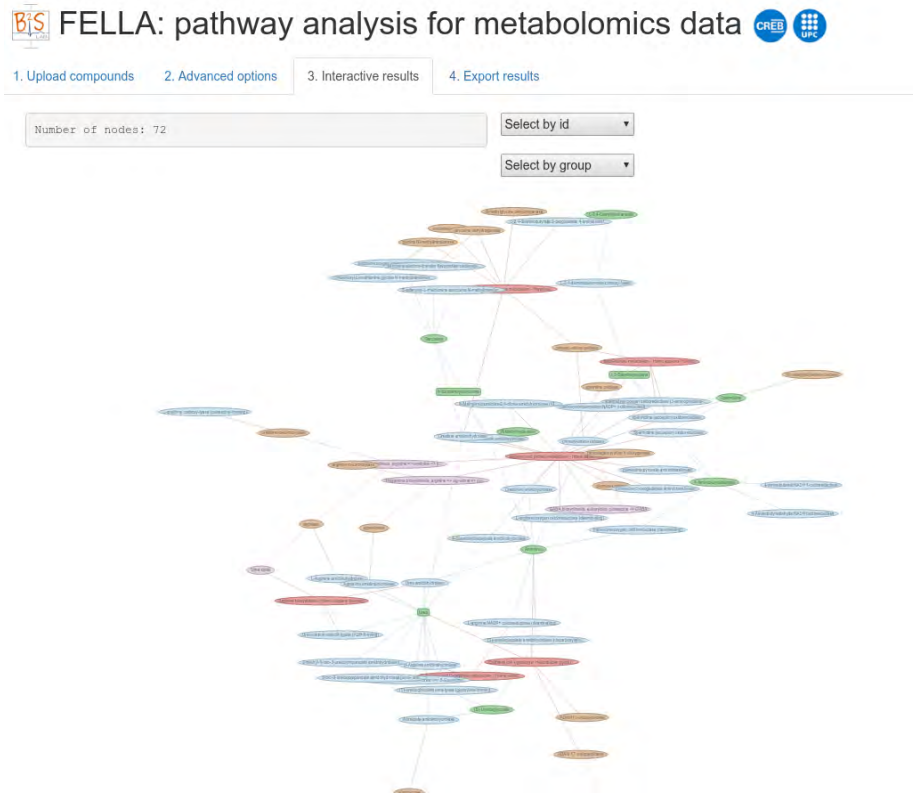


Figure 92: Graphical interface: results

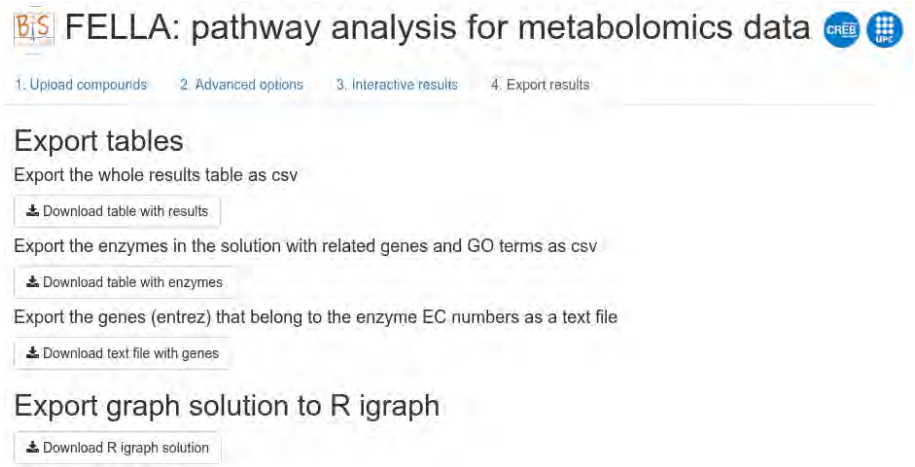


Figure 93: Graphical interface: export

On the other hand, functions starting by `list-` provide general purpose data about the package (`listMethods`, `listApprox`, `listCategories`) and a listing of the available internal databases (`listInternalDatabases`).

Finally, functions starting by `is-` check if an object belongs to a certain class: `is.FELLA.DATA` and `is.FELLA.USER`.

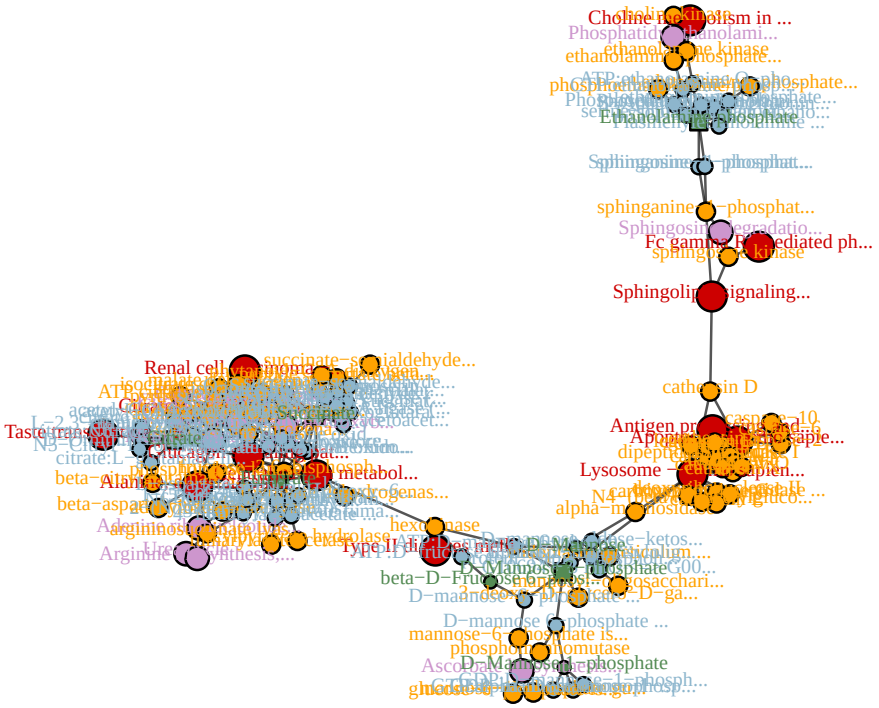
Ovarian cancer cells dataset

The next example has been extracted from the study on metabolic responses of ovarian cancer cells (Vermeersch et al., 2014). The original files

can be found in the MTBLS150 study in the Metabolights respository: https://www.ebi.ac.uk/metabolights/MTBLS150. OCSCs are isogenic ovarian cancer stem cells derived from the OVCAR-3 ovarian cancer cells. The abundances of six metabolites are affected by the exposure to several environmental conditions: glucose deprivation, hypoxia and ischemia (column "All" in "Figure 3" from their main manuscript).

The common names have been converted to KEGG ids prior to applying *FELLA*. The analysis is performed using the wrapper `enrich` that maps the compounds to the internal representation and runs the desired methods.

```
compounds.ovarian <- c(
    "C00275", "C00158", "C00042",
    "C00346", "C00122", "C06468")
analysis.ovarian <- enrich(
    compounds = compounds.ovarian,
    data = fella.data,
    methods = "diffusion")

## No background compounds specified.  Default background will be used.
## Warning in defineCompounds(compounds = compounds, compoundsBackground
= compoundsBackground, :  Some compounds were introduced as affected
but they do not belong to the background.  These compounds will be excluded
from the analysis.  Use 'getExcluded' to see them.
## Running diffusion...
## Computing p-scores through the specified distribution.
## Done.

plot(
    analysis.ovarian,
    method = "diffusion",
    data = fella.data,
    nlimit = 150,
    vertex.label.cex = vertex.label.cex,
    plotLegend = FALSE)

## 176 nodes below the threshold have been limited to 150 nodes.
```



The resulting subnetwork[2] reports several **TCA cycle**-related entities, also reported by the authors and by previous work (Pollard et al., 2003). It also mentions *sphingosine degradation*, closely related to the reported **sphingosine metabolism** in the original work. Enzymes that have been formerly related to cancer are suggested within the TCA cycle, like *fumarate hydratase* (Lehtonen et al., 2007; Pithukpakorn et al., 2006; Pollard et al., 2003) *succinate dehydrogenase* (Ni et al., 2008; Pollard et al., 2003) and *aconitase* (Singh et al., 2006). Another suggestion is *lysosome* - lysosomes suffer changes in cancer cells and directly affect apoptosis (Kirkegaard and Jäättelä, 2009). Finally, the graph contains several *hexokinases*, potential targets to disrupt glycolysis, a fundamental need in cancer cells (Kaelin and Thompson, 2010).

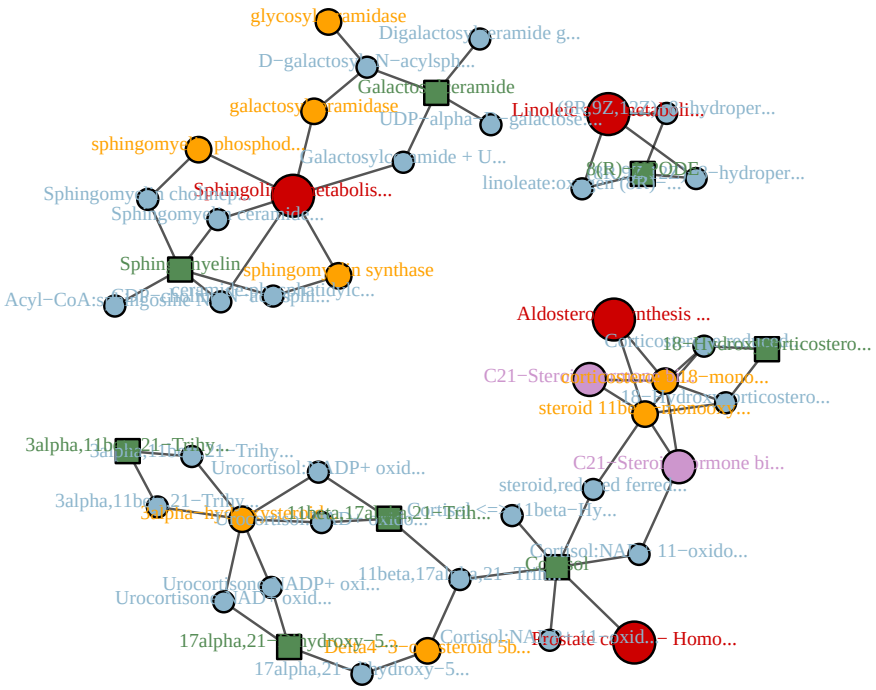
Malaria dataset

The metabolites in the last example are related to the distinction between malaria and other febrile ilnesses in (Decuypere et al., 2016). The study files can be found under the MTBLS315 identifier in Metabolights: `https://www.ebi.ac.uk/metabolights/MTBLS315`. Specifically, the list of KEGG identifiers has been extracted from the supplementary data spreadsheet, using all the possible KEGG matches for the "non malaria" patient group.

```
compounds.malaria <- c(
    "C05471", "C14831", "C02686", "C06462", "C00735", "C14833",
    "C18175", "C00550", "C01124", "C05474", "C05469")

analysis.malaria <- enrich(
    compounds = compounds.malaria,
    data = fella.data,
```

2 This analysis is subject to KEGG release 83.0, from August 17th, 2017. Posterior KEGG releases might alter the reported sub-network

```
    methods = "diffusion")

## No background compounds specified.  Default background will be used.
## Warning in defineCompounds(compounds = compounds, compoundsBackground
= compoundsBackground, :  Some compounds were introduced as affected
but they do not belong to the background.  These compounds will be excluded
from the analysis.  Use 'getExcluded' to see them.
## Running diffusion...
## Computing p-scores through the specified distribution.
## Done.

plot(
    analysis.malaria,
    method = "diffusion",
    data = fella.data,
    nlimit = 50,
    vertex.label.cex = vertex.label.cex,
    plotLegend = FALSE)

## 171 nodes below the threshold have been limited to 50 nodes.
```



In this case, the depicted subnetwork[3] contains the modules *C21-Steroid hormone biosynthesis, progesterone => corticosterone/aldosterone* and *C21-Steroid hormone biosynthesis, progesterone => cortisol/cortisone*, related to the **corticosteroids** as a main pathway reported in the original text. This is part of the also reported *Aldosterone synthesis and secretion*; aldosterone is known to show changes related to fever as a metabolic response to infection (Beisel, 1975). Another plausible hit in the sub-network is *linoleic acid metabolism*, as erythrocytes infected by various malaria parasytes can be enriched in linoleic

3 This analysis is subject to KEGG release 83.0, from August 17th, 2017. Posterior KEGG releases might alter the reported sub-network

acid (Fitch et al., 2000). In addition, the pathway *sphingolipid metabolism* can play a role in the immune response (Maceyka and Spiegel, 2014; Seo et al., 2011). As for the enzymes, *3alpha-hydroxysteroid 3-dehydrogenase (Si-specific)* and *Delta4-3-oxosteroid 5beta-reductase* are related to three input metabolites each and might be candidates for further examination.

D.2.5 Conclusions

The *FELLA* R package provides a simple, programmatic and intuitive enrichment tool for metabolomics summary data. Starting from a list of metabolites, *FELLA* not only pinpoints relevant pathways but also intermediate reactions, enzymes and modules that links the input metabolites to the pathways. The reported entries have a network structure focused on interpretability and new hypotheses generation, giving a richer perspective than classical pathway enrichment tools. This comprehensive layout can also suggest potential enzymes and new metabolites for further study. Finally, *FELLA* comes equipped with a graphical user interface that promotes its usage to a wider audience and offers interactive sub-network examination.

Funding

This work was supported by the Spanish Ministry of Economy and Competitiveness (MINECO) [BFU2014-57466-P to O.Y., TEC2014-60337-R and DPI2017-89827-R to A.P.]. O.Y., A.P. and S.P. thank for funding the Spanish Biomedical Research Centre in Diabetes and Associated Metabolic Disorders (CIBERDEM) and the Networking Biomedical Research Centre in the subject area of Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), both initiatives of Instituto de Investigación Carlos III (ISCIII). SP. thanks the AGAUR FI-scholarship programme.

D.2.6 Session info

Here is the output of `sessionInfo()` on the system that compiled this vignette:

- R version 3.6.2 (2019-12-12), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=es_ES.UTF-8`, `LC_COLLATE=en_US.UTF-8`, `LC_MONETARY=es_ES.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=es_ES.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=es_ES.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 16.04.6 LTS`
- Matrix products: default
- BLAS: `/usr/lib/atlas-base/atlas/libblas.so.3.0`
- LAPACK: `/usr/lib/atlas-base/atlas/liblapack.so.3.0`
- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.46.0, Biobase 2.44.0, BiocGenerics 0.29.2, FELLA 1.5.3, IRanges 2.17.5, knitr 1.22, org.Hs.eg.db 3.8.2, S4Vectors 0.21.24
- Loaded via a namespace (and not attached): assertthat 0.2.1, backports 1.1.4, BiocManager 1.30.4, BiocStyle 2.12.0, biomaRt 2.40.1, Biostrings 2.51.5, bit 1.1-14, bit64 0.9-7, bitops 1.0-6, blob 1.2.0, compiler 3.6.2, crayon 1.3.4, curl 3.3, DBI 1.0.0, digest 0.6.18, evaluate 0.13, GO.db 3.8.2, GOSemSim 2.10.0, grid 3.6.2, highr 0.8, hms 0.5.0, htmltools 0.3.6, httr 1.4.0, igraph 1.2.4.1, KEGGREST 1.24.1, lattice 0.20-38, magrittr 1.5, Matrix 1.2-18, memoise 1.1.0, pillar 1.4.0, pkgconfig 2.0.2, plyr 1.8.4, png 0.1-7, prettyunits 1.0.2, progress 1.2.2, R6 2.4.0, Rcpp 1.0.1, RCurl 1.95-4.12, rlang 0.4.0, rmarkdown 1.12, RSQLite 2.1.1, stringi 1.4.3, stringr 1.4.0, tcltk 3.6.2, tibble 2.1.1, tools 3.6.2, vctrs 0.2.0, xfun 0.6, XML 3.98-1.20, XVector 0.23.2, yaml 2.2.0, zeallot 0.1.0, zlibbioc 1.29.0

D.3 ADDITIONAL FILE 3: GILT-HEAD BREAM STUDY

D.3.1 Introduction

This vignette contains a case study of the effects of environmental contamination on gilt-head bream (*Sparus aurata*) (Ziarrusta et al., 2018). Fish were exposed over 14 days to *oxybenzone* and changes were sought in their brain, liver and plasma using untargeted metabolomics. Samples were processed using Ultra-performance liquid chromatography mass-spectrometry (UHPLC-qOrbitrap MS) in positive and negative modes with both C18 and HILIC separation.

The mortality of exposed fish was not altered, as well as the brain-related metabolites. However, liver and plasma showed perturbations, proving that adverse effects beyond the well-studied hormonal activity were present.

The enrichment procedure implemented in `FELLA` (Picart-Armada et al., 2017) was used in the study for a deeper understanding of the dysregulated metabolites in both tissues.

Building the database

At the time of publication, the KEGG database (Kanehisa, Furumichi, et al., 2016) –upon which `FELLA` is based– did not have pathway annotations for the *Sparus aurata* organism. It is common, however, to use the zebrafish (*Danio rerio*) pathways as a good approximation. KEGG provides pathway annotations for it under the organismal code `dre`, which will be used to build the `FELLA.DATA` object.

```
library(FELLA)

library(igraph)
library(magrittr)

set.seed(1)
# Filter the dre01100 overview pathway, as in the article
graph <- buildGraphFromKEGGREST(
    organism = "dre",
    filter.path = c("01100"))

tmpdir <- paste0(tempdir(), "/my_database")
# Make sure the database does not exist from a former vignette build
# Otherwise the vignette will rise an error
# because FELLA will not overwrite an existing database
unlink(tmpdir, recursive = TRUE)
buildDataFromGraph(
    keggdata.graph = graph,
    databaseDir = tmpdir,
    internalDir = FALSE,
    matrices = "none",
```

```
    normality = "diffusion",
    niter = 100)
```

We load the `FELLA.DATA` object to run both analyses:

```
fella.data <- loadKEGGdata(
    databaseDir = tmpdir,
    internalDir = FALSE,
    loadMatrix = "none"
)
```

Given the 11-month temporal gap between the study and this vignette, small changes to the amount of nodes in each category are expected (see section *2.4 Data handling and statistical analyses* from the study). Please see the Note on reproducibility to understand why.

```
fella.data
```

```
## General data:
## - KEGG graph:
##   * Nodes:  10821
##   * Edges:  32013
##   * Density:  0.0002734209
##   * Categories:
##     + pathway [163]
##     + module [171]
##     + enzyme [1021]
##     + reaction [5467]
##     + compound [3999]
##   * Size:  5.9 Mb
## - KEGG names are ready.
## ----------------------------
## Hypergeometric test:
## - Matrix not loaded.
## ----------------------------
## Heat diffusion:
## - Matrix not loaded.
## - RowSums are ready.
## ----------------------------
## PageRank:
## - Matrix not loaded.
## - RowSums not loaded.
```

Note on reproducibility

We want to emphasise that each time this vignette is built, `FELLA` constructs its `FELLA.DATA` object using the most recent version of the KEGG database. KEGG is frequently updated and therefore small changes can take place in the knowledge graph between different releases. The discussion on our findings was written at the date specified in the vignette header and using the KEGG release in the Reproducibility section.

D.3.2 Enrichment analysis on liver tissue

Defining the input and running the enrichment

Table 1 from the main body in (Ziarrusta et al., 2018) contains 5 KEGG identifiers associated to metabolic changes in liver tissue and 12 in plasma. Our first enrichment analysis with `FELLA` will be based on the liver-derived metabolites. Also note that we use the faster `approx = "normality"` approach, whereas the original article uses `approx = "simulation"` with `niter = 15000` This is not only intended to keep the bulding time of this vignette as low as possible, but also to demonstrate that the findings using both statistical approaches are consistent.

```
cpd.liver <- c(
    "C12623",
    "C01179",
    "C05350",
    "C05598",
    "C01586"
)

analysis.liver <- enrich(
    compounds = cpd.liver,
    data = fella.data,
    method = "diffusion",
    approx = "normality")
```

```
## No background compounds specified. Default background will be used.

## Running diffusion...

## Computing p-scores through the specified distribution.

## Done.
```

All the metabolites are successfully mapped:

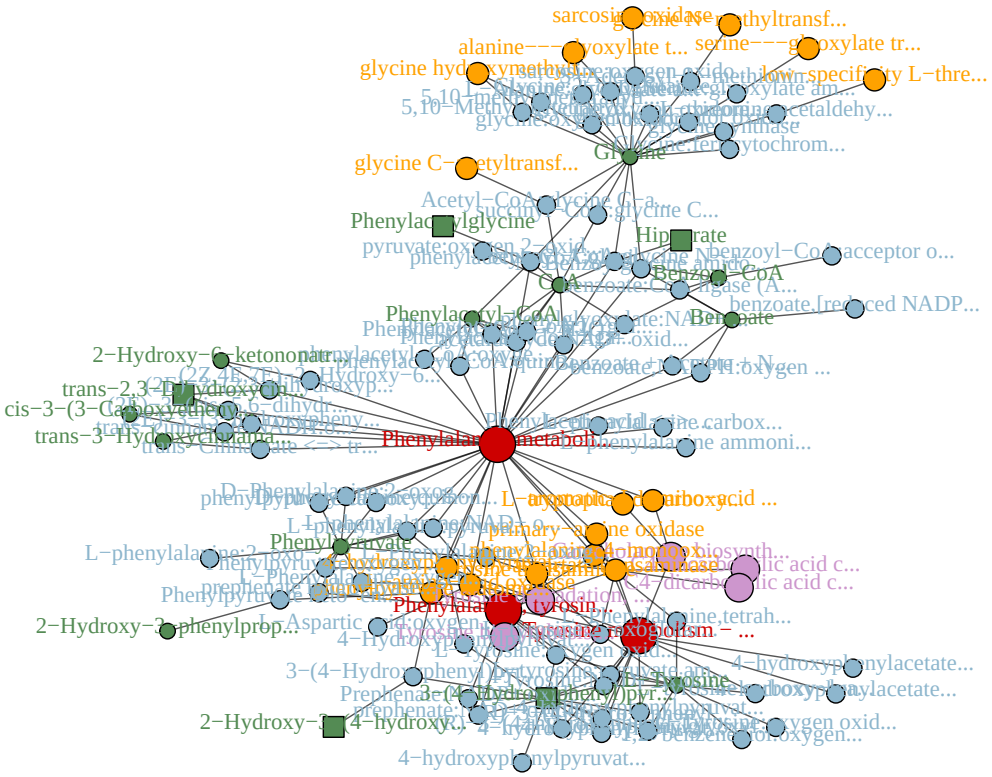
```
analysis.liver %>%
    getInput %>%
    getName(data = fella.data)
```

```
## $C12623
## [1] "trans-2,3-Dihydroxycinnamate"
## [2] "(2E)-3-(2,3-Dihydroxyphenyl)prop-2-enoate"
##
## $C01179
## [1] "3-(4-Hydroxyphenyl)pyruvate" "4-Hydroxyphenylpyruvate"
## [3] "p-Hydroxyphenylpyruvic acid"
##
## $C05350
```

```
## [1] "2-Hydroxy-3-(4-hydroxyphenyl)propenoate"
## [2] "4-Hydroxy-enol-phenylpyruvate"
##
## $C05598
## [1] "Phenylacetylglycine"
##
## $C01586
## [1] "Hippurate"                "Hippuric acid"
## [3] "N-Benzoylglycine"         "Benzoylaminoacetic acid"
```

Below is a plot of the reported sub-network using the default parameters. The five metabolites are present and lie within the same connected component.

```
plot(
    analysis.liver,
    method = "diffusion",
    data = fella.data,
    nlimit = 250,
    plotLegend = FALSE)
```



We will examine the igraph object with the reported sub-network and some of its reported entities in tabular format:

```
g.liver <-  generateResultsGraph(
    object = analysis.liver,
    data = fella.data,
    method = "diffusion")
```

```
tab.liver <- generateResultsTable(
    object = analysis.liver,
    data = fella.data,
    method = "diffusion")
```

```
## Writing diffusion results...
```

```
## Done.
```

The reported sub-network contains around 100 nodes and can be manually inquired:

```
g.liver
```

```
## IGRAPH 8327330 UNW- 112 181 --
## + attr: organism (g/c), name (v/c), com (v/n), NAME (v/x), entrez
## | (v/x), label (v/c), input (v/l), weight (e/n)
## + edges from 8327330 (vertex names):
##  [1] dre00400--M00025     dre00350--M00042     dre00350--M00044
##  [4] dre00360--1.13.11.27 M00044  --1.13.11.27 dre00360--1.14.16.1
##  [7] dre00400--1.14.16.1  dre00360--1.4.3.2    dre00400--1.4.3.2
## [10] M00044  --1.4.3.2    dre00350--1.4.3.21   dre00360--1.4.3.21
## [13] dre00350--2.6.1.1    dre00360--2.6.1.1    dre00400--2.6.1.1
## [16] M00170  --2.6.1.1    M00171  --2.6.1.1    dre00360--2.6.1.5
## [19] M00025  --2.6.1.5    M00044  --2.6.1.5    dre00360--4.1.1.105
## + ... omitted several edges
```

Examining the pathways

Figure 2 from the original study frames the five metabolites in the input around *Phenylalanine metabolism*. We can verify that FELLA finds such pathway and two closely related suggestions: *Tyrosine metabolism* and *Phenylalanine, tyrosine and tryptophan biosynthesis*.

```
path.fig2 <- "dre00360" # Phenylalanine metabolism
path.fig2 %in% V(g.liver)$name
```

```
## [1] TRUE
```

These are the reported pathways:

```
tab.liver[tab.liver$Entry.type == "pathway", ]
```

```
##    KEGG.id Entry.type                                        KEGG.name
## 1 dre00350    pathway    Tyrosine metabolism - Danio rerio (zebrafish)
## 2 dre00360    pathway Phenylalanine metabolism - Danio rerio (zebra...
## 3 dre00400    pathway Phenylalanine, tyrosine and tryptophan biosyn...
##        p.score
## 1 2.768611e-06
## 2 1.000000e-06
## 3 2.554160e-02
```

Examining the metabolites

Figure 2 also gathers two types of metabolites: metabolites in the input (inside shaded frames) and other contextual metabolites (no frames) that link the input metabolites.

First of all, we can check that all the input metabolites appear in the suggested sub-network. While it's expected that most of the input metabolites appear as relevant, it is an important property of our method, in order to elaborate a sensible biological justification of the experimental differences.

```
cpd.liver %in% V(g.liver)$name
```

```
## [1] TRUE TRUE TRUE TRUE TRUE
```

On the other hand, one of the two contextual metabolites is also suggested by FELLA, proving its usefulness to fill the gaps between the input metabolites.

```
cpd.fig2 <- c(
    "C00079", # Phenylalanine
    "C00082"  # Tyrosine
)
cpd.fig2 %in% V(g.liver)$name
```

```
## [1] FALSE  TRUE
```

D.3.3 Enrichment analysis on plasma

Defining the input and running the enrichment

As shown in section Defining the input and running the enrichment, 12 KEGG identifiers (one ID is repeated) are related to the experimental changes observed in plasma, which are the starting point of the enrichment:

```
cpd.plasma <- c(
    "C16323",
    "C00740",
    "C08323",
    "C00623",
    "C00093",
    "C06429",
    "C16533",
    "C00740",
    "C06426",
    "C06427",
    "C07289",
    "C01879"
) %>% unique
```

```
analysis.plasma <- enrich(
    compounds = cpd.plasma,
    data = fella.data,
    method = "diffusion",
    approx = "normality")
```

```
## No background compounds specified. Default background will be used.

## Running diffusion...

## Computing p-scores through the specified distribution.

## Done.
```

The totality of the 11 unique metabolites map to the `FELLA.DATA` object:

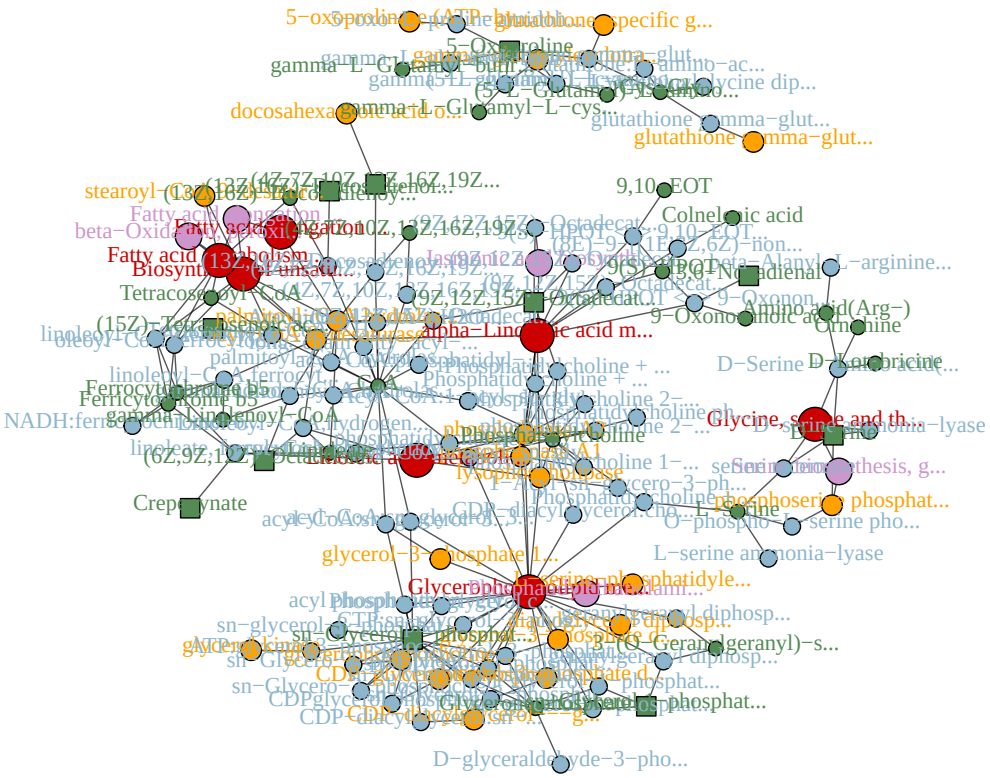
```
analysis.plasma %>%
    getInput %>%
    getName(data = fella.data)
```

```
## $C16323
## [1] "3,6-Nonadienal"
##
## $C00740
## [1] "D-Serine"
##
## $C08323
## [1] "(15Z)-Tetracosenoic acid"  "Nervonic acid"
## [3] "(Z)-15-Tetracosenoic acid"
##
## $C00623
## [1] "sn-Glycerol 1-phosphate" "sn-Gro-1-P"
## [3] "L-Glycerol 1-phosphate"
##
## $C00093
## [1] "sn-Glycerol 3-phosphate" "Glycerophosphoric acid"
## [3] "D-Glycerol 1-phosphate"
##
## $C06429
## [1] "(4Z,7Z,10Z,13Z,16Z,19Z)-Docosahexaenoic acid"
## [2] "4,7,10,13,16,19-Docosahexaenoic acid"
## [3] "Docosahexaenoic acid"
## [4] "Docosahexaenoate"
## [5] "4Z,7Z,10Z,13Z,16Z,19Z-Docosahexaenoic acid"
## [6] "(4Z,7Z,10Z,13Z,16Z,19Z)-Docosa-4,7,10,13,16,19-hexaenoic acid"
##
## $C16533
## [1] "(13Z,16Z)-Docosadienoic acid"
## [2] "(13Z,16Z)-Docosa-13,16-dienoic acid"
```

```
## [3] "13Z,16Z-Docosadienoic acid"
##
## $C06426
## [1] "(6Z,9Z,12Z)-Octadecatrienoic acid" "6,9,12-Octadecatrienoic acid"
## [3] "gamma-Linolenic acid"              "Gamolenic acid"
##
## $C06427
## [1] "(9Z,12Z,15Z)-Octadecatrienoic acid"
## [2] "alpha-Linolenic acid"
## [3] "9,12,15-Octadecatrienoic acid"
## [4] "Linolenate"
## [5] "alpha-Linolenate"
##
## $C07289
## [1] "Crepenynate"                   "(9Z)-Octadec-9-en-12-ynoate"
## [3] "(Z)-9-Octadecen-12-ynoic acid" "Crepenynic acid"
##
## $C01879
## [1] "5-Oxoproline"                  "Pidolic acid"
## [3] "Pyroglutamic acid"             "5-Pyrrolidone-2-carboxylic acid"
## [5] "Pyroglutamate"                 "5-Oxo-L-proline"
## [7] "L-Pyroglutamic acid"           "L-5-Pyrrolidone-2-carboxylic acid"
```

Again, the reported sub-network consists of a large connected component encompassing most input metabolites:

```
plot(
    analysis.plasma,
    method = "diffusion",
    data = fella.data,
    nlimit = 250,
    plotLegend = FALSE)
```



We will export the results as a network and as a table:

```
g.plasma <-  generateResultsGraph(
    object = analysis.plasma,
    data = fella.data,
    method = "diffusion")

tab.plasma <- generateResultsTable(
    object = analysis.plasma,
    data = fella.data,
    method = "diffusion")
```

```
## Writing diffusion results...

## Done.
```

The reported sub-network is a bit larger than the one from liver, containing roughly 120 nodes:

```
g.plasma
```

```
## IGRAPH 6b5f854 UNW- 137 216 --
## + attr: organism (g/c), name (v/c), com (v/n), NAME (v/x), entrez
## | (v/x), label (v/c), input (v/l), weight (e/n)
## + edges from 6b5f854 (vertex names):
##  [1] dre00260--M00020    dre00564--M00093    dre00592--M00113
##  [4] dre00062--M00415    dre01040--M00415    dre01212--M00415
##  [7] dre01040--M00861    dre01212--M00861    dre00564--1.1.1.8
## [10] dre01040--1.14.19.1 dre01212--1.14.19.1 dre00592--1.14.19.3
```

```
## [13] dre01040--1.14.19.3 dre01212--1.14.19.3 dre00564--1.1.5.3
## [16] dre00564--2.3.1.15  dre00564--2.7.8.29  dre00564--2.7.8.5
## [19] dre00564--3.1.1.32  dre00591--3.1.1.32  dre00592--3.1.1.32
## + ... omitted several edges
```

Examining the pathways

Figure 3 from the original study is a holistic view of the affected metabolites found in plasma, based on literature and on an analysis with FELLA. The 11 metabolites are depicted within their core metabolic pathways. We will check whether FELLA is able to highlight them, by first showing the reported metabolic pathways:

```
tab.plasma[tab.plasma$Entry.type == "pathway", ]
```

```
##    KEGG.id Entry.type                                          KEGG.name
## 1 dre00062    pathway Fatty acid elongation - Danio rerio (zebrafis...
## 2 dre00260    pathway Glycine, serine and threonine metabolism - Da...
## 3 dre00564    pathway Glycerophospholipid metabolism - Danio rerio ...
## 4 dre00591    pathway Linoleic acid metabolism - Danio rerio (zebra...
## 5 dre00592    pathway alpha-Linolenic acid metabolism - Danio rerio...
## 6 dre01040    pathway Biosynthesis of unsaturated fatty acids - Dan...
## 7 dre01212    pathway Fatty acid metabolism - Danio rerio (zebrafis...
##        p.score
## 1 1.000000e-06
## 2 1.934171e-06
## 3 1.080598e-05
## 4 2.639328e-02
## 5 1.000000e-06
## 6 1.000000e-06
## 7 2.448355e-05
```

And then comparing against the ones in *Figure 3*:

```
path.fig3 <- c(
    "dre00591", # Linoleic acid metabolism
    "dre01040", # Biosynthesis of unsaturated fatty acids
    "dre00592", # alpha-Linolenic acid metabolism
    "dre00564", # Glycerophospholipid metabolism
    "dre00480", # Glutathione metabolism
    "dre00260"  # Glycine, serine and threonine metabolism
)
path.fig3 %in% V(g.plasma)$name
```

```
## [1]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
```

All of them but *Glutathione metabolism* are recovered, showing how FELLA can help gaining perspective on the input metabolites.

Examining the metabolites

As in the analogous section for liver, we will quantify how many input metabolites, drawn within a shaded frame in *Figure 3*, are reported in the sub-network:

```
cpd.plasma %in% V(g.plasma)$name
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

From the 11 highlighted metabolites, only one is not reported by FELLA: *5-Oxo-L-proline*.

Conversely, two out of the three contextual metabolites from the same figure are reported:

```
cpd.fig3 <- c(
    "C01595", # Linoleic acid
    "C00157", # Phosphatidylcholine
    "C00037"  # Glycine
)
cpd.fig3 %in% V(g.plasma)$name
```

```
## [1]  TRUE  TRUE FALSE
```

As *Figure 3* shows, the addition of *linoleic acid* and *phosphatidylcholine*, backed up by FELLA, helps connecting almost all the metabolites found in blood.

FELLA misses *glycine* and, in fact, stays consistent with the pathway (*Glutathione metabolism*) and the input metabolite (*5-Oxo-L-proline*) that it left out from *Figure 3*. The fact that FELLA does not suggest such pathway seems to happen at several molecular levels and therefore none of its metabolites are pinpointed.

Even if the glutathione pathway was not reported, FELLA can greatly ease the creation of elaborated contextual figures, such as *Figure 3*, by suggesting the intermediate metabolites and the metabolic pathways that link the input compounds.

D.3.4 Conclusions

In this vignette, we apply FELLA to an untargeted metabolic study of gilt-head bream exposed to an environmental contaminat (oxybenzome). This study is an example of how FELLA can be useful for (1) organisms not limited to *Homo sapiens*, and (2) conditions not limited to a specific disease.

On one hand, FELLA helps creating complex contextual interpretations of the data, such as the comprehensive *Figure 3* from the original article (Ziarrusta et al., 2018). This material would be challenging to build through regular over-representation analysis of the input metabolites. On the other hand, metabolites and pathways suggested by FELLA were also mentioned in the literature and supported the main findings in the study. In particular, it

helped identify key processes such as *phenylalanine metabolism*, *alpha-linoleic acid metabolism* and *serine metabolism*, which ultimately pointed to alterations in *oxidative stress*.

D.3.5 Reproducibility

This is the result of running `sessionInfo()`

```
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/atlas-base/atlas/libblas.so.3.0
## LAPACK: /usr/lib/atlas-base/atlas/liblapack.so.3.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=es_ES.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=es_ES.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=es_ES.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] magrittr_1.5         igraph_1.2.4.1       KEGGREST_1.24.1
##  [4] org.Mm.eg.db_3.8.2   org.Hs.eg.db_3.8.2   AnnotationDbi_1.46.0
##  [7] IRanges_2.17.5       S4Vectors_0.21.24    Biobase_2.44.0
## [10] BiocGenerics_0.29.2  FELLA_1.5.3          knitr_1.22
##
## loaded via a namespace (and not attached):
##  [1] progress_1.2.2     xfun_0.6           lattice_0.20-38
##  [4] tcltk_3.6.2        vctrs_0.2.0        htmltools_0.3.6
##  [7] yaml_2.2.0         blob_1.2.0         XML_3.98-1.20
## [10] rlang_0.4.0        pillar_1.4.0       DBI_1.0.0
## [13] bit64_0.9-7        plyr_1.8.4         stringr_1.4.0
## [16] zlibbioc_1.29.0    Biostrings_2.51.5  GOSemSim_2.10.0
## [19] evaluate_0.13      memoise_1.1.0      biomaRt_2.40.1
## [22] curl_3.3           highr_0.8          Rcpp_1.0.1
## [25] backports_1.1.4    BiocManager_1.30.4 XVector_0.23.2
## [28] bit_1.1-14         BiocStyle_2.12.0   hms_0.5.0
## [31] png_0.1-7          digest_0.6.18      stringi_1.4.3
## [34] grid_3.6.2         tools_3.6.2        bitops_1.0-6
```

```
## [37] RCurl_1.95-4.12    RSQLite_2.1.1      tibble_2.1.1
## [40] GO.db_3.8.2        crayon_1.3.4       pkgconfig_2.0.2
## [43] zeallot_0.1.0      Matrix_1.2-18      prettyunits_1.0.2
## [46] assertthat_0.2.1   rmarkdown_1.12     httr_1.4.0
## [49] R6_2.4.0           compiler_3.6.2
```

KEGG version:

```
cat(getInfo(fella.data))
```

```
## T01004           Danio rerio (zebrafish) KEGG Genes Database
## dre              Release 93.0+/02-23, Feb 20
##                  Kanehisa Laboratories
##                  26,968 entries
##
## linked db        pathway
##                  brite
##                  module
##                  ko
##                  genome
##                  enzyme
##                  ncbi-geneid
##                  ncbi-proteinid
##                  uniprot
```

Date of generation:

```
date()
```

```
## [1] "Sun Feb 23 11:01:59 2020"
```

Image of the workspace (for submission):

```
tempfile(pattern = "vignette_dre_", fileext = ".RData") %T>%
    message("Saving workspace to ", .) %>%
    save.image(compress = "xz")
```

```
## Saving workspace to /tmp/RtmpAOhDlw/vignette_dre_1493c57be73.RData
```

D.4 ADDITIONAL FILE 4: MOUSE MODEL

D.4.1 Introduction

This vignette shows the utility of the `FELLA` package, which is based in a statistically normalised diffusion process (Picart-Armada et al., 2017), on non-human organisms. In particular, we will work on a multi-omic *Mus musculus* study. The original study (Gogiashvili et al., 2017) presents a mouse model of the non-alcoholic fatty liver disease (NAFLD). Metabolites in liver tissue from leptin-deficient *ob/ob* mice and wild type mice were compared using Nuclear Magnetic Resonance (NMR). Afterwards, quantitative real-time polymerase chain reaction (qRT-PCR) helped identify changes at the gene expression level. Finally, biological mechanisms behind NAFLD were elucidated by leveraging the data from both omics.

Building the database

The first step is to build the `FELLA.DATA` object for the `mmu` organism from the KEGG database (Kanehisa, Furumichi, et al., 2016).

```
library(FELLA)
library(org.Mm.eg.db)
library(KEGGREST)

library(igraph)
library(magrittr)

set.seed(1)
# Filter overview pathways
graph <- buildGraphFromKEGGREST(
    organism = "mmu",
    filter.path = c("01100", "01200", "01210", "01212", "01230"))

tmpdir <- paste0(tempdir(), "/my_database")
# Mke sure the database does not exist from a former vignette build
# Otherwise the vignette will rise an error
# because FELLA will not overwrite an existing database
unlink(tmpdir, recursive = TRUE)
buildDataFromGraph(
    keggdata.graph = graph,
    databaseDir = tmpdir,
    internalDir = FALSE,
    matrices = "none",
    normality = "diffusion",
    niter = 100)
```

We load the `FELLA.DATA` object and two mappings (from gene symbol to entrez identifiers, and from enzyme EC numbers to their annotated entrez genes).

```
alias2entrez <- as.list(org.Mm.eg.db::org.Mm.egSYMBOL2EG)
entrez2ec <- KEGGREST::keggLink("enzyme", "mmu")
entrez2path <- KEGGREST::keggLink("pathway", "mmu")

fella.data <- loadKEGGdata(
    databaseDir = tmpdir,
    internalDir = FALSE,
    loadMatrix = "none"
)
```

Summary of the database:

```
fella.data
```

```
## General data:
## - KEGG graph:
##   * Nodes:  11099
##   * Edges:  34562
##   * Density:  0.0002805888
##   * Categories:
##     + pathway [323]
##     + module [173]
##     + enzyme [1137]
##     + reaction [5467]
##     + compound [3999]
##   * Size:  6.2 Mb
## - KEGG names are ready.
## ----------------------------
## Hypergeometric test:
## - Matrix not loaded.
## ----------------------------
## Heat diffusion:
## - Matrix not loaded.
## - RowSums are ready.
## ----------------------------
## PageRank:
## - Matrix not loaded.
## - RowSums not loaded.
```

In addition, we will store the ids of all the metabolites, reactions and enzymes in the database:

```
id.cpd <- getCom(fella.data, level = 5, format = "id") %>% names
id.rx <- getCom(fella.data, level = 4, format = "id") %>% names
id.ec <- getCom(fella.data, level = 3, format = "id") %>% names
```

Note on reproducibility

We want to emphasise that `FELLA` builds its `FELLA.DATA` object using the most recent version of the KEGG database. KEGG is frequently updated

and therefore small changes can take place in the knowledge graph between different releases. The discussion on our findings was written at the date specified in the vignette header and using the KEGG release in the Reproducibility section.

D.4.2 Enrichment analysis

Defining the input and running the enrichment

Table 2 from the main body in (Gogiashvili et al., 2017) contains six metabolites that show significant changes between the experimental classes by a univariate test followed by multiple test correction. These are the start of our enrichment analysis:

```
cpd.nafld <- c(
    "C00020", # AMP
    "C00719", # Betaine
    "C00114", # Choline
    "C00037", # Glycine
    "C00160", # Glycolate
    "C01104"  # Trimethylamine-N-oxide
)

analysis.nafld <- enrich(
    compounds = cpd.nafld,
    data = fella.data,
    method = "diffusion",
    approx = "normality")
```

```
## No background compounds specified. Default background will be used.

## Running diffusion...

## Computing p-scores through the specified distribution.

## Done.
```

Five compounds are successfully mapped to the graph object:

```
analysis.nafld %>%
    getInput %>%
    getName(data = fella.data)
```

```
## $C00020
## [1] "AMP"                          "Adenosine 5'-monophosphate"
## [3] "Adenylic acid"                "Adenylate"
## [5] "5'-AMP"                       "5'-Adenylic acid"
## [7] "5'-Adenosine monophosphate" "Adenosine 5'-phosphate"
##
## $C00719
```

```
## [1] "Betaine"                 "Trimethylaminoacetate"
## [3] "Glycine betaine"         "N,N,N-Trimethylglycine"
## [5] "Trimethylammonioacetate"
##
## $C00114
## [1] "Choline"     "Bilineurine"
##
## $C00037
## [1] "Glycine"          "Aminoacetic acid" "Gly"
##
## $C00160
## [1] "Glycolate"          "Glycolic acid"      "Hydroxyacetic acid"
##
## $C01104
## [1] "Trimethylamine N-oxide" "(CH3)3NO"
```

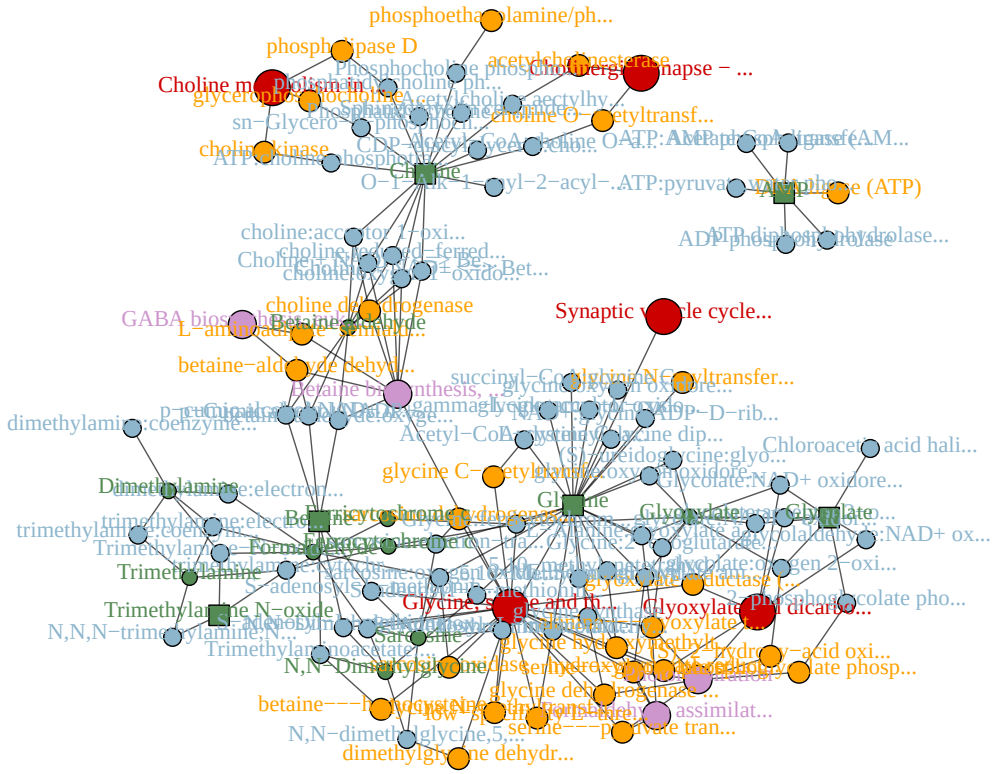
Likewise, one compound does not map:

```
getExcluded(analysis.nafld)
```

```
## character(0)
```

The highlighted subgraph with the default parameters has the following appeareance, with large connected components that involve the metabolites in the input:

```
plot(
    analysis.nafld,
    method = "diffusion",
    data = fella.data,
    nlimit = 250,
    plotLegend = FALSE)
```



We will also extract all the p-scores and the suggested sub-network for further analysis:

```
g.nafld <-  generateResultsGraph(
    object = analysis.nafld,
    data = fella.data,
    method = "diffusion")

pscores.nafld <- getPscores(
    object = analysis.nafld,
    method = "diffusion")
```

Examining the metabolites

FROM TABLE 2 The authors find 5 extra metabolites in *Table 2* that are significant at $p < 0.05$ but do not appear after thresholding the false discovery rate at 5%. Such metabolites, highlighted in italics but without an asterisk, are also relevant and play a role in their discussion. We will examine how FELLA prioritises such metabolites:

```
cpd.nafld.suggestive <- c(
    "C00008", # ADP
    "C00791", # Creatinine
    "C00025", # Glutamate
    "C01026", # N,N-dimethylglycine
    "C00079", # Phenylalanine
    "C00299"  # Uridine
)
getName(cpd.nafld.suggestive, data = fella.data)
```

```
## $C00008
## [1] "ADP"                       "Adenosine 5'-diphosphate"
##
## $C00791
## [1] "Creatinine"              "1-Methylglycocyamidine"
##
## $C00025
## [1] "L-Glutamate"       "L-Glutamic acid"   "L-Glutaminic acid"
## [4] "Glutamate"
##
## $C01026
## [1] "N,N-Dimethylglycine" "Dimethylglycine"
##
## $C00079
## [1] "L-Phenylalanine"
## [2] "(S)-alpha-Amino-beta-phenylpropionic acid"
##
## $C00299
## [1] "Uridine"
```

When checking if any of these metabolites are found in the reported subnetwork, we find that `C01026` is already reported:

```
V(g.nafld)$name %>%
    intersect(cpd.nafld.suggestive) %>%
    getName(data = fella.data)
```

```
## $C01026
## [1] "N,N-Dimethylglycine" "Dimethylglycine"
```

Abbreviated as **DMG** in their study, N,N-Dimethylglycine is a cornerstone of their findings. It is reported in Figure 6a as part of the folate-independent remethylation to explain the metabolic changes observed in the *ob/ob* mice. **DMG** is also mentioned in the conclusions as part of one of the most prominent alterations found in the study: a reduced conversion of betaine to **DMG**.

FROM FIGURE 6A *Figure 6a* contains the metabolic context of the observed alterations, with processes such as transsulfuration and folate-dependent remethylation. These were identified with the help of gene expression analysis. We will now check for coincidences between the metabolites in *Figure 6a*, excluding choline and betaine for being in the input and DMG since it was already discussed.

```
cpd.new.fig6 <- c(
    "C00101", # THF
    "C00440", # 5-CH3-THF
    "C00143", # 5,10-CH3-THF
    "C00073", # Methionine
```

```
    "C00019", # SAM
    "C00021", # SAH
    "C00155", # Homocysteine
    "C02291", # Cystathione
    "C00097"  # Cysteine
)
getName(cpd.new.fig6, data = fella.data)
```

```
## $C00101
## [1] "Tetrahydrofolate"          "5,6,7,8-Tetrahydrofolate"
## [3] "Tetrahydrofolic acid"      "THF"
## [5] "(6S)-Tetrahydrofolate"     "(6S)-Tetrahydrofolic acid"
## [7] "(6S)-THFA"
##
## $C00440
## [1] "5-Methyltetrahydrofolate"
##
## $C00143
## [1] "5,10-Methylenetetrahydrofolate"
## [2] "(6R)-5,10-Methylenetetrahydrofolate"
## [3] "5,10-Methylene-THF"
##
## $C00073
## [1] "L-Methionine"                      "Methionine"
## [3] "L-2-Amino-4methylthiobutyric acid"
##
## $C00019
## [1] "S-Adenosyl-L-methionine" "S-Adenosylmethionine"
## [3] "AdoMet"                  "SAM"
##
## $C00021
## [1] "S-Adenosyl-L-homocysteine" "S-Adenosylhomocysteine"
##
## $C00155
## [1] "L-Homocysteine"                   "L-2-Amino-4-mercaptobutyric acid"
## [3] "Homocysteine"
##
## $C02291
## [1] "L-Cystathionine"
##
## $C00097
## [1] "L-Cysteine"
## [2] "L-2-Amino-3-mercaptopropionic acid"
```

This time, there are no coincidences with the reported sub-network:

```
cpd.new.fig6 %in% V(g.nafld)$name
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

However, we can further inquire whether the p-scores of such metabolites tend to be low among all the metabolites in the whole network from the `fella.data` object.

```
wilcox.test(
    x = pscores.nafld[cpd.new.fig6], # metabolites from fig6
    y = pscores.nafld[setdiff(id.cpd, cpd.new.fig6)], # rest of metabolites
    alternative = "less")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pscores.nafld[cpd.new.fig6] and pscores.nafld[setdiff(id.cpd, cpd.new.fig6)]
## W = 1292, p-value = 7.321e-07
## alternative hypothesis: true location shift is less than 0
```

The test is indeed significant – despite `FELLA` does not directly report such metabolites, its metabolite ranking supports the claims by the authors.

Examining the genes

CBS The authors complement the metabolomic profilings with a differential gene expression study. One of the main findings is a change of *Cbs* expression levels. To link *Cbs* to the enrichment from `FELLA`, we will first map it to its EC number, *4.2.1.22* at the time of writing:

```
ec.cbs <- entrez2ec[[paste0("mmu:", alias2entrez[["Cbs"]])]] %>%
    gsub(pattern = "ec:", replacement = "")

getName(fella.data, ec.cbs)
```

```
## $`4.2.1.22`
## [1] "cystathionine beta-synthase"
## [2] "serine sulfhydrase"
## [3] "beta-thionase"
## [4] "methylcysteine synthase"
## [5] "cysteine synthase (incorrect)"
## [6] "serine sulfhydrylase"
## [7] "L-serine hydro-lyase (adding homocysteine)"
```

In *Figure 6a*, the reaction linked to *Cbs* and catalysed by the enzyme *4.2.1.22* has the KEGG identifier *R01290*.

```
rx.cbs <- "R01290"

getName(fella.data, rx.cbs)
```

```
## $R01290
## [1] "L-serine hydro-lyase (adding homocysteine"
## [2] "L-cystathionine-forming)"
## [3] "L-Serine + L-Homocysteine <=> L-Cystathionine + H2O"
```

As shown in *Figure 6a*, *Cbs* is not directly linked to the metabolites found through NMR, and nor the reaction neither the enzyme are suggested by FELLA:

```
c(rx.cbs, ec.cbs) %in% V(g.nafld)$name
```

```
## [1] FALSE FALSE
```

However, both of them have a relatively low p-score in their respective categories. This can be seen through the proportion of enzymes (resp. reactions) that show a p-score as low or lower than *4.2.1.22* (resp. *R01290*)

```
# enzyme
pscores.nafld[ec.cbs]
```

```
##  4.2.1.22
## 0.4299332
```

```
mean(pscores.nafld[id.ec] <= pscores.nafld[ec.cbs])
```

```
## [1] 0.2040457
```

```
# reaction
pscores.nafld[rx.cbs]
```

```
##    R01290
## 0.2774493
```

```
mean(pscores.nafld[id.rx] <= pscores.nafld[rx.cbs])
```

```
## [1] 0.03347357
```

It's not surprising that none of them is directly reported, because none of the metabolites participating in the reaction is found in the input. The main evidence for finding *Cbs* is gene expression, and our approach gives indirect hints of this connection.

BHMT The alteration of *Bhmt* activity is related to the downregulation of *Cbs*. Despite not finding evidence of change in *Bhmt* expression, the authors argue that its inhibition would explain the increased betaine-to-DMG ratio in *ob/ob* mice. Such claim is also backed up by prior studies. To find out the role of *Cbs* in our analysis, we will again map it to its EC number, *2.1.1.5*:

```
ec.bhmt <- entrez2ec[[paste0("mmu:", alias2entrez[["Bhmt"]])]] %>%
    gsub(pattern = "ec:", replacement = "")

getName(fella.data, ec.bhmt)
```

```
## $`2.1.1.5`
## [1] "betaine---homocysteine S-methyltransferase"
## [2] "betaine-homocysteine methyltransferase"
## [3] "betaine-homocysteine transmethylase"
```

This time, FELLA not only reports it, but also its associated reaction *R02821* (represented by an arrow in *Figure 6a*) and both of its metabolites. While **betaine** was already an input metabolite, **DMG** was a novel finding as discussed earlier

```
ec.bhmt %in% V(g.nafld)$name
```

```
## [1] TRUE
```

```
"R02821" %in% V(g.nafld)$name
```

```
## [1] TRUE
```

This illustrates how FELLA can translate knowledge from dysregulated metabolites to other molecular levels, such as reactions and enzymes.

SLC22A5 The decrease of *Bhmt* activity is later connected to the upregulation of *Slc22a5*, also proved within the original study. However, *Slc22a5* does not map to any EC number and therefore it cannot be found through FELLA:

```
entrez.slc22a5 <- alias2entrez[["Slc22a5"]]
entrez.slc22a5 %in% names(entrez2ec)
```

```
## [1] FALSE
```

As a matter of fact, the only connection that can be found from KEGG is the role of *Slc22a5* in the *Choline metabolism in cancer* pathway.

```
path.slc22a5 <- entrez2path[paste0("mmu:", entrez.slc22a5)] %>%
    gsub(pattern = "path:", replacement = "")

getName(fella.data, path.slc22a5)
```

```
## $mmu05231
## [1] "Choline metabolism in cancer - Mus musculus (mouse)"
```

Coincidentally, this pathway is reported in the sub-graph:

```
path.slc22a5 %in% V(g.nafld)$name
```

```
## [1] TRUE
```

GENES FROM FIGURE 3 We also examined if genes from *Table 3* were reachable in our analysis. These five literature-derived genes were experimentally confirmed to show gene expression changes, in order to prove that RNA extracted after the metabolomic profiling was still reliable for further transcriptomic analyses. However, only *Scd2* maps to an enzymatic family:

```
symbol.fig3 <- c(
    "Cd36",
    "Scd2",
    "Apoa4",
    "Lcn2",
    "Apom")

entrez.fig3 <- alias2entrez[symbol.fig3] %>% unlist %>% unique
ec.fig3 <- entrez2ec[paste0("mmu:", entrez.fig3)] %T>%
    print %>%
    unlist %>%
    unique %>%
    na.omit %>%
    gsub(pattern = "ec:", replacement = "")
```

```
##           <NA>      mmu:20250           <NA>           <NA>           <NA>
##             NA "ec:1.14.19.1"             NA             NA             NA
```

```
getName(fella.data, ec.fig3)
```

```
## $`1.14.19.1`
## [1] "stearoyl-CoA 9-desaturase"
## [2] "Delta9-desaturase"
## [3] "acyl-CoA desaturase"
## [4] "fatty acid desaturase"
## [5] "stearoyl-CoA, hydrogen-donor:oxygen oxidoreductase"
```

Such family is not reported in our sub-graph

```
ec.fig3 %in% V(g.nafld)$name
```

```
## [1] FALSE
```

In addition, its p-score is high compared to other enzymes

```
pscores.nafld[ec.fig3]
```

```
## 1.14.19.1
## 0.5816303
```

```
mean(pscores.nafld[id.ec] <= pscores.nafld[ec.fig3])
```

```
## [1] 0.7985928
```

The fact that only one gene mapped to an EC number hinders the potential findings using FELLA, and is probably the main reason why FELLA missed *Scd2*. In addition, FELLA defines a knowledge model that offers simplicity and interpretability, at the cost of introducing limitations on how sophisticated its findings can be.

GENES FROM TABLE S2 In parallel with the original study, and cited within its main body, gene array expression data was collected (Godoy et al., 2016) and its hits are included in the supplementary *Table S2* from (Gogiashvili et al., 2017). These genes include the already discussed *Cbs*. We will attempt to link the genes marked as significantly changed to our reported sub-network. In contrast with *Figure 3*, all the genes map to an EC number:

```
symbol.tableS2 <- c(
    "Mat1a",
    "Ahcyl2",
    "Cbs",
    "Mat2b",
    "Mtr")
entrez.tableS2 <- alias2entrez[symbol.tableS2] %>% unlist %>% unique
ec.tableS2 <- entrez2ec[paste0("mmu:", entrez.tableS2)] %T>%
    print %>%
    unlist %>%
    unique %>%
    na.omit %>%
    gsub(pattern = "ec:", replacement = "")
```

```
##     mmu:11720     mmu:74340     mmu:12411    mmu:108645    mmu:238505
##  "ec:2.5.1.6"  "ec:3.3.1.1" "ec:4.2.1.22"  "ec:2.5.1.6" "ec:2.1.1.13"
```

None of these EC families are reported in the sub-graph:

```
ec.tableS2 %in% V(g.nafld)$name
```

```
## [1] FALSE FALSE FALSE FALSE
```

But in this case, their scores tend to be lower than the rest of enzymes:

```
wilcox.test(
    x = pscores.nafld[ec.tableS2], # enzymes from table S2
    y = pscores.nafld[setdiff(id.ec, ec.tableS2)], # rest of enzymes
    alternative = "less")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pscores.nafld[ec.tableS2] and pscores.nafld[setdiff(id.ec, ec.tableS2)]
## W = 1203, p-value = 0.05254
## alternative hypothesis: true location shift is less than 0
```

These findings suggest that if the annotation database is complete enough, FELLA can provide a meaningful priorisation of the enzymes surrounding the affected metabolites.

D.4.3 Conclusions

FELLA has been used to give a biological meaning to a list of 6 metabolites extracted from a multi-omic study of a mouse model of NAFLD. It has been able to reproduce some findings at the metabolite and gene expression levels, whereas most of the times missed entities would still present a low ranking compared to their background in the database.
The bottom line from our analysis in the present vignette is that FELLA not only works on human studies, but also generalises to animal models.

D.4.4 Reproducibility

This is the result of running `sessionInfo()`

```
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/atlas-base/atlas/libblas.so.3.0
## LAPACK: /usr/lib/atlas-base/atlas/liblapack.so.3.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=es_ES.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=es_ES.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=es_ES.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] magrittr_1.5         igraph_1.2.4.1       KEGGREST_1.24.1
##  [4] org.Mm.eg.db_3.8.2   org.Hs.eg.db_3.8.2   AnnotationDbi_1.46.0
##  [7] IRanges_2.17.5       S4Vectors_0.21.24    Biobase_2.44.0
## [10] BiocGenerics_0.29.2  FELLA_1.5.3          knitr_1.22
##
## loaded via a namespace (and not attached):
##  [1] progress_1.2.2     xfun_0.6           lattice_0.20-38
##  [4] tcltk_3.6.2        vctrs_0.2.0        htmltools_0.3.6
```

```
##  [7] yaml_2.2.0          blob_1.2.0          XML_3.98-1.20
## [10] rlang_0.4.0         pillar_1.4.0        DBI_1.0.0
## [13] bit64_0.9-7         plyr_1.8.4          stringr_1.4.0
## [16] zlibbioc_1.29.0     Biostrings_2.51.5   GOSemSim_2.10.0
## [19] evaluate_0.13       memoise_1.1.0       biomaRt_2.40.1
## [22] curl_3.3            highr_0.8           Rcpp_1.0.1
## [25] backports_1.1.4     BiocManager_1.30.4  XVector_0.23.2
## [28] bit_1.1-14          BiocStyle_2.12.0    hms_0.5.0
## [31] png_0.1-7           digest_0.6.18       stringi_1.4.3
## [34] grid_3.6.2          tools_3.6.2         bitops_1.0-6
## [37] RCurl_1.95-4.12     RSQLite_2.1.1       tibble_2.1.1
## [40] GO.db_3.8.2         crayon_1.3.4        pkgconfig_2.0.2
## [43] zeallot_0.1.0       Matrix_1.2-18       prettyunits_1.0.2
## [46] assertthat_0.2.1    rmarkdown_1.12      httr_1.4.0
## [49] R6_2.4.0            compiler_3.6.2
```

KEGG version:

```
cat(getInfo(fella.data))
```

```
## T01002           Mus musculus (mouse) KEGG Genes Database
## mmu              Release 93.0+/02-23, Feb 20
##                  Kanehisa Laboratories
##                  25,849 entries
##
## linked db        pathway
##                  brite
##                  module
##                  ko
##                  genome
##                  mgi
##                  enzyme
##                  ncbi-geneid
##                  ncbi-proteinid
##                  uniprot
```

Date of generation:

```
date()
```

```
## [1] "Sun Feb 23 10:40:29 2020"
```

Image of the workspace (for submission):

```
tempfile(pattern = "vignette_mmu_", fileext = ".RData") %T>%
    message("Saving workspace to ", .) %>%
    save.image(compress = "xz")
```

```
## Saving workspace to /tmp/RtmpAOhDlw/vignette_mmu_149468fbae3.RData
```

REFERENCES

Aggio, Raphael BM, Katya Ruggiero, and Silas Granato Villas-Bôas

2010 "Pathway Activity Profiling (PAPi): from the metabolite profile to the metabolic pathway activity", *Bioinformatics*, 26, 23, pp. 2969-2976.

Alhamdoosh, Monther, Milica Ng, Nicholas J Wilson, Julie M Sheridan, Huy Huynh, Michael J Wilson, and Matthew E Ritchie

2017 "Combining multiple tools outperforms individual methods in gene set enrichment analyses", *Bioinformatics*, 33, 3, pp. 414-424.

Beisel, William R

1975 "Metabolic response to infection", *Annual review of medicine*, 26, 1, pp. 9-20.

Chagoyen, Monica and Florencio Pazos

2012 "Tools for the functional interpretation of metabolomic experiments", *Briefings in bioinformatics*, 14, 6, pp. 737-744.

Chang, Winston, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson

2017 *shiny: Web Application Framework for R*, R package version 1.0.5, https://CRAN.R-project.org/package=shiny.

Chen, Liyan, Jing Li, Tiannan Guo, Sujoy Ghosh, Siew Kwan Koh, Dechao Tian, Liang Zhang, Deyong Jia, Roger W Beuerman, Ruedi Aebersold, et al.

2015 "Global metabonomic and proteomic analysis of human conjunctival epithelial cells (IOBA-NHC) in response to hyperosmotic stress", *Journal of proteome research*, 14, 9, pp. 3982-3995.

Consortium, Gene Ontology et al.

2015 "Gene ontology consortium: going forward", *Nucleic acids research*, 43, D1, pp. D1049-D1056.

Csardi, Gabor and Tamas Nepusz

2006 "The igraph software package for complex network research", *InterJournal*, Complex Systems, p. 1695.

Decuypere, Saskia, Jessica Maltha, Stijn Deborggraeve, Nicholas JW Rattray, Guiraud Issa, Kaboré Bérenger, Palpouguini Lompo, Marc C Tahita, Thusitha Ruspasinghe, Malcolm McConville, et al.

2016 "Towards Improving Point-of-Care Diagnosis of Non-malaria Febrile Illness: A Metabolomics Approach", *PLoS neglected tropical diseases*, 10, 3, e0004480.

Fabregat, Antonio, Konstantinos Sidiropoulos, Phani Garapati, Marc Gillespie, Kerstin Hausmann, Robin Haw, Bijay Jassal, Steven Jupe, Florian Korninger, Sheldon McKay, et al.

2015 "The reactome pathway knowledgebase", *Nucleic acids research*, 44, D1, pp. D481-D487.

Fernández-Albert, Francesc, Rafael Llorach, Cristina Andrés-Lacueva, and Alexandre Perera

2014 "An R package to analyse LC/MS metabolomic data: MAIT (Metabolite Automatic Identification Toolkit)", *Bioinformatics*, 30, 13, pp. 1937-1939.

Fitch, Coy D, Guang-zuan Cai, and James D Shoemaker

2000 "A role for linoleic acid in erythrocytes infected with Plasmodium berghei", *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1535, 1, pp. 45-49.

Godoy, Patricio, Agata Widera, Wolfgang Schmidt-Heck, Gisela Campos, Christoph Meyer, Cristina Cadenas, Raymond Reif, Regina Stöber, Seddik Hammad, Larissa Pütter, et al.

2016 "Gene network activity in cultivated primary hepatocytes is highly similar to diseased mammalian liver tissue", *Archives of toxicology*, 90, 10, pp. 2513-2529.

Gogiashvili, Mikheil, Karolina Edlund, Kathrin Gianmoena, Rosemarie Marchan, Alexander Brik, Jan T Andersson, Jörg Lambert, Katrin Madjar, Birte Hellwig, Jörg Rahnenführer, et al.

2017 "Metabolic profiling of ob/ob mouse fatty liver using HR-MAS 1 H-NMR combined with gene expression analysis reveals alterations in betaine metabolism and the transsulfuration pathway", *Analytical and bioanalytical chemistry*, 409, 6, pp. 1591-1606.

Grapov, Dmitry, Kwanjeera Wanichthanarak, and Oliver Fiehn

2015 "MetaMapR: pathway independent metabolomic network analysis incorporating unknowns", *Bioinformatics*, 31, 16, pp. 2757-2760.

Haug, Kenneth, Reza M Salek, Pablo Conesa, Janna Hastings, Paula de Matos, Mark Rijnbeek, Tejasvi Mahendraker, Mark Williams, Steffen Neumann, Philippe Rocca-Serra, et al.

2012 "MetaboLights – an open-access general-purpose repository for metabolomics studies and associated meta-data", *Nucleic Acids Res.*, 41, D1, pp. D781-D786.

Kaelin, William G and Craig B Thompson

2010 "Q&A: Cancer: clues from cell metabolism." *Nature*, 465, 7298, pp. 562-564, ISSN: 0028-0836, DOI: 10.1038/465562a.

Kamburov, Atanas, Rachel Cavill, Timothy MD Ebbels, Ralf Herwig, and Hector C Keun

2011 "Integrated pathway-level analysis of transcriptomics and metabolomics data with IMPaLA", *Bioinformatics*, 27, 20, pp. 2917-2918.

Kanehisa, Minoru, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima

2016 "KEGG: new perspectives on genomes, pathways, diseases and drugs", *Nucleic acids research*, 45, D1, pp. D353-D361.

Kanehisa, Minoru, Susumu Goto, Yoko Sato, Miho Furumichi, and Mao Tanabe

2011 "KEGG for integration and interpretation of large-scale molecular data sets", *Nucleic acids research*, 40, D1, pp. D109-D114.

Karnovsky, Alla, Terry Weymouth, Tim Hull, V Glenn Tarcea, Giovanni Scardoni, Carlo Laudanna, Maureen A Sartor, Kathleen A Stringer, HV Jagadish, Charles Burant, et al.

2011 "Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data", *Bioinformatics*, 28, 3, pp. 373-380.

Kessler, Nikolas, Heiko Neuweger, Anja Bonte, Georg Langenkämper, Karsten Niehaus, Tim W Nattkemper, and Alexander Goesmann

2013 "MeltDB 2.0–advances of the metabolomics software system", *Bioinformatics*, 29, 19, pp. 2452-2459.

Khatri, Purvesh, Marina Sirota, and Atul J. Butte

2012 "Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges", *PLoS Computational Biology*, 8, 2.

Kirkegaard, Thomas and Marja Jäättelä

2009 "Lysosomal involvement in cell death and cancer", *Biochimica et Biophysica Acta - Molecular Cell Research*, 1793, 4, pp. 746-754, ISSN: 01674889, DOI: 10.1016/j.bbamcr.2008.09.008.

Kuhl, Carsten, Ralf Tautenhahn, Christoph Bottcher, Tony R Larson, and Steffen Neumann

2011 "CAMERA: an integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets", *Analytical chemistry*, 84, 1, pp. 283-289.

Kutmon, Martina, Anders Riutta, Nuno Nunes, Kristina Hanspers, Egon L Willighagen, Anwesha Bohler, Jonathan Mélius, Andra Waagmeester, Sravanthi R Sinha, Ryan Miller, et al.

2015 "WikiPathways: capturing the full diversity of pathway knowledge", *Nucleic acids research*, 44, D1, pp. D488-D494.

Lee, Insuk, U Martin Blom, Peggy I Wang, Jung Eun Shim, and Edward M Marcotte

2011 "Prioritizing candidate disease genes by network-based boosting of genome-wide association data", *Genome research*, 21, 7, pp. 1109-1121.

Lehtonen, Heli J., Ignacio Blanco, Jose M. Piulats, Riitta Herva, Virpi Launonen, and Lauri A. Aaltonen

2007 "Conventional renal cancer in a patient with fumarate hydratase mutation", *Human Pathology*, 38, 5, pp. 793-796, ISSN: 00468177, DOI: 10.1016/j.humpath.2006.10.011.

Li, Chunquan, Xia Li, Yingbo Miao, Qianghu Wang, Wei Jiang, Chun Xu, Jing Li, Junwei Han, Fan Zhang, Binsheng Gong, et al.

2009 "SubpathwayMiner: a software package for flexible identification of pathways", *Nucleic acids research*, 37, 19, e131-e131.

Li, Feng, Yanjun Xu, Desi Shang, Haixiu Yang, Wei Liu, Junwei Han, Zeguo Sun, Qianlan Yao, Chunlong Zhang, Jiquan Ma, et al.

2014 "MPINet: Metabolite pathway identification via coupling of global metabolite network structure and metabolomic profile", *BioMed research international*, 2014.

Maceyka, Michael and Sarah Spiegel

2014 "Sphingolipid metabolites in inflammatory disease", *Nature*, 510, 7503, p. 58.

Madsen, Rasmus, Torbjörn Lundstedt, and Johan Trygg

2010 "Chemometrics in metabolomics – a review in human disease diagnosis", *Analytica chimica acta*, 659, 1, pp. 23-33.

Mostafavi, Sara, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris

2008 "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function", *Genome biology*, 9, 1, S4.

Ni, Ying, Kevin M. Zbuk, Tammy Sadler, Attila Patocs, Glenn Lobo, Emily Edelman, Petra Platzer, Mohammed S. Orloff, Kristin A. Waite, and Charis Eng

2008 "Germline Mutations and Variants in the Succinate Dehydrogenase Genes in Cowden and Cowden-like Syndromes", *American Journal of Human Genetics*, 83, 2, pp. 261-268, ISSN: 00029297, DOI: 10.1016/j.ajhg.2008.07.011.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd

1999 *The PageRank citation ranking: Bringing order to the web.* Tech. rep., Stanford InfoLab.

Paull, Evan O, Daniel E Carlin, Mario Niepel, Peter K Sorger, David Haussler, and Joshua M Stuart

2013 "Discovering causal pathways linking genomic events to transcriptional states using Tied Diffusion Through Interacting Events (TieDIE)", *Bioinformatics*, 29, 21, pp. 2757-2764.

Picart-Armada, Sergio, Francesc Fernández-Albert, Maria Vinaixa, Miguel A Rodríguez, Suvi Aivio, Travis H Stracker, Oscar Yanes, and Alexandre Perera-Lluna

2017 "Null diffusion-based enrichment for metabolomics data", *PloS one*, 12, 12, e0189012.

Pithukpakorn, M, M-H Wei, O Toure, P J Steinbach, G M Glenn, B Zbar, W M Linehan, and J R Toro

2006 "Fumarate hydratase enzyme activity in lymphoblastoid cells and fibroblasts of individuals in families with hereditary leiomyomatosis and renal cell cancer." *Journal of medical genetics*, 43, 9, pp. 755-62, ISSN: 1468-6244, DOI: 10.1136/jmg.2006.041087.

Pollard, Patrick, Noel Wortham, and Ian Tomlinson

2003 "The TCA cycle and tumorigenesis: the examples of fumarate hydratase and succinate dehydrogenase", *Annals of Medicine*, 35, 8, pp. 634-639, ISSN: 0785-3890, DOI: 10.1080/07853890310018458.

R Core Team

2017 *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Rodriguez-Martinez, Andrea, Rafael Ayala, Joram M Posma, Ana L Neves, Dominique Gauguier, Jeremy K Nicholson, and Marc-Emmanuel Dumas

2017 "MetaboSignal: a network-based approach for topological analysis of metabotype regulation via metabolic and signaling pathways", *Bioinformatics*, 33, 5, pp. 773-775.

Seo, Young-Jin, Stephen Alexander, and Bumsuk Hahm

2011 "Does cytokine signaling link sphingolipid metabolism to host defense and immunity against virus infections?", *Cytokine & growth factor reviews*, 22, 1, pp. 55-61.

Singh, Keshav K, Mohamed M Desouki, Renty B Franklin, and Leslie C Costello

2006 "Mitochondrial aconitase and citrate metabolism in malignant and nonmalignant human prostate tissues." *Molecular cancer*, 5, p. 14, ISSN: 1476-4598, DOI: 10.1186/1476-4598-5-14.

Smith, Colin A, Elizabeth J Want, Grace O'Maille, Ruben Abagyan, and Gary Siuzdak

2006 "XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification", *Analytical chemistry*, 78, 3, pp. 779-787.

Smoot, Michael E, Keiichiro Ono, Johannes Ruscheinski, Peng-Liang Wang, and Trey Ideker

2010 "Cytoscape 2.8: new features for data integration and network visualization", *Bioinformatics*, 27, 3, pp. 431-432.

Tenenbaum, Dan

2017 *KEGGREST: Client-side REST access to KEGG*, R package version 1.16.1.

Vandin, Fabio, Eli Upfal, and Benjamin J Raphael

2011 "Algorithms for detecting significantly mutated pathways in cancer", *J. Comput. Biol.*, 18, 3, pp. 507-522.

Vermeersch, Kathleen A, Lijuan Wang, John F McDonald, and Mark P Styczynski

2014 "Distinct metabolic responses of an ovarian cancer stem cell line", *BMC systems biology*, 8, 1, p. 134.

Weckwerth, Wolfram

2003 "Annual Review of Plant Biology", 54, 1, pp. 669-689.

Wishart, David S, Timothy Jewison, An Chi Guo, Michael Wilson, Craig Knox, Yifeng Liu, Yannick Djoumbou, Rupasri Mandal, Farid Aziat, Edison Dong, et al.

2012 "HMDB 3.0 – the human metabolome database in 2013", *Nucleic acids research*, 41, D1, pp. D801-D807.

Xia, Jianguo, Igor V Sinelnikov, Beomsoo Han, and David S Wishart

2015 "MetaboAnalyst 3.0 – making metabolomics more meaningful", *Nucleic Acids Research*, 43, Web Server issue, W251-W257.

Xia, Jianguo and David S Wishart

2010 "MSEA: a web-based tool to identify biologically meaningful patterns in quantitative metabolomic data", *Nucleic acids research*, 38, suppl_2, W71-W77.

Yu, Guangchuang, Fei Li, Yide Qin, Xiaochen Bo, Yibo Wu, and Shengqi Wang

2010 "GOSemSim: an R package for measuring semantic similarity among GO terms and gene products", *Bioinformatics*, 26, 7, pp. 976-978.

Yu, Guangchuang, Li-Gen Wang, Yanyan Han, and Qing-Yu He

2012 "clusterProfiler: an R package for comparing biological themes among gene clusters", *Omics: a journal of integrative biology*, 16, 5, pp. 284-287.

Zhang, Jitao David and Stefan Wiemann

2009 "KEGGgraph: a graph approach to KEGG PATHWAY in R and bioconductor", *Bioinformatics*, 25, 11, pp. 1470-1471.

Ziarrusta, Haizea, Leire Mijangos, Sergio Picart-Armada, Mireia Irazola, Alexandre Perera-Lluna, Aresatz Usobiaga, Ailette Prieto, Nestor Etxebarria, Maitane Olivares, and Olatz Zuloaga

2018 "Non-targeted metabolomics reveals alterations in liver and plasma of gilt-head bream exposed to oxybenzone", *Chemosphere*, 211, pp. 624-631.

E | DISEASE GENE IDENTIFICATION

APPENDIX: BENCHMARKING NETWORK PROPAGATION METHODS FOR DISEASE GENE IDENTIFICATION

E.1 DESCRIPTIVE STATISTICS

E.1.1 OpenTargets data streams

File `17.06_association_data.json` with gene-disease associations from June 2017 was downloaded from the Open Targets data download page. The original table consists of 187,246 rows and 7 data streams, encompassing associations between 90 diseases and genes with evidence on one or more streams. We selected those diseases that had at least 50 drug-associated and genetically-associated genes, resulting in a final list of 22 common diseases as shown in the main body.

Table 28: Descriptive statistics on the seven OpenTargets data streams and the overall score. Included are the binarised scores (genetic and drugs) used for the benchmark.

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
association_score.overall	187,246	0.079	0.160	0.0004	0.016	0.032	0.064	1.000
association_score.datatypes.genetic_association	187,246	0.015	0.082	0.000	0.000	0.000	0.000	1.000
association_score.datatypes.somatic_mutation	187,246	0.0004	0.015	0.000	0.000	0.000	0.000	1.000
association_score.datatypes.known_drug	187,246	0.023	0.140	0.000	0.000	0.000	0.000	1.000
association_score.datatypes.affected_pathway	187,246	0.0003	0.017	0	0	0	0	1
association_score.datatypes.rna_expression	187,246	0.015	0.037	0.000	0.000	0.000	0.015	0.651
association_score.datatypes.literature	187,246	0.021	0.030	0.000	0.000	0.014	0.030	0.321
association_score.datatypes.animal_model	187,246	0.008	0.036	0.000	0.000	0.000	0.000	0.313
known_drug_binary	187,246	0.035	0.184	0	0	0	0	1
known_gene_binary	187,246	0.032	0.176	0	0	0	0	1

E.1.2 Networks from the STRING database

STRING data: version 10, species 9606, score threshold 400. STRING uses the ENSEMBL protein identifiers (Zerbino et al., 2018), so the OpenTargets associations were mapped from ENSEMBL gene to ENSEMBL protein through the *map()* function from the STRINGdb package (Szklarczyk et al., 2014). No collisions (i.e. two genes mapping to the same protein) were encountered.

This appendix reproduces the supplementary data (S1 Appendix) of: Picart-Armada, Sergio, Steven J. Barrett, David R. Willé, Alexandre Perera-Lluna, Alex Gutteridge, and Benoit H. Dessailly. "Benchmarking network propagation methods for disease gene identification". *PLoS computational biology 15, no. 9 (2019): e1007276.*

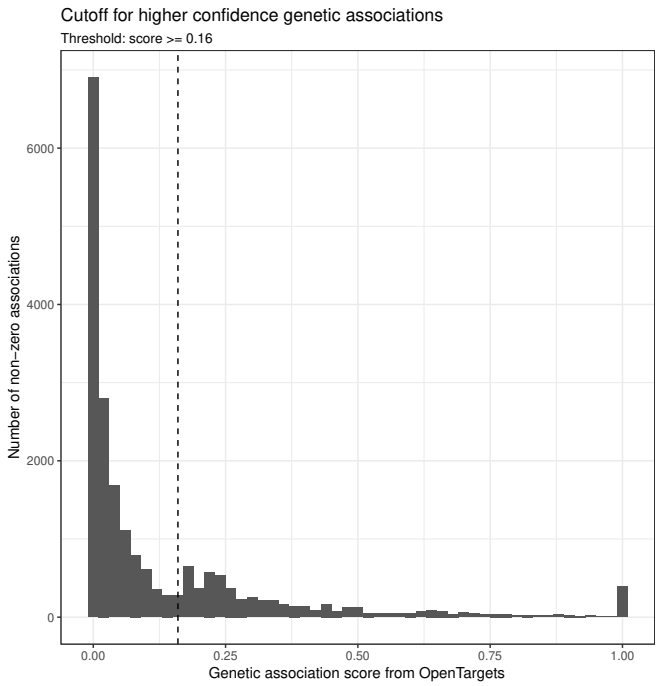


Figure 94: Histogram of genetic associations. The proposed threshold of 0.16 separates lower and higher quality genetic associations and is therefore used to binarise this data stream.

We choose Net4 as our main network, as it provides a good balance between mapping and coverage. The edge weights were obtained by rescaling the STRING combined score to lie in $[0, 1]$. For the MashUp network-based feature generation, the experimental and the database STRING-based networks before combined through their algorithm, instead of using the combined weight provided by STRING.

The (unweighted) shortest path distribution of the final STRING network is depicted in figure 95:

E.1.3 The OmniPath network

The original OmniPath file contained a total of 8,951 nodes and 50,247 edges. Removing duplicated edges and keeping the largest component left a network with 8,580 nodes and 42,145 edges. The proteins are represented by their UniProt identifier (TheUniProtConsortium, 2017) and later mapped to ENSEMBL protein (Zerbino et al., 2018). After mapping, 62 genes mapped to a non-unique protein. For these proteins with multiple gene annotations, we chose:

1. The gene with a known drug target
2. In case of tie(s), the gene with a known genetic association
3. In case of tie(s), the gene with the highest overall association score
4. In case of tie(s), pick a random gene

Table 29: Summary of the STRING networks with several filtering options; edges that meet the filtering condition are dropped. Described are the number of nodes, edges, rows from the disease table whose protein maps to the network (originally, 187,246 rows) and coverage of the binarised drugs and genetic scores. Numbers referring to the largest connected component are outside the parentheses, while the original amount is detailed inside them. The filters apply only to the edges, therefore all the networks have the same order (18884), mapped rows and mapped genes before taking the largest connected component.

network	filter	nodes	edges	coverage_allrows	coverage_drug	coverage_genetic
Net1	combined_score < 400 | experiments < 1	13307(18884)	103607(103648)	153747(178622)	5687(6395)	4593(5751)
Net2	experiments < 600	8854(18884)	37084(37288)	109535(178622)	3791(6395)	3115(5751)
Net3	experiments < 400 & database < 400	14149(18884)	284759(284786)	159554(178622)	6190(6395)	4750(5751)
Net4	combined_score < 700 | (experiments < 1 & database < 1)	11748(18884)	236963(237049)	144920(178622)	6121(6395)	4170(5751)
Net5	combined_score < 700 | (experiments < 1 & database < 1 & textmining < 900)	12022(18884)	240082(240193)	147635(178622)	6160(6395)	4266(5751)
Net6	database < 400	7564(18884)	205866(206177)	110293(178622)	5711(6395)	2881(5751)
All	combined_score < 0	18884(18884)	740950(740950)	178622(178622)	6395(6395)	5751(5751)

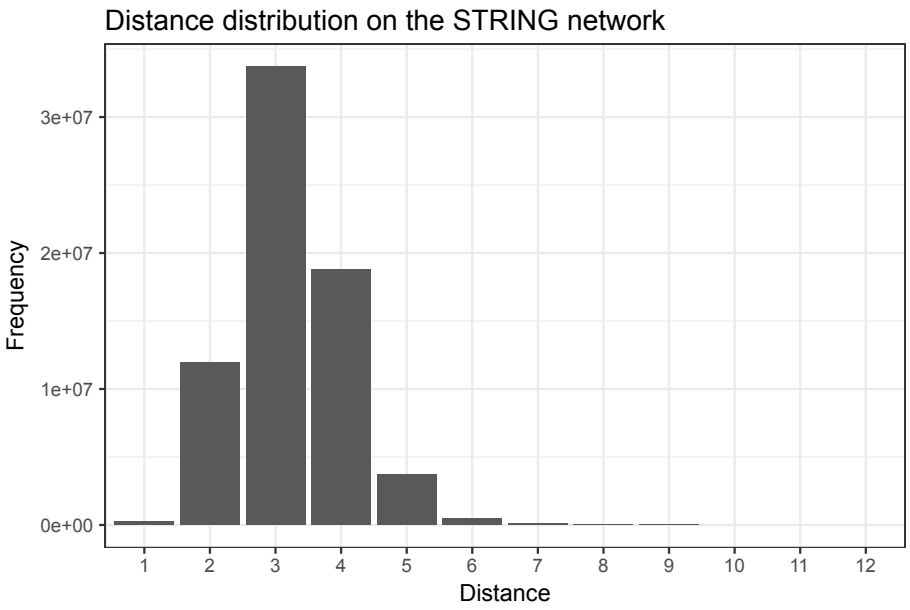


Figure 95: Distance distribution in the STRING network, computed for every pair of nodes. Most of the nodes lie within a distance of 5 or less, suggesting the presence of biological hubs.

From the original 187,246 rows in the disease table, 125,007 mapped to the OmniPath network, encompassing a total of 5,084 drugs-related genes and 3,442 genetics-related genes.

E.1.4 Descriptive disease statistics in the STRING network

After mapping the (drugs-related) disease genes to the main STRING network, we observe that every pair of diseases shows overlap. This can range from a modest amount (less than 10 genes) to more than 100 genes. Examples of the latter include type II diabetes, coronary heart disease, obesity, hypertension and bipolar disorder, all of which share a notable background.

In turn, this suggests that some genes might participate in multiple diseases, which is confirmed in figure 97. Several genes participate in 10 or more diseases (out of 22), thus creating an overlap between any pair of them.

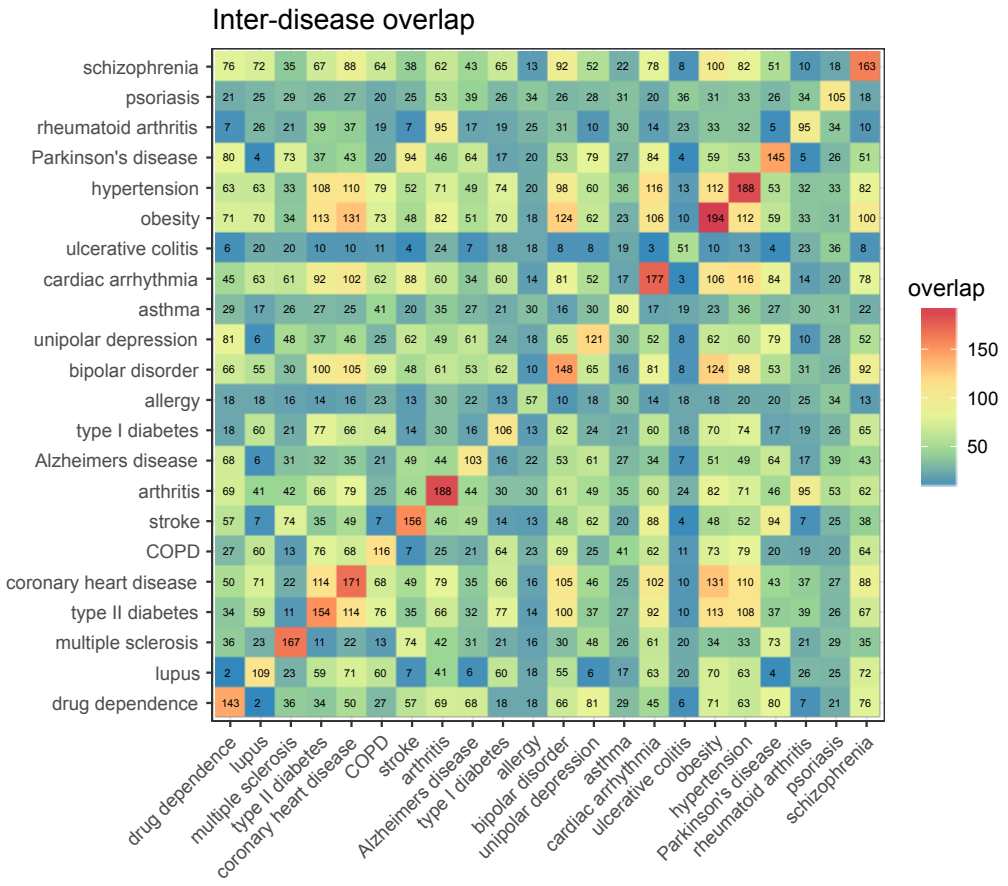


Figure 96: Disease overlap after mapping the genes to the STRING network. There are no disjoint diseases.

The fact that all diseases share at least one gene implies that their distance within the network is always 0. However, we can examine the mean distance between two diseases, defined as the mean of the distance of every pair of genes (g_i, g_j) with g_i belonging to the first disease and g_j to the second. If we group the rows and columns using the UPGMA algorithm (Gronau and Moran, 2007), taking into account that, at the starting point, every disease is in practice equivalent to a cluster of genes.

Each disease, in turn, tends to form a module within the network. To show this, we have computed the modularity of each disease, as implemented in the `modularity` function from the igraph R package (Csardi and Nepusz, 2006). A modularity greater than zero indicates that the number of connections within the disease genes is greater than that of a randomly rewired network. Figure 99 shows how all diseases deviate from their randomised gene sets, which is something expected. Diseases with higher modularity can be easier to predict: an example is cardiac arrhythmia, very modular and well predicted, see the additive models on drugs data.

Another way to examine how close disease genes lie is by representing the mean distance to the disease genes, starting either (1) from the disease genes or (2) from the rest of genes (i.e. non-disease genes for this particular disease). Figure 100 shows how drugs-related genes from a given disease

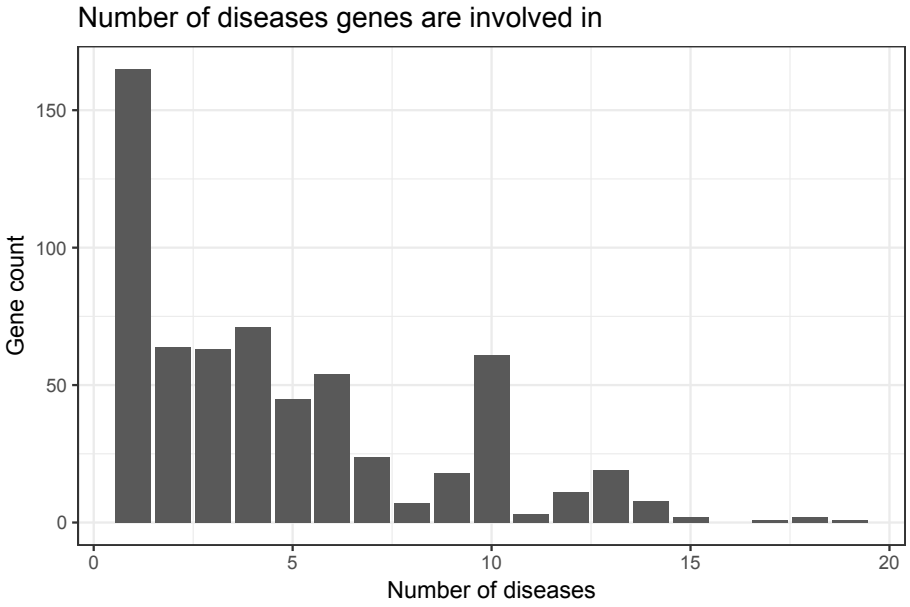


Figure 97: Histogram of the number of diseases genes participate in. The majority of genes belong to a single disease, but a small part of genes are found in 10 or more diseases, unveiling a common core in drug targets.

have a shorter mean distance to themselves than the rest of genes in the network.

Finally, we observe that drugs-related disease genes have larger centrality measures than the rest of nodes in the network (figure 101). This supports the hypothesis that the centrality itself has predictive power, hereby examined by including the PageRank centrality measure as a baseline method.

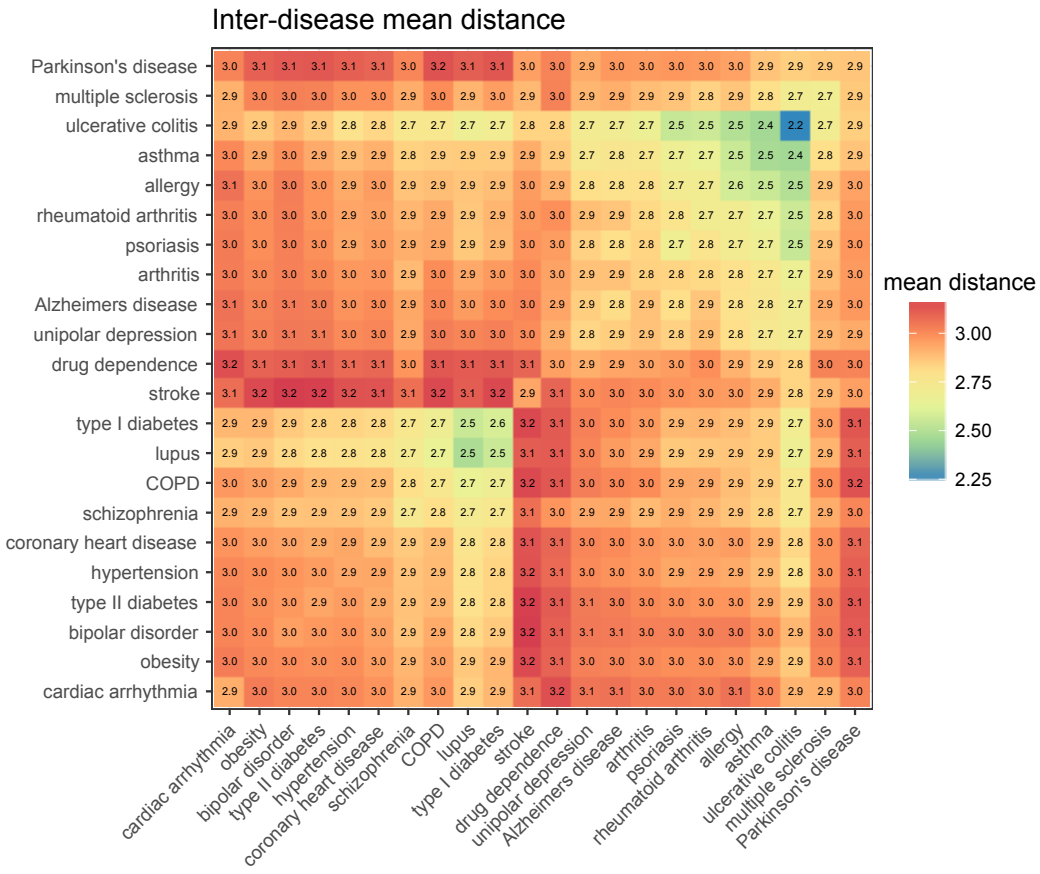


Figure 98: Mean distance between diseases on the STRING network, grouping rows and columns by UPGMA. The lower-left block is coincident with the diseases having a high overlap, as shown in figure 96.

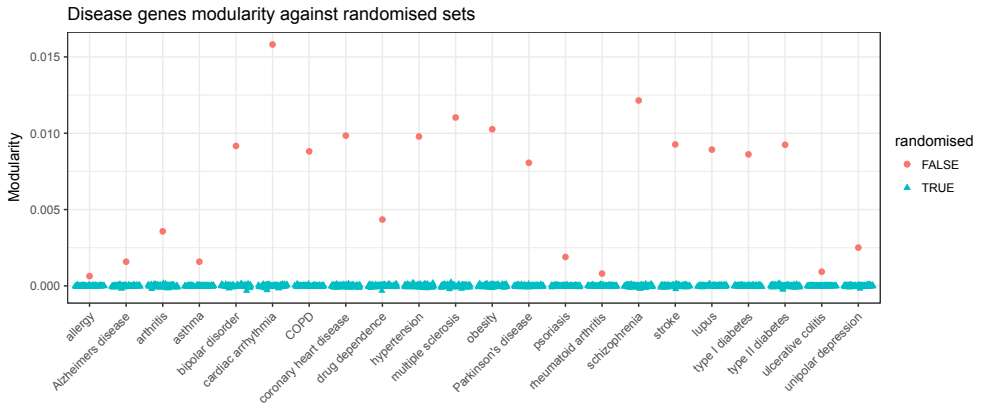


Figure 99: Modularity by disease, compared to randomised trials of the same number of genes. First, we have computed the modularity of the drugs-related disease genes for a given disease, represented through a red dot. Then, we have sampled the same number of genes uniformly from the network, a total of 100 times per disease, and computed their modularity (blue triangles). Random trials lie close to 0, due to the definition of modularity itself, and actual diseases show positive values.

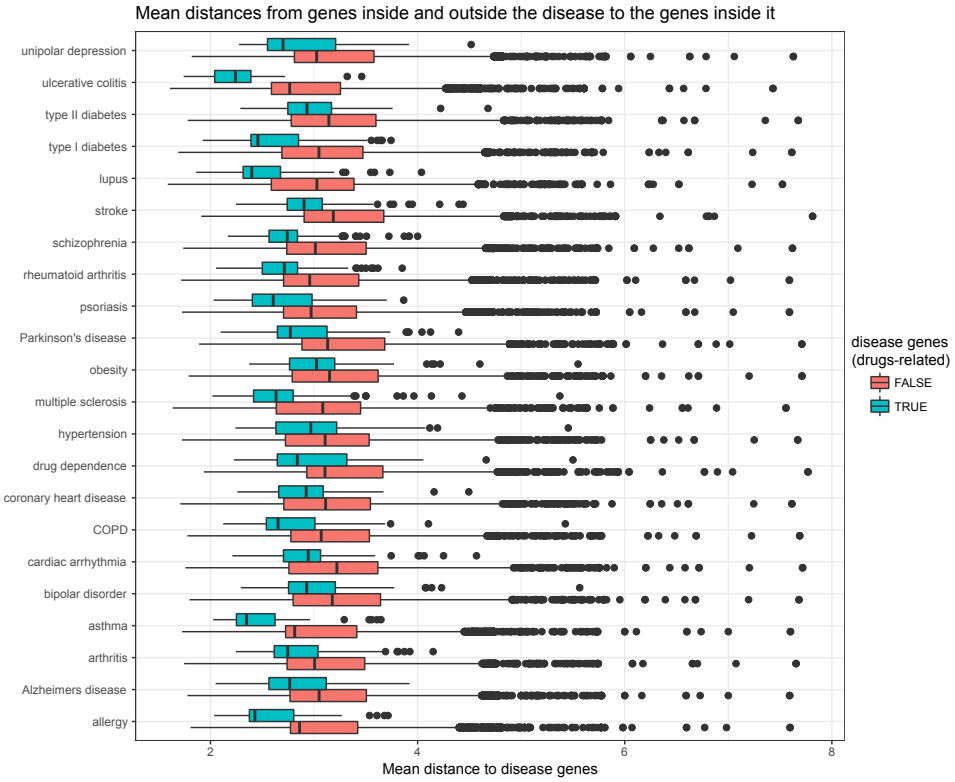


Figure 100: Mean distances to the drugs-related disease genes, computed either from the disease genes (average distance from a disease gene to all the disease genes) or from the non-disease genes (average distance from a non-disease gene to all the disease genes).

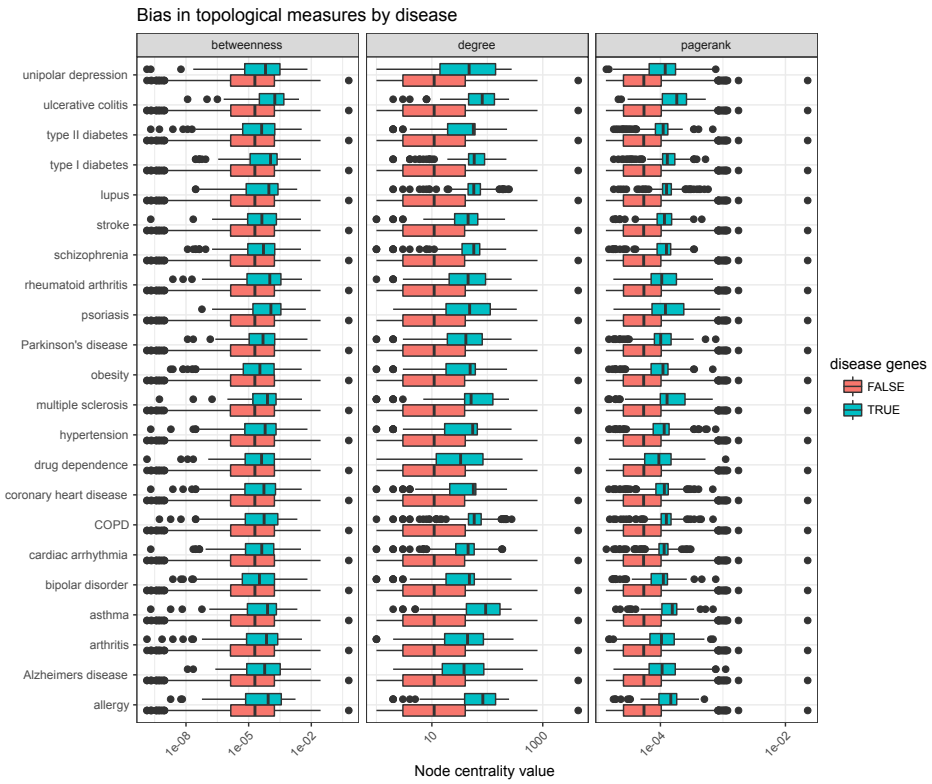


Figure 101: Comparison of the centrality of disease and non-disease genes. We have computed three centrality measures for all the genes in the network: the node degree, the PageRank (as implemented in `page.rank`, with uniform prior and default damping factor $d = 0.85$) in the R package igraph (Csardi and Nepusz, 2006), and the node betweenness, also implemented in the `betweenness` function in igraph. Note that centrality measures are a topological property and do not use disease data as input. For each disease, all the genes have been separated into drugs-related disease genes (blue) and non-disease (red). We can appreciate how, consistently along the three metrics, drugs-related genes tend to have higher centralities.

E.1.5 Complex data

Chembl complex data was retrieved from https://www.ebi.ac.uk/chembl/downloads, specifically release 23 (doi 10.6019/CHEMBL.database.23). The original data comprises 214 complexes with a mean of 9.29 proteins in each and a standard error of 14.08. Having mapped the complexes to the STRING network, 207 non-empty complexes remain, with a mean of 3.251 ENSEMBL ids in each and a standard error of 4.728.

Table 30: Summary statistics of the size (in proteins per complex) of the 214 complexes before and after mapping to the STRING network. Complexes that fail to map have been dropped, hence the differences in their total number N.

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
original	214	9.290	14.080	2	4	6	9	120
mapped to STRING	207	3.251	4.728	1	2	2	3	47
mapped to OmniPath	206	2.981	3.828	1	2	2	3	37

E.1.6 Cross validation splits

Table 31: Number of folds computed for the cross validation in the STRING network. Block cross validation contains slightly less folds because invalid folds have been discarded.

cv_scheme	count
classic	1650
block	1647
representative	1650

Table 32: Summary statistics on the cross validation folds on drugs input. Specifically, on (1) the number of positives in the training fold, (2) positives in the validation fold and (3) number of split complexes. The mean values are outside the parentheses, which contain its standard deviation. We can observe how classic cross validation splits complexes, but none of the complex-aware strategies do. Also, block cross validation can lead to data imbalance, contrary to classic and representative schemes.

	train_pos			validation_pos			split_complexes		
disease	classic	block	representative	classic	block	representative	classic	block	representative
allergy	38.0(0.00)	38.0(1.42)	32.0(0.00)	19.0(0.00)	19.0(1.42)	16.0(0.00)	5.2(1.73)	0.0(0.00)	0.0(0.00)
Alzheimers disease	68.7(0.47)	68.7(4.79)	48.0(0.00)	34.3(0.47)	34.3(4.79)	24.0(0.00)	19.0(3.79)	0.0(0.00)	0.0(0.00)
arthritis	125.3(0.47)	125.3(4.91)	81.3(0.47)	62.7(0.47)	62.7(4.91)	40.7(0.47)	20.5(4.05)	0.0(0.00)	0.0(0.00)
asthma	53.3(0.47)	53.3(1.22)	48.7(0.47)	26.7(0.47)	26.7(1.22)	24.3(0.47)	6.3(3.23)	0.0(0.00)	0.0(0.00)
bipolar disorder	98.7(0.47)	98.7(20.02)	50.0(0.00)	49.3(0.47)	49.3(20.02)	25.0(0.00)	16.8(3.34)	0.0(0.00)	0.0(0.00)
cardiac arrhythmia	118.0(0.00)	118.0(22.54)	59.3(0.47)	59.0(0.00)	59.0(22.54)	29.7(0.47)	17.6(3.68)	0.0(0.00)	0.0(0.00)
COPD	77.3(0.47)	77.3(21.00)	44.7(0.47)	38.7(0.47)	38.7(21.00)	22.3(0.47)	6.9(3.34)	0.0(0.00)	0.0(0.00)
coronary heart disease	114.0(0.00)	114.0(19.94)	57.3(0.47)	57.0(0.00)	57.0(19.94)	28.7(0.47)	19.8(3.35)	0.0(0.00)	0.0(0.00)
drug dependence	95.3(0.47)	95.3(10.61)	58.7(0.47)	47.7(0.47)	47.7(10.61)	29.3(0.47)	24.8(4.46)	0.0(0.00)	0.0(0.00)
hypertension	125.3(0.47)	125.3(18.51)	70.7(0.47)	62.7(0.47)	62.7(18.51)	35.3(0.47)	24.7(4.82)	0.0(0.00)	0.0(0.00)
multiple sclerosis	111.3(0.47)	111.3(9.62)	74.0(0.00)	55.7(0.47)	55.7(9.62)	37.0(0.00)	11.5(2.00)	0.0(0.00)	0.0(0.00)
obesity	129.3(0.47)	129.3(16.64)	69.3(0.47)	64.7(0.47)	64.7(16.64)	34.7(0.47)	20.3(3.80)	0.0(0.00)	0.0(0.00)
Parkinson's disease	96.7(0.47)	96.7(3.73)	77.3(0.47)	48.3(0.47)	48.3(3.73)	38.7(0.47)	17.3(3.91)	0.0(0.00)	0.0(0.00)
psoriasis	70.0(0.00)	70.0(3.18)	53.3(0.47)	35.0(0.00)	35.0(3.18)	26.7(0.47)	17.6(3.87)	0.0(0.00)	0.0(0.00)
rheumatoid arthritis	63.3(0.47)	63.3(2.30)	51.3(0.47)	31.7(0.47)	31.7(2.30)	25.7(0.47)	7.3(2.29)	0.0(0.00)	0.0(0.00)
schizophrenia	108.7(0.47)	108.7(20.06)	46.7(0.47)	54.3(0.47)	54.3(20.06)	23.3(0.47)	9.1(1.94)	0.0(0.00)	0.0(0.00)
stroke	104.0(0.00)	104.0(8.53)	66.0(0.00)	52.0(0.00)	52.0(8.53)	33.0(0.00)	18.1(3.23)	0.0(0.00)	0.0(0.00)
lupus	72.7(0.47)	71.7(22.34)	30.7(0.47)	36.3(0.47)	37.3(22.34)	15.3(0.47)	5.8(1.71)	0.0(0.00)	0.0(0.00)
type I diabetes mellitus	70.7(0.47)	70.2(22.46)	32.7(0.47)	35.3(0.47)	35.8(22.46)	16.3(0.47)	5.7(1.78)	0.0(0.00)	0.0(0.00)
type II diabetes mellitus	102.7(0.47)	102.7(17.99)	54.7(0.47)	51.3(0.47)	51.3(17.99)	27.3(0.47)	15.6(3.43)	0.0(0.00)	0.0(0.00)
ulcerative colitis	34.0(0.00)	34.0(1.68)	27.3(0.47)	17.0(0.00)	17.0(1.68)	13.7(0.47)	5.1(1.70)	0.0(0.00)	0.0(0.00)
unipolar depression	80.7(0.47)	80.7(3.49)	59.3(0.47)	40.3(0.47)	40.3(3.49)	29.7(0.47)	22.9(4.20)	0.0(0.00)	0.0(0.00)

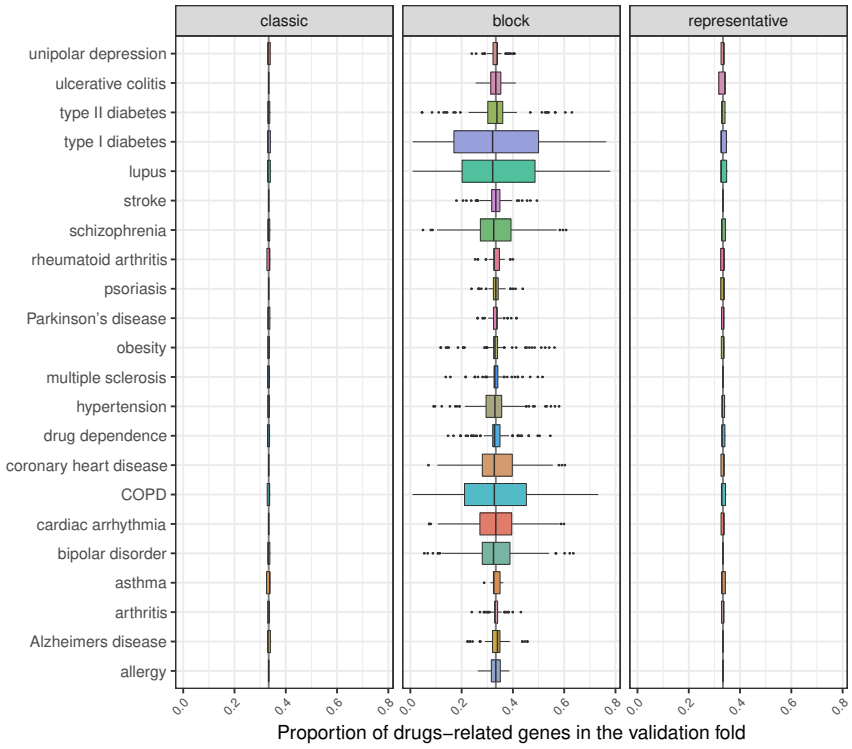


Figure 102: Data balance by cross-validation strategy. Each boxplot summarises the folds for a particular disease and cross-validation strategy, whilst the vertical grey line corresponds to the theoretical balanced proportion. Due to their definition, the classic and representative strategies keep the dataset balanced: one third of the drugs-related genes are used to validate and two thirds are used as seed genes. Inevitably, small deviations arise if the total number of disease genes is not a multiple of 3. Note, however, how the block scheme sometimes keeps the balance (diseases such as asthma and Parkinson's disease), but can lead to data imbalance if large complexes are involved, like in COPD and type I diabetes.

E.2 RAW METRICS PLOTS

E.2.1 By method

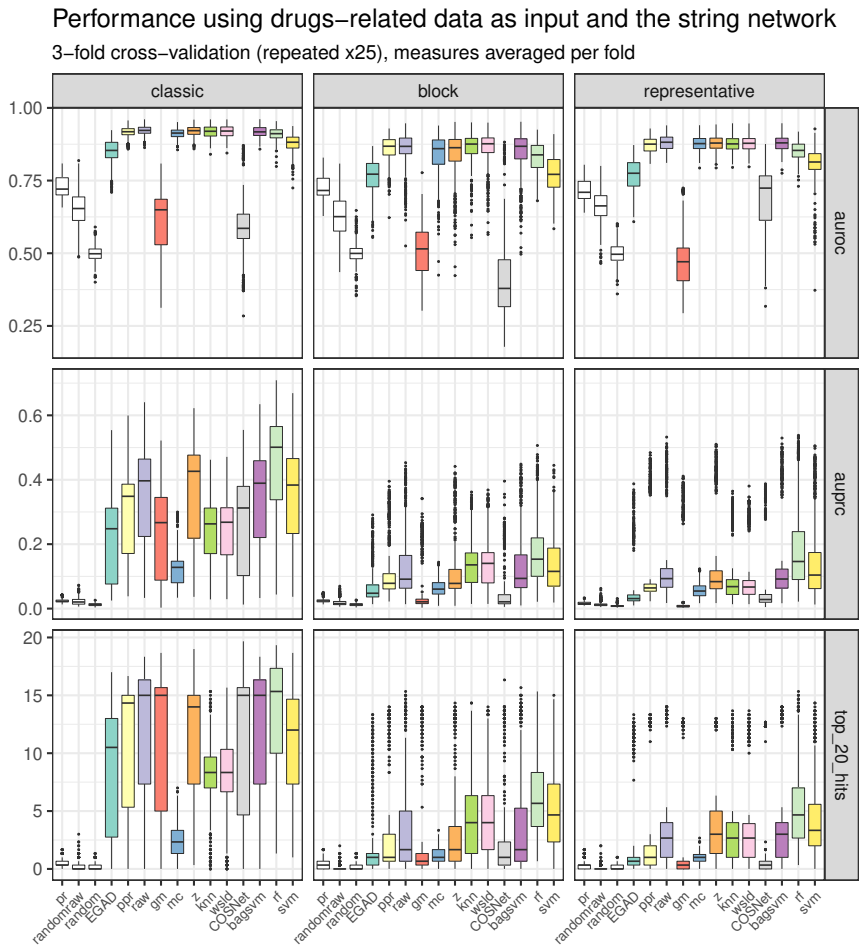


Figure 103: Performance by method using drugs input and the STRING network. Methods `pr`, `randomraw` and `random` have no fill colour to represent their "null model" role.

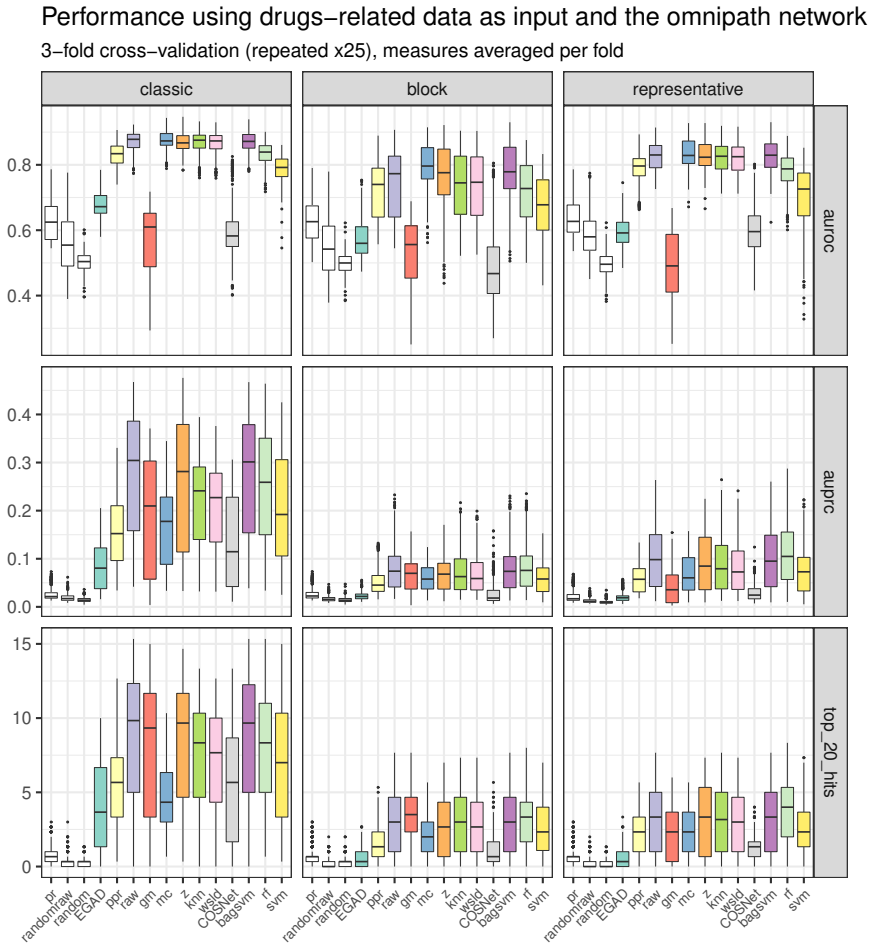


Figure 104: Performance by method using drugs input and the OmniPath network. Methods `pr`, `randomraw` and `random` have no fill colour to represent their "null model" role.

E.2.2 By disease

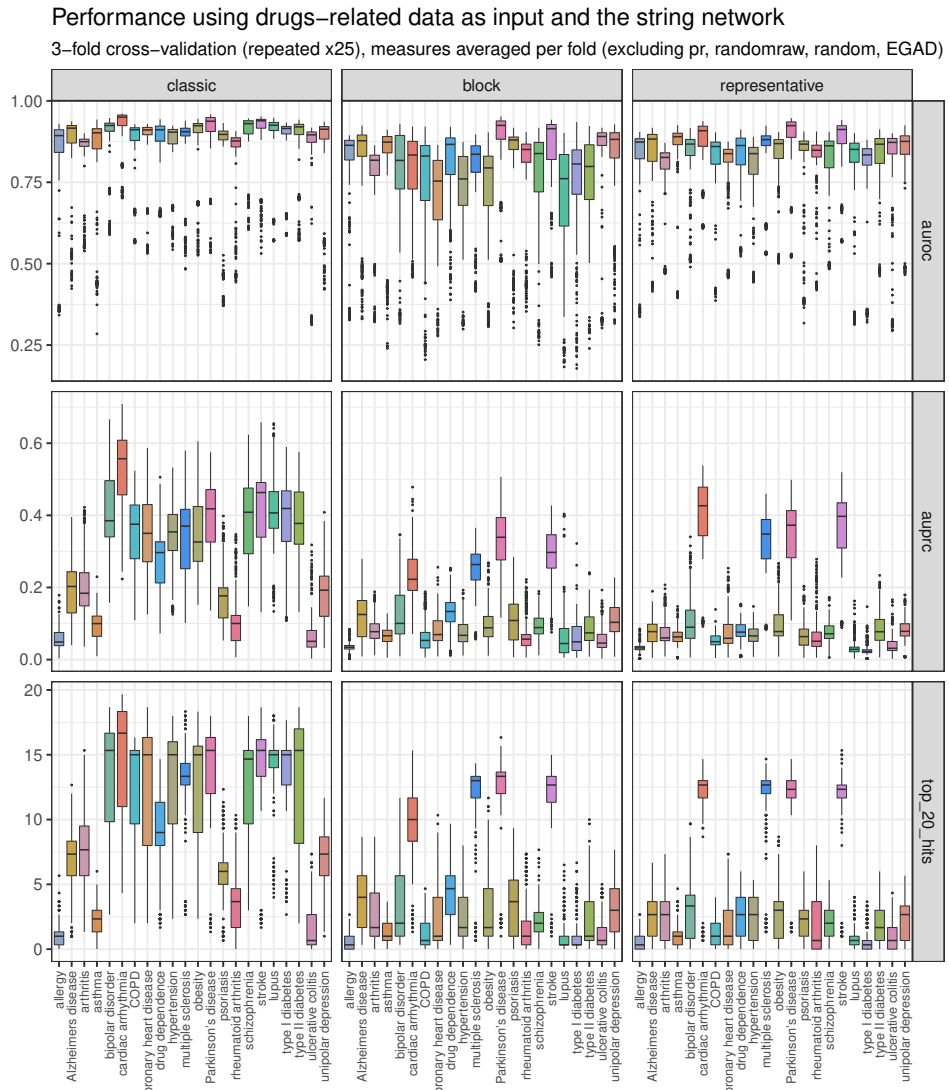


Figure 105: Performance by disease using drugs input and the STRING network. Baseline methods `pr`, `randomraw`, `random` and `EGAD` are left out for visual clarity.

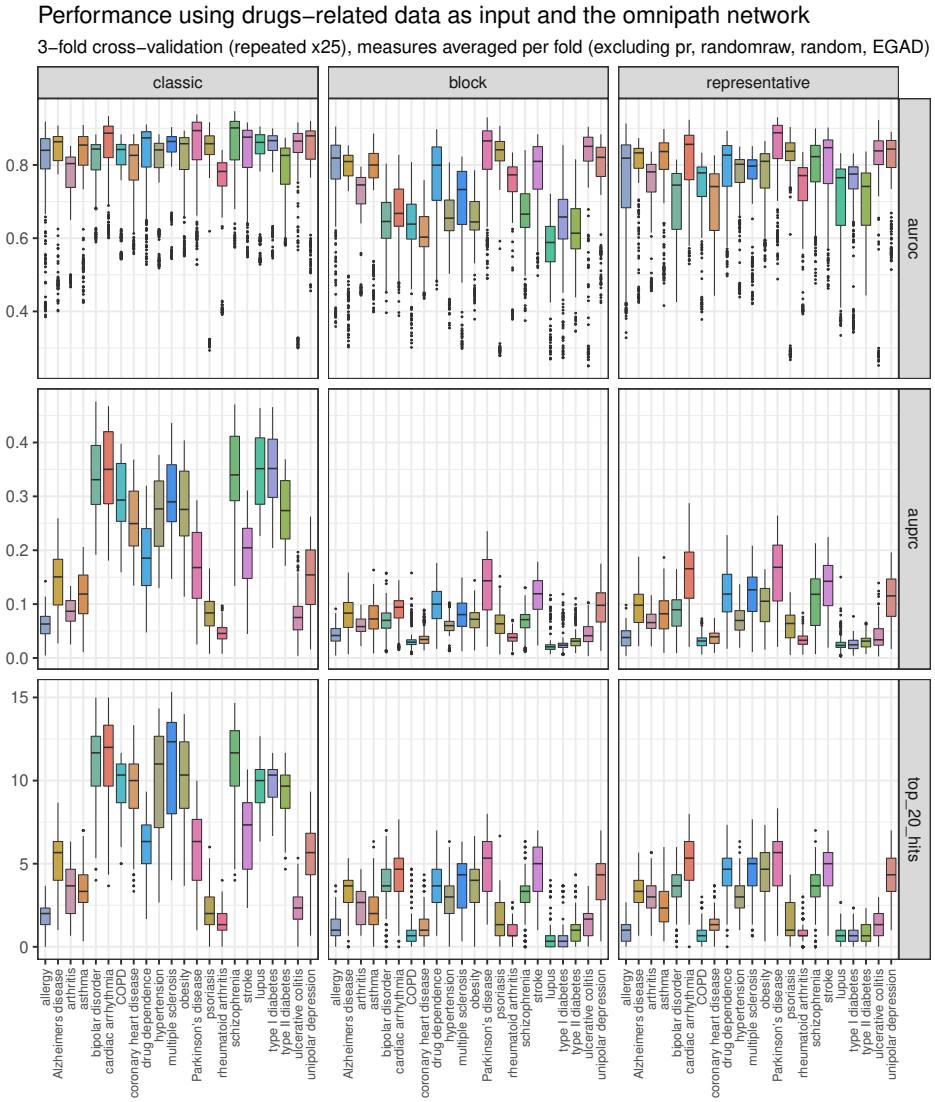


Figure 106: Performance by disease using drugs input and the OmniPath network. Baseline methods `pr`, `randomraw`, `random` and `EGAD` are left out for visual clarity.

E.2.3 Overall performance by disease

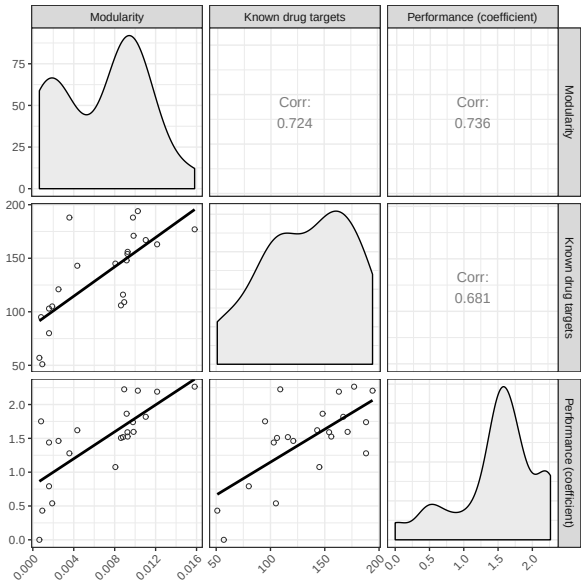


Figure 107: Pairs plot involving disease-level performance (top 20 hits), number of known drug targets and disease modularity (in STRING). The three magnitudes correlate positively, implying that in general diseases with (i) more drugs-related genes and (ii) higher modularity in the network used by the prioritisers will exhibit better performance. Likewise, diseases with larger gene lists tend to be more modular.

E.3 NETWORK-BASED METHODS

E.3.1 Method details

All tests and batch runs were set-up and conducted using the R statistical programming language (R Core Team, 2016). When no R package was available, the methodology was re-implemented, building upon existing R packages whenever possible. Standard R machine learning libraries were used to train the support vector machine and random forest classifiers. Only the MashUp algorithm (Cho et al., 2016) required feature generation outside of the R environment, using the Matlab code from their publication. The versions of the R packages can be found in table 47.

EGAD (Extending "Guilt by Association" by Degree (Ballouz et al., 2017)) was used here as a baseline comparator. EGAD performs a naïve diffusion approach via near-neighbours voting since EGAD's *neighbor_voting* function uses the adjacency matrix of the network and no additional parameters.

PageRank is a standard web ranking technology based upon the original work of Page et al. (Page et al., 1999). The igraph R package implementation of PageRank (here `ppr`) was used with default damping factor, `d = 0.85`. The latter implements what is commonly referred to as personalised PageRank, because of the custom prior distribution. This prior gives a probability of $1/n_{input}$ to each input gene and 0 otherwise, and forces random walks to start from the input genes. PageRank has been employed to diffuse disease seeding information across a two-layered network comprising PPI and GO hierarchy information (Jiang et al., 2017). Two approaches were developed: BirgRank (applying traditional PageRank with fixed decay parameters) and AptRank (with an adaptive diffusion mechanism). Here we considered only fixed decay parameter PageRank diffusion on the regular, weighted PPI network.

The diffuStats Bioconductor package (Picart-Armada, Thompson, et al., 2017) implements a variety of diffusion kernels and scoring schemes. Here we employed the regularised Laplacian kernel with the following diffusion propagation scores, as summarised in (Picart-Armada, Thompson, et al., 2017): `raw`, `gm` (Genemania-based weighting for positives, negatives and unlabelled nodes), `mc` and `z`. `raw` comes from (Vandin et al., 2011), while `gm` uses the weighting scheme from (Mostafavi et al., 2008). `mc` was inspired in (Bersanelli et al., 2016) and `z` is an exact version of (Erten et al., 2011) without controlling for degree; both have been used for the enrichment of metabolomics data (Picart-Armada, Fernández-Albert, et al., 2017). The regularised Laplacian kernel had the following (default) parameters: `sigma2 = 1`, `add_diag = 1`, `normalized = FALSE`. The weights from the network are scaled to lie in $[0, 1]$.

RANKS (RAnking of Nodes with Kernelized Score functions (Valentini, Armano, et al., 2016)) employs kernelised score functions in semi-supervised learning (here `knn` and `wsld`), and has been assessed for disease gene identification (Valentini, Paccanaro, et al., 2014). Default package settings were used in all cases (number of neighbours, $k = 3$ for k-nn and coefficient of linear decay, `d = 2` for `wsld`). We used the kernel computed with diffuStats.

The bagging SVM method (here `bagsvm`) is an implementation of ProDiGe1 (Mordelet and Vert, 2011). It approximates a form of PU-learning (Elkan and Noto, 2008; Yang et al., 2012) by iteratively choosing random subsets from the unlabelled genes (i.e. those genes that are not known to be associated with the disease) when training classifiers. This method was directly applied to the regularised Laplacian kernel computed with diffuStats.

The `svm` (Support Vector Machine; kernlab R package) and `rf` (randomForest R package) methods apply classical machine learning approaches on network-based features. Network-based features were generated using MashUp with default parameters for the human network (800-dimensional, as recommended by the authors) (Cho et al., 2016). We used the caret (Kunn, 2008) and mlr (Bischl et al., 2016) R packages to define the classification tasks, grid-search the parameters and make predictions for these two methods.

The SVM method used here is a nu-SVM with RBF kernel. In training, the negative class examples were randomly under-sampled to match the number of positive class examples. Parameters were determined via inner cross-validation with parameter ranges of $(0.1, 0.9)$ for `nu` and $(10^{-6}, 10^{2})$ for `sigma`, with search space linear on `nu` and logarithmic on `sigma`. A grid of resolution 5 in each direction was explored to choose the best parameters, with an internal loop of 3 repetitions of 3-fold CV.

Random forest parameters were set to default values (see mlr documentation on *classif.randomForest*) apart from those tuned via inner cross validation. These were ranges of $(10, 500)$ for `ntree` and $(1, 5)$ for `nodesize`, with linear search space in both. A grid of resolution 3 in each direction was explored to choose optimal parameters with an internal loop of 3 repetitions of standard 3-fold CV.

COSNet (COst Sensitive neural Network (Bertoni et al., 2011; Frasca et al., 2013)) consists of a parametric Hopfield recurrent neural network classifier, employed within a semi-supervised, cost-sensitive learning context to deal with networks seeded with highly unbalanced labellings. The cost (regularisation) parameter in the COSNet R package was set to 0.0001 following the documentation guidelines.

Finally, we included the following three naive baseline methods, for comparison purposes: (1) `pr`, a classic problem naïve 'non-personalised' PageRank implementation where input scores on the genes are ignored; (2) `randomraw`, which applies the `raw` diffusion approach from diffuStats (Picart-Armada, Thompson, et al., 2017) to randomly permuted input scores on the genes; and (3) `random`, a uniform re-ranking of input genes without any network propagation (using *sample(n)* in R, with `n` = number of genes in the test fold).

E.3.2 Comparing methods

Comparing methods using their predictions on seed and novel genes can give insights on similarities and differences among the various methods families. The main body contains a comparison using the drugs genes as seeds and excluding the seeds for the comparison, while here we also include the seeds (figure 108) and an analogous analysis on the genetically associated genes (figure 109). Classical MDS plots for specific diseases can be found in the supplementary file `mds_plots.zip` and are qualitatively consistent with their multiview counterparts.

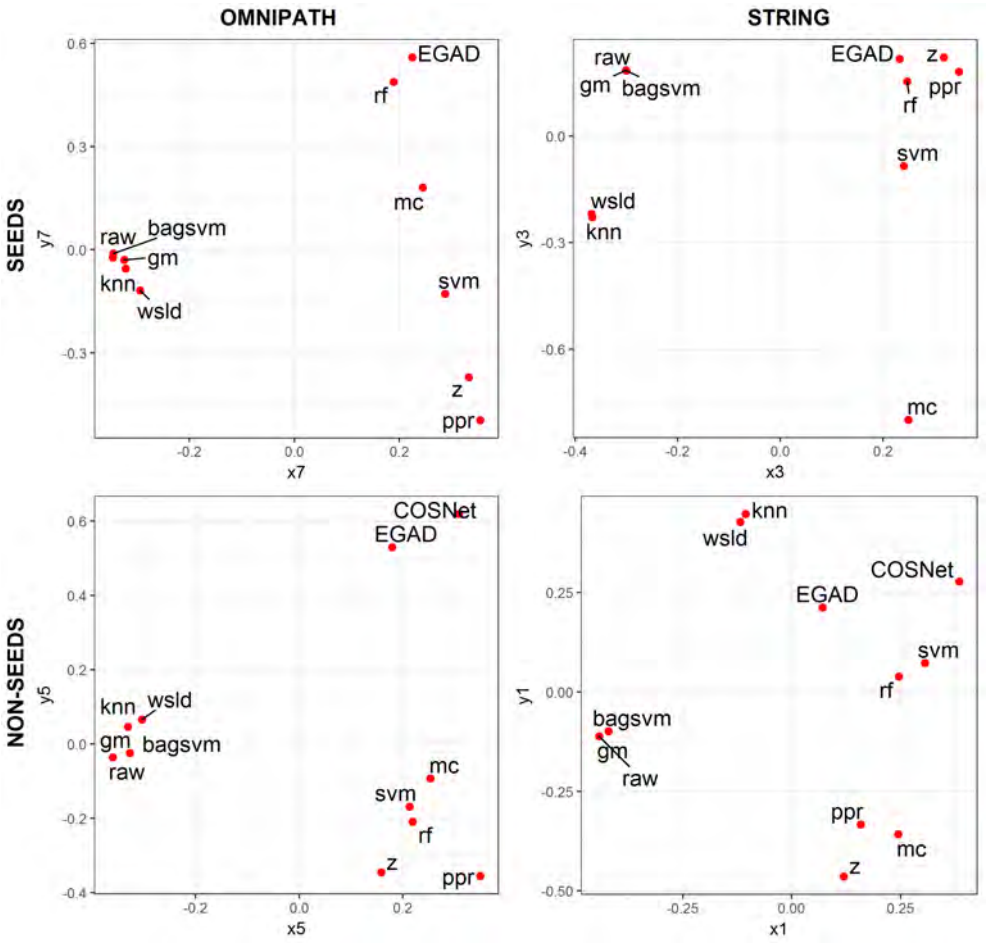


Figure 108: Multi-view MDS plot displaying the preserved Spearman's footrule distances representing the differential ranking behaviours of methods across all 22 diseases when individual sets of drugs seeds were input. Each plot is for a different combination of input network (columns) and the predicted gene set that was ranked (rows). Note how COSNet is excluded from seeds prediction, as by its definition it does not order the seeds.

In figure 108, two groups of diffusion-based methods consistently clustered together: (i) `raw`, `gm`, `bagsvm`, and (ii) `knn` and `wsld`. As a consequence, the supervised, bagged SVM based in the regularised Laplacian kernel behaved like usual diffusion scores (`raw`) that use the same kernel. Despite their common background, groups (i) and (ii) appeared together in OmniPath but not in STRING, implying that even small methodological differ-

ences can have a noticeable overall impact. A third group (iii) was formed by `ppr`, `z` and `mc`, although the latter did not cluster as clearly in some cases. These methods are also diffusion-based, but `mc` and `z` have statistical differences with `raw` (Picart-Armada, Thompson, et al., 2017) and this is reflected in the MDS plot. The supervised methods (iv) `rf` and `svm` also tended to agree, since they were trained on the same network-based features. Finally, (v) `EGAD` and `COSNet` closes the method grouping, suggesting that the artificial neural network from `COSNet` resembled neighbour voting approaches.

In figure 109, groups (i) and (ii) stick together in all the scenarios and become one single family. Group (iii) is only obvious in STRING and non-seed genes, becoming diluted in the rest.

The tight five method group (`raw`, `gm`, `bagsvm`, `knn` and `wsld`), seen only for OmniPath with input drug seeds in the main body, is apparent with both networks. Group (iv) and (v) still behaves as so, further justifying this classification. Despite some differences, these groupings do agree to those seen for the corresponding networks under drug seed input in figure 108.

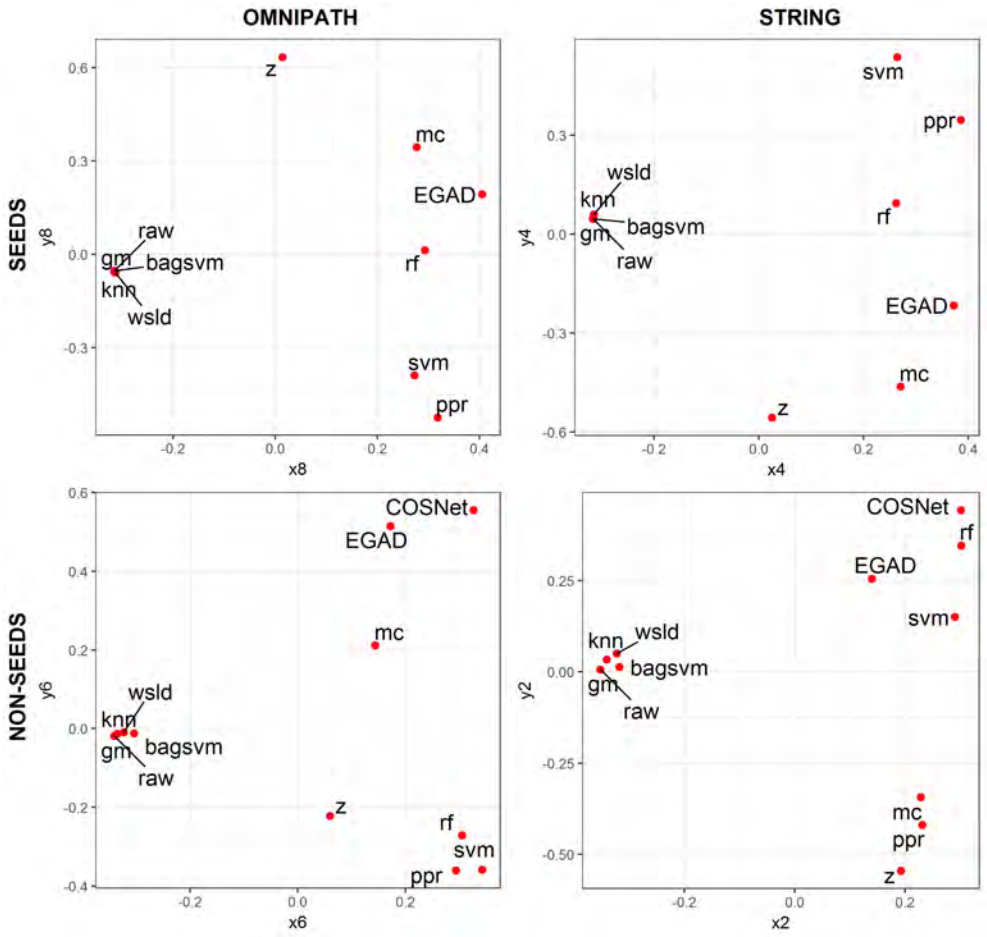


Figure 109: Multi-view MDS plot displaying the preserved Spearman's footrule distances representing the differential ranking behaviours of methods across all 22 diseases when individual sets of genetic seeds were input.

E.3.3 Methods ranking using all the metrics

In the main text we show the method prioritisations using the main metrics. Figure 110 contains the same data for all the metrics hereby analysed. The metrics have been arranged from farthest (top 20 hits) to closest to AUROC. Two conclusions can be drawn from figure 110. First, AUROC behaves differently from the other five metrics, which in turn behave alike. This is expected as AUPRC, pAUROC and top k hits emphasise on the performance at the top ranked entities. Second, as the parameter of pAUROC and top k hits grows, both metrics rank closer to AUROC, which is also natural.

The fact that top 20 hits, top 100 hits and AUPRC behave so similarly suggests that the ranking under top k hits is robust for small values of k ($k \leqslant 100$) and that AUPRC is indeed a meaningful performance metric for real scenarios in drug development.

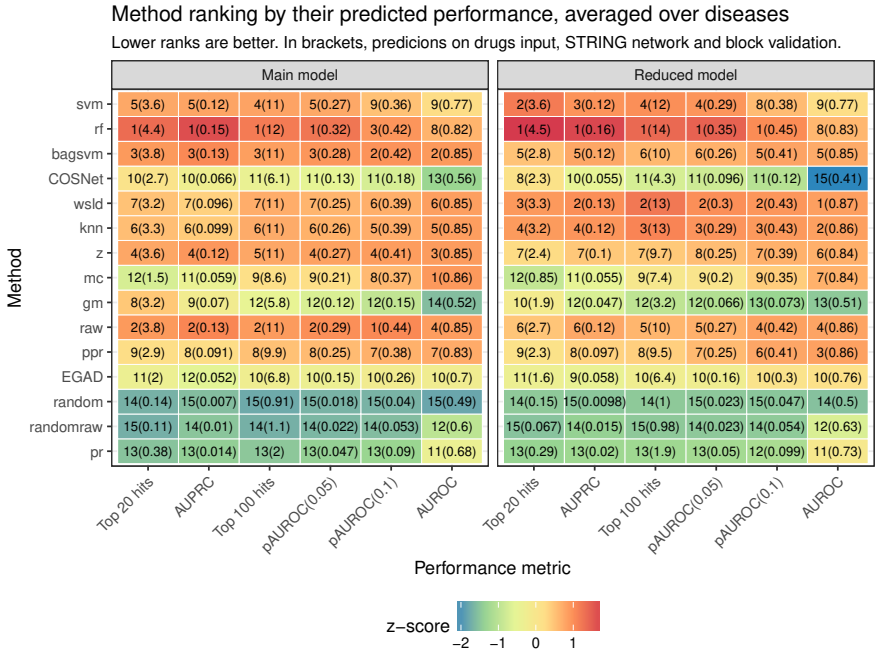


Figure 110: Ranking of all the methods, using the predictions of the main and the reduced models on the drugs input, STRING network, block cross validation and averaging over diseases. A column-wise z-score on the predicted mean is depicted, in order to illustrate the magnitude of the difference.

E.4 MODEL SUMMARIES AND CONFIDENCE INTERVALS

E.4.1 Model description

Table 33: Summary of all the complete models fitted in this study. Models are adjusted separately by input type, not to mix notably different patterns. The model formulae are R-like, where the left hand side contains the response and the right hand side describes the independent variables. In all these models, the reference levels are the **pr** method, **STRING** network, **classic** cross-validation scheme and **allergy**. Statistical significance on each coefficient is computed by comparing the full model with the model that lacks such regressor.

Model	Input type	Model type	Family	Formula
DA1	Drugs	Additive	Quasibinomial	AUROC ~ method + cv_scheme + network + disease
DA2	Drugs	Additive	Quasibinomial	AUPRC ~ method + cv_scheme + network + disease
DA3	Drugs	Additive	Quasipoisson	Top20 ~ method + cv_scheme + network + disease
DA4	Drugs	Additive	Quasibinomial	pAUROC0.1~ method + cv_scheme + network + disease
DA5	Drugs	Additive	Quasibinomial	pAUROC0.05 ~ method + cv_scheme + network + disease
DA6	Drugs	Additive	Quasipoisson	Top100 ~ method + cv_scheme + network + disease
GA1	Genetic	Additive	Quasibinomial	AUROC ~ method + cv_scheme + network + disease
GA2	Genetic	Additive	Quasibinomial	AUPRC ~ method + cv_scheme + network + disease
GA3	Genetic	Additive	Quasipoisson	Top20 ~ method + cv_scheme + network + disease
GA4	Genetic	Additive	Quasibinomial	pAUROC0.1~ method + cv_scheme + network + disease
GA5	Genetic	Additive	Quasibinomial	pAUROC0.05 ~ method + cv_scheme + network + disease
GA6	Genetic	Additive	Quasipoisson	Top100 ~ method + cv_scheme + network + disease
SA1	Stream	Additive	Quasibinomial	AUROC ~ method + cv_scheme + network + disease
SA2	Stream	Additive	Quasibinomial	AUPRC ~ method + cv_scheme + network + disease
SA3	Stream	Additive	Quasipoisson	Top20 ~ method + cv_scheme + network + disease
SA4	Stream	Additive	Quasibinomial	pAUROC0.1~ method + cv_scheme + network + disease
SA5	Stream	Additive	Quasibinomial	pAUROC0.05 ~ method + cv_scheme + network + disease
SA6	Stream	Additive	Quasipoisson	Top100 ~ method + cv_scheme + network + disease

Table 34: Summary of all the reduced models. These additive models have been fitted to the most relevant scenario: **drugs** input, **STRING** network and **block** cross-validation strategy. In all these models, the reference levels are the **pr** method and **allergy**.

Model	Input type	Model type	Family	Formula
DA1r	Drugs	Additive	Quasibinomial	AUROC ~ method + disease
DA2r	Drugs	Additive	Quasibinomial	AUPRC ~ method + disease
DA3r	Drugs	Additive	Quasipoisson	Top20 ~ method + disease
DA4r	Drugs	Additive	Quasibinomial	pAUROC0.1~ method + disease
DA5r	Drugs	Additive	Quasibinomial	pAUROC0.05 ~ method + disease
DA6r	Drugs	Additive	Quasipoisson	Top100 ~ method + disease

E.4.2 Drugs input

Additive models

Table 35: Models for the metrics auroc, auprc, top_20_hits using the drugs input (model names DA1, DA2 and DA3)

	auroc	auprc	top_20_hits
Constant	1.264*** (1.243, 1.285)	−4.286*** (−4.346, −4.227)	−1.462*** (−1.539, −1.385)
methodrandomraw	−0.328*** (−0.345, −0.312)	−0.308*** (−0.372, −0.244)	−1.223*** (−1.334, −1.112)
methodrandom	−0.773*** (−0.790, −0.757)	−0.685*** (−0.756, −0.613)	−0.994*** (−1.096, −0.892)
methodEGAD	0.122*** (0.105, 0.139)	1.358*** (1.310, 1.406)	1.662*** (1.604, 1.720)
methodppr	0.861*** (0.842, 0.880)	1.964*** (1.917, 2.010)	2.022*** (1.965, 2.078)
methodraw	0.990*** (0.970, 1.009)	2.352*** (2.306, 2.397)	2.299*** (2.244, 2.355)
methodgm	−0.652*** (−0.668, −0.636)	1.681*** (1.634, 1.728)	2.126*** (2.070, 2.182)
methodmc	1.044*** (1.024, 1.064)	1.488*** (1.440, 1.536)	1.376*** (1.317, 1.436)
methodz	1.005*** (0.986, 1.025)	2.286*** (2.241, 2.332)	2.253*** (2.197, 2.308)
methodknn	0.981*** (0.962, 1.001)	2.060*** (2.014, 2.106)	2.162*** (2.106, 2.217)
methodwsld	0.976*** (0.956, 0.995)	2.028*** (1.982, 2.074)	2.148*** (2.092, 2.204)
methodCOSNet	−0.511*** (−0.527, −0.494)	1.615*** (1.568, 1.662)	1.962*** (1.906, 2.019)
methodbagsvm	1.028*** (1.008, 1.048)	2.337*** (2.292, 2.383)	2.299*** (2.243, 2.354)
methodrf	0.782*** (0.763, 0.801)	2.569*** (2.524, 2.615)	2.454*** (2.399, 2.509)
methodsvm	0.462*** (0.445, 0.480)	2.233*** (2.187, 2.279)	2.246*** (2.190, 2.302)
cv_schemeblock	−0.441*** (−0.450, −0.433)	−1.243*** (−1.256, −1.230)	−0.984*** (−0.997, −0.970)
cv_schemerepresentative	−0.218*** (−0.227, −0.210)	−1.182*** (−1.195, −1.169)	−1.003*** (−1.017, −0.990)
networkomnipath	−0.392*** (−0.399, −0.385)	−0.517*** (−0.528, −0.506)	−0.357*** (−0.367, −0.346)
diseaseAlzheimers disease	−0.001 (−0.024, 0.022)	1.081*** (1.032, 1.131)	1.439*** (1.377, 1.500)
diseasearthritis	−0.192*** (−0.214, −0.169)	0.846*** (0.795, 0.897)	1.279*** (1.216, 1.341)
diseaseasthma	0.012 (−0.011, 0.035)	0.671*** (0.618, 0.723)	0.792*** (0.725, 0.859)
diseasebipolar disorder	−0.211*** (−0.234, −0.188)	1.652*** (1.604, 1.699)	1.864*** (1.805, 1.924)
diseasecardiac arrhythmia	0.007 (−0.016, 0.030)	2.291*** (2.245, 2.337)	2.264*** (2.206, 2.322)
diseaseCOPD	−0.188*** (−0.210, −0.165)	1.301*** (1.252, 1.350)	1.519*** (1.457, 1.580)
diseasecoronary heart disease	−0.299*** (−0.321, −0.276)	1.299*** (1.250, 1.348)	1.596*** (1.535, 1.656)
diseasedrug dependence	−0.018 (−0.041, 0.005)	1.356*** (1.308, 1.405)	1.620*** (1.559, 1.681)
diseasehypertension	−0.207*** (−0.230, −0.185)	1.372*** (1.324, 1.421)	1.739*** (1.679, 1.799)
diseasemultiple sclerosis	−0.071*** (−0.094, −0.048)	1.970*** (1.923, 2.017)	2.225*** (2.167, 2.283)
diseaseobesity	−0.164*** (−0.186, −0.141)	1.506*** (1.458, 1.554)	1.819*** (1.759, 1.878)
diseaseParkinson's disease	0.207*** (0.184, 0.231)	2.080*** (2.034, 2.127)	2.205*** (2.147, 2.263)
diseasepsoriasis	0.083*** (0.059, 0.106)	0.856*** (0.804, 0.907)	1.076*** (1.012, 1.140)
diseaserheumatoid arthritis	−0.163*** (−0.185, −0.140)	0.355*** (0.300, 0.410)	0.539*** (0.469, 0.609)
diseaseschizophrenia	−0.083*** (−0.106, −0.060)	1.603*** (1.555, 1.651)	1.752*** (1.692, 1.812)
diseasestroke	0.085*** (0.062, 0.109)	2.057*** (2.011, 2.104)	2.190*** (2.132, 2.249)
diseaselupus	−0.248*** (−0.270, −0.225)	1.396*** (1.347, 1.444)	1.526*** (1.465, 1.587)
diseasetype I diabetes	−0.174*** (−0.196, −0.151)	1.364*** (1.316, 1.413)	1.506*** (1.445, 1.568)
diseasetype II diabetes	−0.252*** (−0.274, −0.229)	1.373*** (1.324, 1.421)	1.590*** (1.530, 1.651)
diseaseulcerative colitis	0.132*** (0.108, 0.155)	0.251*** (0.195, 0.308)	0.431*** (0.360, 0.502)
diseaseunipolar depression	0.022* (−0.002, 0.045)	1.113*** (1.064, 1.163)	1.462*** (1.400, 1.523)
Observations	49,500	49,500	49,500

Note: *p<0.1; **p<0.05; ***p<0.01

Table 36: Predictions of the models DA1, DA2, DA3 (95% confidence intervals after averaging over disease).

Input: drugs data		STRING			OmniPath		
metric	method	classic	block	representative	classic	block	representative
	pr	(0.764, 0.768)	(0.675, 0.681)	(0.722, 0.727)	(0.686, 0.691)	(0.584, 0.590)	(0.637, 0.643)
	randomraw	(0.699, 0.705)	(0.599, 0.606)	(0.652, 0.657)	(0.611, 0.617)	(0.503, 0.509)	(0.558, 0.565)
	random	(0.599, 0.605)	(0.490, 0.496)	(0.545, 0.552)	(0.502, 0.508)	(0.393, 0.399)	(0.448, 0.454)
	EGAD	(0.785, 0.789)	(0.701, 0.707)	(0.746, 0.751)	(0.711, 0.717)	(0.613, 0.620)	(0.665, 0.671)
	ppr	(0.884, 0.887)	(0.831, 0.835)	(0.860, 0.864)	(0.837, 0.842)	(0.768, 0.774)	(0.805, 0.810)
	raw	(0.896, 0.900)	(0.848, 0.852)	(0.874, 0.878)	(0.854, 0.858)	(0.790, 0.795)	(0.825, 0.829)
	gm	(0.627, 0.633)	(0.520, 0.526)	(0.575, 0.581)	(0.532, 0.538)	(0.423, 0.429)	(0.478, 0.484)
auroc	mc	(0.901, 0.904)	(0.855, 0.859)	(0.880, 0.884)	(0.861, 0.865)	(0.799, 0.804)	(0.832, 0.837)
	z	(0.898, 0.901)	(0.850, 0.854)	(0.876, 0.880)	(0.856, 0.860)	(0.793, 0.798)	(0.827, 0.832)
	knn	(0.896, 0.899)	(0.847, 0.851)	(0.873, 0.877)	(0.853, 0.857)	(0.789, 0.794)	(0.823, 0.828)
	wsld	(0.895, 0.898)	(0.846, 0.850)	(0.873, 0.877)	(0.852, 0.856)	(0.788, 0.793)	(0.823, 0.827)
	COSNet	(0.660, 0.666)	(0.555, 0.561)	(0.609, 0.615)	(0.567, 0.573)	(0.457, 0.464)	(0.513, 0.519)
	bagsvm	(0.900, 0.903)	(0.853, 0.857)	(0.879, 0.882)	(0.859, 0.863)	(0.796, 0.802)	(0.830, 0.835)
	rf	(0.876, 0.879)	(0.819, 0.824)	(0.850, 0.854)	(0.826, 0.831)	(0.754, 0.760)	(0.793, 0.798)
	svm	(0.837, 0.841)	(0.767, 0.772)	(0.805, 0.809)	(0.776, 0.781)	(0.690, 0.696)	(0.736, 0.741)
	pr	(0.045, 0.048)	(0.013, 0.014)	(0.014, 0.015)	(0.027, 0.029)	(0.008, 0.009)	(0.008, 0.009)
	randomraw	(0.033, 0.036)	(0.010, 0.011)	(0.010, 0.011)	(0.020, 0.022)	(0.006, 0.006)	(0.006, 0.007)
	random	(0.023, 0.025)	(0.007, 0.007)	(0.007, 0.008)	(0.014, 0.015)	(0.004, 0.004)	(0.004, 0.005)
	EGAD	(0.156, 0.162)	(0.051, 0.053)	(0.054, 0.056)	(0.099, 0.104)	(0.031, 0.032)	(0.033, 0.034)
	ppr	(0.254, 0.261)	(0.089, 0.093)	(0.094, 0.098)	(0.168, 0.174)	(0.055, 0.057)	(0.058, 0.061)
	raw	(0.334, 0.343)	(0.126, 0.131)	(0.133, 0.138)	(0.230, 0.237)	(0.079, 0.082)	(0.084, 0.087)
	gm	(0.204, 0.211)	(0.069, 0.072)	(0.073, 0.076)	(0.132, 0.137)	(0.042, 0.044)	(0.045, 0.047)
auprc	mc	(0.174, 0.181)	(0.057, 0.060)	(0.061, 0.063)	(0.111, 0.116)	(0.035, 0.037)	(0.037, 0.039)
	z	(0.320, 0.328)	(0.119, 0.124)	(0.126, 0.130)	(0.219, 0.225)	(0.075, 0.078)	(0.079, 0.082)
	knn	(0.272, 0.280)	(0.097, 0.101)	(0.103, 0.107)	(0.182, 0.189)	(0.060, 0.063)	(0.064, 0.067)
	wsld	(0.266, 0.274)	(0.095, 0.098)	(0.100, 0.104)	(0.178, 0.184)	(0.059, 0.061)	(0.062, 0.065)
	COSNet	(0.193, 0.200)	(0.064, 0.067)	(0.068, 0.071)	(0.125, 0.130)	(0.039, 0.041)	(0.042, 0.044)
	bagsvm	(0.331, 0.339)	(0.125, 0.129)	(0.131, 0.136)	(0.228, 0.234)	(0.078, 0.081)	(0.083, 0.086)
	rf	(0.384, 0.393)	(0.152, 0.158)	(0.160, 0.166)	(0.271, 0.278)	(0.097, 0.100)	(0.102, 0.106)
	svm	(0.308, 0.316)	(0.114, 0.118)	(0.120, 0.124)	(0.210, 0.216)	(0.071, 0.074)	(0.075, 0.078)
	pr	(0.96, 1.07)	(0.36, 0.40)	(0.35, 0.39)	(0.67, 0.75)	(0.25, 0.28)	(0.25, 0.27)
	randomraw	(0.27, 0.33)	(0.10, 0.12)	(0.10, 0.12)	(0.19, 0.23)	(0.07, 0.09)	(0.07, 0.08)
	random	(0.34, 0.41)	(0.13, 0.15)	(0.13, 0.15)	(0.24, 0.29)	(0.09, 0.11)	(0.09, 0.11)
	EGAD	(5.21, 5.46)	(1.94, 2.05)	(1.91, 2.01)	(3.64, 3.83)	(1.36, 1.43)	(1.33, 1.41)
	ppr	(7.49, 7.80)	(2.80, 2.92)	(2.74, 2.87)	(5.24, 5.47)	(1.96, 2.05)	(1.92, 2.01)
	raw	(9.91, 10.28)	(3.70, 3.85)	(3.63, 3.78)	(6.93, 7.20)	(2.59, 2.70)	(2.54, 2.64)
	gm	(8.32, 8.65)	(3.10, 3.24)	(3.04, 3.18)	(5.82, 6.06)	(2.17, 2.27)	(2.13, 2.23)
top_20_hits	mc	(3.90, 4.12)	(1.46, 1.54)	(1.43, 1.51)	(2.73, 2.89)	(1.02, 1.08)	(1.00, 1.06)
	z	(9.45, 9.81)	(3.53, 3.68)	(3.46, 3.60)	(6.61, 6.87)	(2.47, 2.57)	(2.42, 2.52)
	knn	(8.62, 8.96)	(3.22, 3.36)	(3.16, 3.29)	(6.03, 6.28)	(2.25, 2.35)	(2.21, 2.31)
	wsld	(8.51, 8.84)	(3.17, 3.31)	(3.11, 3.25)	(5.95, 6.19)	(2.22, 2.32)	(2.18, 2.28)
	COSNet	(7.05, 7.36)	(2.63, 2.76)	(2.58, 2.70)	(4.93, 5.15)	(1.84, 1.93)	(1.81, 1.89)
	bagsvm	(9.90, 10.27)	(3.69, 3.85)	(3.62, 3.77)	(6.93, 7.19)	(2.58, 2.69)	(2.53, 2.64)
	rf	(11.58, 11.98)	(4.32, 4.49)	(4.24, 4.40)	(8.10, 8.39)	(3.02, 3.14)	(2.96, 3.08)
	svm	(9.39, 9.74)	(3.50, 3.65)	(3.44, 3.58)	(6.57, 6.83)	(2.45, 2.56)	(2.40, 2.51)

The **DA3** model was used in the main body to statistically compare method performances. We explored its diagnostic plots (figure 111) to ensure we drew sound conclusions from it. The first panel in figure 111 contains the deviance residuals against the predicted values. The lack of tendencies in it, reflected by the flat red line, supports that the residuals are healthy and that the poisson is a suitable distribution to describe the data (Zuur, Alain F and Ieno, Elena N and Walker, Neil J and Saveliev, Anatoly A and Smith, Graham M, 2009). The fourth panel from figure 111 shows that there are no influential observations using Cook's distance statistic.

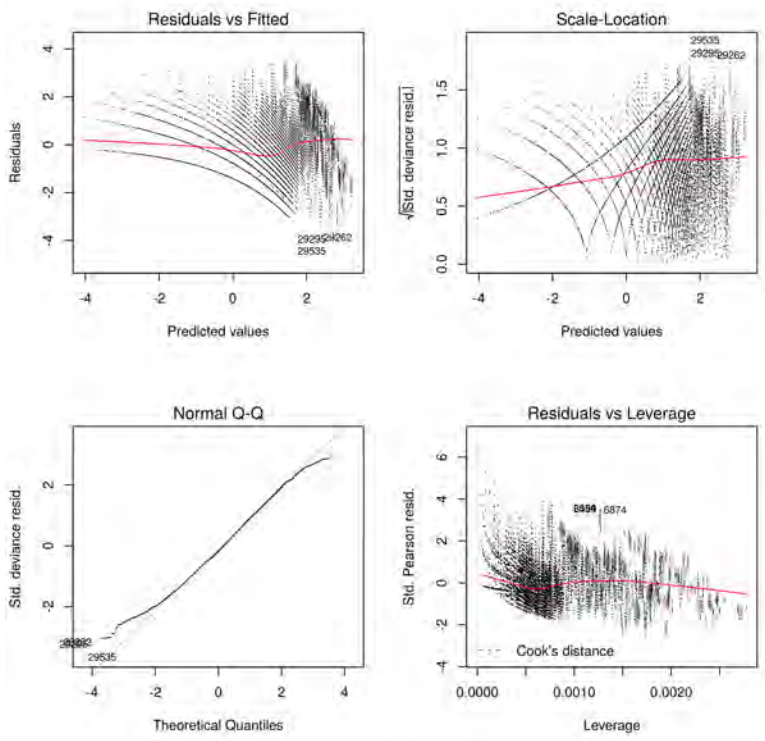


Figure 111: Diagnostics plots for the top 20 hits quasipoisson model **DA3**.

Table 37: Models for the metrics partial_auroc_0.10, partial_auroc_0.05, top_100_hits using the drugs input (model names DA4, DA5 and DA6)

	partial_auroc_0.10	partial_auroc_0.05	top_100_hits
Constant	−1.544*** (−1.575, −1.514)	−2.334*** (−2.372, −2.297)	0.533*** (0.496, 0.570)
methodrandomraw	−0.574*** (−0.610, −0.537)	−0.778*** (−0.828, −0.727)	−0.550*** (−0.593, −0.506)
methodrandom	−0.869*** (−0.908, −0.829)	−0.992*** (−1.046, −0.938)	−0.782*** (−0.829, −0.735)
methodEGAD	1.265*** (1.238, 1.293)	1.287*** (1.253, 1.322)	1.236*** (1.206, 1.266)
methodppr	1.838*** (1.811, 1.865)	1.907*** (1.874, 1.941)	1.606*** (1.577, 1.635)
methodraw	2.061*** (2.034, 2.088)	2.116*** (2.083, 2.149)	1.736*** (1.708, 1.765)
methodgm	0.549*** (0.519, 0.578)	1.005*** (0.970, 1.040)	1.079*** (1.048, 1.110)
methodmc	1.791*** (1.764, 1.819)	1.712*** (1.679, 1.745)	1.472*** (1.442, 1.501)
methodz	1.956*** (1.929, 1.983)	2.034*** (2.001, 2.067)	1.697*** (1.668, 1.726)
methodknn	1.862*** (1.835, 1.889)	1.955*** (1.922, 1.988)	1.683*** (1.654, 1.711)
methodwsld	1.851*** (1.824, 1.878)	1.936*** (1.903, 1.969)	1.674*** (1.645, 1.703)
methodCOSNet	0.792*** (0.764, 0.821)	1.140*** (1.105, 1.174)	1.121*** (1.091, 1.152)
methodbagsvm	2.011*** (1.985, 2.038)	2.089*** (2.056, 2.122)	1.724*** (1.695, 1.753)
methodrf	2.000*** (1.973, 2.027)	2.245*** (2.212, 2.278)	1.837*** (1.809, 1.866)
methodsvm	1.752*** (1.725, 1.780)	2.001*** (1.968, 2.034)	1.711*** (1.682, 1.739)
cv_schemeblock	−0.830*** (−0.841, −0.820)	−0.958*** (−0.969, −0.946)	−0.672*** (−0.680, −0.663)
cv_schemerepresentative	−0.530*** (−0.541, −0.520)	−0.636*** (−0.647, −0.625)	−0.833*** (−0.842, −0.824)
networkomnipath	−0.474*** (−0.483, −0.466)	−0.432*** (−0.441, −0.423)	−0.309*** (−0.316, −0.302)
diseaseAlzheimers disease	0.083*** (0.055, 0.110)	0.304*** (0.272, 0.335)	0.693*** (0.662, 0.723)
diseasearthritis	−0.430*** (−0.459, −0.401)	−0.297*** (−0.330, −0.263)	0.832*** (0.802, 0.862)
diseaseasthma	0.120*** (0.092, 0.148)	0.216*** (0.185, 0.248)	0.509*** (0.477, 0.540)
diseasebipolar disorder	0.156*** (0.128, 0.183)	0.511*** (0.480, 0.542)	1.024*** (0.995, 1.053)
diseasecardiac arrhythmia	0.548*** (0.521, 0.575)	0.952*** (0.922, 0.982)	1.429*** (1.401, 1.457)
diseaseCOPD	−0.080*** (−0.108, −0.051)	0.143*** (0.112, 0.175)	0.644*** (0.613, 0.675)
diseasecoronary heart disease	−0.169*** (−0.198, −0.141)	0.076*** (0.044, 0.108)	0.923*** (0.893, 0.953)
diseasedrug dependence	0.264*** (0.236, 0.292)	0.492*** (0.461, 0.523)	1.075*** (1.046, 1.104)
diseasehypertension	−0.165*** (−0.194, −0.137)	0.101*** (0.069, 0.133)	1.074*** (1.045, 1.103)
diseasemultiple sclerosis	0.211*** (0.183, 0.239)	0.580*** (0.550, 0.611)	1.288*** (1.260, 1.316)
diseaseobesity	0.042*** (0.015, 0.070)	0.301*** (0.270, 0.333)	1.186*** (1.157, 1.214)
diseaseParkinson's disease	0.581*** (0.553, 0.608)	0.842*** (0.812, 0.872)	1.322*** (1.294, 1.350)
diseasepsoriasis	−0.102*** (−0.130, −0.073)	−0.041** (−0.073, −0.008)	0.500*** (0.469, 0.532)
diseaserheumatoid arthritis	−0.353*** (−0.382, −0.324)	−0.236*** (−0.269, −0.203)	0.276*** (0.242, 0.309)
diseaseschizophrenia	0.238*** (0.211, 0.266)	0.506*** (0.476, 0.537)	1.141*** (1.112, 1.170)
diseasestroke	0.463*** (0.436, 0.491)	0.765*** (0.735, 0.795)	1.281*** (1.253, 1.309)
diseaselupus	−0.084*** (−0.112, −0.056)	0.205*** (0.173, 0.236)	0.601*** (0.570, 0.632)
diseasetype I diabetes	−0.103*** (−0.131, −0.075)	0.145*** (0.113, 0.177)	0.536*** (0.504, 0.567)
diseasetype II diabetes	−0.125*** (−0.153, −0.097)	0.138*** (0.106, 0.170)	0.827*** (0.797, 0.857)
diseaseulcerative colitis	0.088*** (0.060, 0.116)	0.173*** (0.141, 0.205)	0.030* (−0.006, 0.065)
diseaseunipolar depression	0.145*** (0.117, 0.173)	0.323*** (0.291, 0.354)	0.890*** (0.860, 0.920)
Observations	49,500	49,500	49,500

Note: *p<0.1; **p<0.05; ***p<0.01

Table 38: Predictions of the models DA4, DA5, DA6 (95% confidence intervals after averaging over disease).

Input: drugs data		STRING			OmniPath		
metric	method	classic	block	representative	classic	block	representative
	pr	(0.181, 0.188)	(0.088, 0.092)	(0.115, 0.120)	(0.121, 0.126)	(0.057, 0.059)	(0.075, 0.078)
	randomraw	(0.110, 0.116)	(0.051, 0.054)	(0.068, 0.072)	(0.072, 0.076)	(0.033, 0.034)	(0.043, 0.046)
	random	(0.084, 0.090)	(0.039, 0.041)	(0.051, 0.055)	(0.054, 0.058)	(0.024, 0.026)	(0.033, 0.035)
	EGAD	(0.441, 0.450)	(0.256, 0.263)	(0.317, 0.325)	(0.330, 0.337)	(0.176, 0.182)	(0.224, 0.230)
	ppr	(0.584, 0.592)	(0.379, 0.387)	(0.452, 0.460)	(0.466, 0.474)	(0.276, 0.282)	(0.339, 0.347)
	raw	(0.637, 0.644)	(0.433, 0.441)	(0.508, 0.516)	(0.522, 0.530)	(0.322, 0.329)	(0.391, 0.399)
	gm	(0.278, 0.286)	(0.144, 0.149)	(0.185, 0.191)	(0.193, 0.199)	(0.094, 0.098)	(0.123, 0.128)
partial_auroc_0.10	mc	(0.572, 0.580)	(0.368, 0.376)	(0.440, 0.449)	(0.454, 0.462)	(0.266, 0.273)	(0.329, 0.336)
	z	(0.612, 0.620)	(0.407, 0.415)	(0.481, 0.490)	(0.495, 0.504)	(0.300, 0.307)	(0.366, 0.374)
	knn	(0.590, 0.597)	(0.385, 0.393)	(0.458, 0.466)	(0.472, 0.480)	(0.280, 0.287)	(0.345, 0.352)
	wsld	(0.587, 0.595)	(0.382, 0.390)	(0.455, 0.463)	(0.469, 0.477)	(0.278, 0.285)	(0.342, 0.349)
	COSNet	(0.329, 0.338)	(0.176, 0.182)	(0.224, 0.231)	(0.234, 0.241)	(0.118, 0.122)	(0.152, 0.157)
	bagsvm	(0.625, 0.633)	(0.421, 0.429)	(0.495, 0.503)	(0.509, 0.517)	(0.311, 0.319)	(0.379, 0.387)
	rf	(0.622, 0.630)	(0.418, 0.426)	(0.492, 0.500)	(0.506, 0.514)	(0.309, 0.316)	(0.376, 0.384)
	svm	(0.563, 0.571)	(0.359, 0.367)	(0.431, 0.439)	(0.445, 0.453)	(0.259, 0.265)	(0.320, 0.328)
	pr	(0.111, 0.117)	(0.046, 0.048)	(0.062, 0.065)	(0.075, 0.079)	(0.030, 0.032)	(0.041, 0.043)
	randomraw	(0.054, 0.058)	(0.021, 0.023)	(0.029, 0.032)	(0.035, 0.038)	(0.014, 0.015)	(0.019, 0.021)
	random	(0.044, 0.047)	(0.017, 0.019)	(0.024, 0.026)	(0.029, 0.031)	(0.011, 0.012)	(0.015, 0.017)
	EGAD	(0.313, 0.322)	(0.149, 0.154)	(0.194, 0.201)	(0.228, 0.235)	(0.102, 0.106)	(0.135, 0.140)
	ppr	(0.459, 0.468)	(0.246, 0.253)	(0.310, 0.318)	(0.356, 0.364)	(0.175, 0.180)	(0.226, 0.232)
	raw	(0.512, 0.520)	(0.287, 0.294)	(0.357, 0.365)	(0.405, 0.413)	(0.207, 0.213)	(0.265, 0.271)
	gm	(0.256, 0.264)	(0.116, 0.121)	(0.154, 0.159)	(0.182, 0.189)	(0.079, 0.082)	(0.105, 0.110)
partial_auroc_0.05	mc	(0.411, 0.420)	(0.211, 0.218)	(0.270, 0.277)	(0.312, 0.320)	(0.148, 0.153)	(0.194, 0.199)
	z	(0.491, 0.500)	(0.270, 0.277)	(0.338, 0.346)	(0.385, 0.393)	(0.194, 0.199)	(0.249, 0.255)
	knn	(0.471, 0.480)	(0.255, 0.262)	(0.320, 0.328)	(0.366, 0.375)	(0.182, 0.187)	(0.234, 0.241)
	wsld	(0.466, 0.475)	(0.251, 0.258)	(0.316, 0.324)	(0.362, 0.370)	(0.179, 0.184)	(0.231, 0.237)
	COSNet	(0.282, 0.291)	(0.131, 0.136)	(0.172, 0.178)	(0.203, 0.210)	(0.089, 0.093)	(0.119, 0.123)
	bagsvm	(0.505, 0.513)	(0.281, 0.288)	(0.350, 0.358)	(0.398, 0.406)	(0.202, 0.208)	(0.259, 0.266)
	rf	(0.544, 0.552)	(0.314, 0.321)	(0.387, 0.395)	(0.436, 0.445)	(0.229, 0.235)	(0.290, 0.298)
	svm	(0.483, 0.492)	(0.264, 0.271)	(0.331, 0.338)	(0.377, 0.386)	(0.189, 0.194)	(0.243, 0.249)
	pr	(3.77, 3.98)	(1.93, 2.04)	(1.64, 1.73)	(2.77, 2.92)	(1.42, 1.49)	(1.20, 1.27)
	randomraw	(2.16, 2.32)	(1.10, 1.18)	(0.94, 1.01)	(1.59, 1.70)	(0.81, 0.87)	(0.69, 0.74)
	random	(1.70, 1.84)	(0.87, 0.94)	(0.74, 0.80)	(1.25, 1.35)	(0.64, 0.69)	(0.54, 0.59)
	EGAD	(13.14, 13.54)	(6.71, 6.93)	(5.71, 5.89)	(9.65, 9.95)	(4.93, 5.09)	(4.19, 4.33)
	ppr	(19.06, 19.55)	(9.73, 10.00)	(8.28, 8.51)	(14.00, 14.36)	(7.14, 7.35)	(6.08, 6.25)
	raw	(21.74, 22.27)	(11.10, 11.39)	(9.44, 9.69)	(15.96, 16.36)	(8.15, 8.37)	(6.93, 7.12)
	gm	(11.22, 11.59)	(5.73, 5.92)	(4.87, 5.04)	(8.24, 8.51)	(4.21, 4.35)	(3.58, 3.70)
top_100_hits	mc	(16.66, 17.11)	(8.50, 8.75)	(7.23, 7.45)	(12.23, 12.57)	(6.24, 6.43)	(5.31, 5.47)
	z	(20.89, 21.41)	(10.67, 10.95)	(9.07, 9.32)	(15.34, 15.73)	(7.83, 8.04)	(6.66, 6.85)
	knn	(20.59, 21.11)	(10.51, 10.79)	(8.94, 9.19)	(15.12, 15.51)	(7.72, 7.93)	(6.57, 6.75)
	wsld	(20.42, 20.94)	(10.43, 10.71)	(8.87, 9.11)	(15.00, 15.38)	(7.66, 7.86)	(6.51, 6.69)
	COSNet	(11.71, 12.08)	(5.98, 6.18)	(5.09, 5.26)	(8.60, 8.88)	(4.39, 4.54)	(3.73, 3.86)
	bagsvm	(21.47, 22.00)	(10.96, 11.25)	(9.32, 9.58)	(15.77, 16.16)	(8.05, 8.27)	(6.85, 7.03)
	rf	(24.06, 24.62)	(12.28, 12.59)	(10.45, 10.72)	(17.66, 18.09)	(9.02, 9.25)	(7.67, 7.87)
	svm	(21.18, 21.71)	(10.81, 11.10)	(9.20, 9.45)	(15.55, 15.95)	(7.94, 8.15)	(6.75, 6.94)

Reduced models

Table 39: Models for the metrics auroc, auprc, top_20_hits using the drugs input, the STRING network and the block cross-validation strategy (model names rDA1, rDA2 and rDA3)

	auroc	auprc	top_20_hits
Constant	1.166*** (1.120, 1.212)	−5.030*** (−5.132, −4.928)	−2.853*** (−3.043, −2.664)
methodrandomraw	−0.451*** (−0.489, −0.414)	−0.282*** (−0.385, −0.180)	−1.452*** (−1.732, −1.172)
methodrandom	−0.986*** (−1.023, −0.949)	−0.720*** (−0.837, −0.604)	−0.676*** (−0.887, −0.466)
methodEGAD	0.191*** (0.151, 0.231)	1.112*** (1.032, 1.191)	1.697*** (1.564, 1.830)
methodppr	0.850*** (0.805, 0.895)	1.664*** (1.588, 1.740)	2.064*** (1.935, 2.194)
methodraw	0.821*** (0.776, 0.866)	1.856*** (1.781, 1.931)	2.249*** (2.121, 2.377)
methodgm	−0.938*** (−0.975, −0.901)	0.877*** (0.795, 0.958)	1.878*** (1.747, 2.009)
methodmc	0.674*** (0.631, 0.718)	1.046*** (0.966, 1.126)	1.083*** (0.942, 1.224)
methodz	0.682*** (0.638, 0.726)	1.712*** (1.636, 1.787)	2.136*** (2.007, 2.265)
methodknn	0.869*** (0.824, 0.915)	1.935*** (1.861, 2.010)	2.421*** (2.294, 2.549)
methodwsld	0.879*** (0.833, 0.924)	1.954*** (1.880, 2.029)	2.431*** (2.304, 2.558)
methodCOSNet	−1.351*** (−1.389, −1.314)	1.050*** (0.969, 1.130)	2.070*** (1.941, 2.200)
methodbagsvm	0.739*** (0.695, 0.783)	1.862*** (1.787, 1.937)	2.258*** (2.130, 2.387)
methodrf	0.613*** (0.570, 0.656)	2.229*** (2.156, 2.303)	2.740*** (2.614, 2.866)
methodsvm	0.251*** (0.210, 0.291)	1.941*** (1.867, 2.016)	2.538*** (2.412, 2.665)
diseaseAlzheimers disease	−0.080*** (−0.133, −0.027)	1.252*** (1.163, 1.341)	1.961*** (1.805, 2.117)
diseasearthritis	−0.350*** (−0.402, −0.299)	0.885*** (0.792, 0.977)	1.561*** (1.401, 1.722)
diseaseasthma	−0.037 (−0.090, 0.016)	0.638*** (0.542, 0.734)	0.885*** (0.712, 1.059)
diseasebipolar disorder	−0.253*** (−0.305, −0.201)	1.308*** (1.220, 1.397)	1.797*** (1.639, 1.954)
diseasecardiac arrhythmia	−0.143*** (−0.195, −0.090)	2.135*** (2.051, 2.218)	2.889*** (2.739, 3.039)
diseaseCOPD	−0.327*** (−0.379, −0.275)	0.656*** (0.560, 0.752)	0.866*** (0.692, 1.040)
diseasecoronary heart disease	−0.526*** (−0.577, −0.475)	0.905*** (0.813, 0.998)	1.501*** (1.339, 1.662)
diseasedrug dependence	−0.088*** (−0.141, −0.035)	1.391*** (1.304, 1.479)	2.056*** (1.901, 2.211)
diseasehypertension	−0.461*** (−0.513, −0.410)	0.868*** (0.775, 0.961)	1.526*** (1.365, 1.687)
diseasemultiple sclerosis	−0.159*** (−0.211, −0.106)	2.190*** (2.107, 2.274)	3.041*** (2.892, 3.190)
diseaseobesity	−0.411*** (−0.463, −0.360)	1.088*** (0.998, 1.179)	1.671*** (1.512, 1.830)
diseaseParkinson's disease	0.267*** (0.212, 0.322)	2.594*** (2.511, 2.676)	3.130*** (2.981, 3.279)
diseasepsoriasis	−0.042 (−0.095, 0.011)	1.205*** (1.115, 1.294)	1.841*** (1.684, 1.998)
diseaserheumatoid arthritis	−0.202*** (−0.254, −0.150)	0.630*** (0.534, 0.726)	1.120*** (0.952, 1.288)
diseaseschizophrenia	−0.245*** (−0.297, −0.192)	1.057*** (0.966, 1.148)	1.379*** (1.216, 1.542)
diseasestroke	0.136*** (0.082, 0.190)	2.395*** (2.312, 2.478)	3.064*** (2.915, 3.213)
diseaselupus	−0.488*** (−0.539, −0.437)	0.648*** (0.552, 0.744)	0.464*** (0.278, 0.650)
diseasetype I diabetes	−0.369*** (−0.420, −0.317)	0.602*** (0.505, 0.698)	0.641*** (0.461, 0.821)
diseasetype II diabetes	−0.353*** (−0.404, −0.301)	0.947*** (0.855, 1.039)	1.395*** (1.232, 1.558)
diseaseulcerative colitis	0.192*** (0.137, 0.247)	0.485*** (0.387, 0.584)	0.809*** (0.634, 0.985)
diseaseunipolar depression	−0.030 (−0.083, 0.023)	1.148*** (1.058, 1.238)	1.772*** (1.614, 1.929)
Observations	8,250	8,250	8,250
Note:			*p<0.1; **p<0.05; ***p<0.01

Table 40: Models for the metrics partial_auroc_0.10, partial_auroc_0.05, top_100_hits using the drugs input, the STRING network and the block cross-validation strategy (model names rDA4, rDA5 and rDA6)

	partial_auroc_0.10	partial_auroc_0.05	top_100_hits
Constant	−2.106*** (−2.172, −2.039)	−3.168*** (−3.253, −3.084)	−0.339*** (−0.438, −0.241)
methodrandomraw	−0.663*** (−0.748, −0.579)	−0.791*** (−0.908, −0.673)	−0.640*** (−0.759, −0.521)
methodrandom	−0.812*** (−0.900, −0.724)	−0.832*** (−0.951, −0.713)	−0.623*** (−0.741, −0.505)
methodEGAD	1.369*** (1.307, 1.431)	1.291*** (1.212, 1.370)	1.243*** (1.163, 1.322)
methodppr	1.837*** (1.777, 1.898)	1.866*** (1.790, 1.942)	1.626*** (1.550, 1.703)
methodraw	1.868*** (1.807, 1.928)	1.919*** (1.843, 1.995)	1.730*** (1.654, 1.806)
methodgm	−0.332*** (−0.410, −0.254)	0.289*** (0.199, 0.379)	0.556*** (0.469, 0.644)
methodmc	1.590*** (1.529, 1.651)	1.527*** (1.450, 1.605)	1.387*** (1.309, 1.465)
methodz	1.769*** (1.708, 1.829)	1.844*** (1.768, 1.920)	1.653*** (1.577, 1.730)
methodknn	1.923*** (1.862, 1.984)	2.062*** (1.987, 2.138)	1.913*** (1.838, 1.988)
methodwsld	1.936*** (1.875, 1.996)	2.082*** (2.007, 2.158)	1.962*** (1.888, 2.037)
methodCOSNet	0.192*** (0.121, 0.262)	0.700*** (0.616, 0.784)	0.836*** (0.752, 0.920)
methodbagsvm	1.852*** (1.792, 1.913)	1.914*** (1.839, 1.990)	1.724*** (1.649, 1.800)
methodrf	1.991*** (1.931, 2.052)	2.324*** (2.249, 2.399)	2.016*** (1.942, 2.090)
methodsvm	1.705*** (1.645, 1.766)	2.060*** (1.985, 2.136)	1.874*** (1.799, 1.949)
diseaseAlzheimers disease	0.066** (0.005, 0.128)	0.482*** (0.412, 0.551)	0.999*** (0.917, 1.081)
diseasearthritis	−0.627*** (−0.693, −0.562)	−0.353*** (−0.431, −0.275)	0.949*** (0.866, 1.031)
diseaseasthma	0.158*** (0.096, 0.219)	0.312*** (0.242, 0.383)	0.598*** (0.510, 0.685)
diseasebipolar disorder	−0.040 (−0.101, 0.022)	0.365*** (0.295, 0.436)	1.193*** (1.113, 1.273)
diseasecardiac arrhythmia	0.245*** (0.184, 0.306)	0.869*** (0.802, 0.937)	1.705*** (1.628, 1.781)
diseaseCOPD	−0.388*** (−0.452, −0.325)	−0.189*** (−0.264, −0.113)	0.552*** (0.464, 0.640)
diseasecoronary heart disease	−0.532*** (−0.597, −0.467)	−0.162*** (−0.238, −0.087)	0.900*** (0.817, 0.983)
diseasedrug dependence	0.205*** (0.144, 0.266)	0.553*** (0.484, 0.622)	1.347*** (1.268, 1.426)
diseasehypertension	−0.655*** (−0.721, −0.589)	−0.273*** (−0.350, −0.196)	0.964*** (0.882, 1.047)
diseasemultiple sclerosis	0.099*** (0.038, 0.160)	0.689*** (0.621, 0.757)	1.561*** (1.483, 1.638)
diseaseobesity	−0.385*** (−0.449, −0.321)	−0.062 (−0.136, 0.012)	1.125*** (1.045, 1.206)
diseaseParkinson's disease	0.832*** (0.772, 0.893)	1.302*** (1.236, 1.368)	1.785*** (1.709, 1.861)
diseasepsoriasis	−0.002 (−0.064, 0.060)	0.281*** (0.210, 0.352)	0.841*** (0.757, 0.925)
diseaserheumatoid arthritis	−0.255*** (−0.318, −0.192)	0.097*** (0.025, 0.170)	0.652*** (0.566, 0.739)
diseaseschizophrenia	−0.205*** (−0.267, −0.142)	0.102*** (0.030, 0.175)	1.146*** (1.065, 1.226)
diseasestroke	0.519*** (0.458, 0.579)	0.966*** (0.899, 1.033)	1.661*** (1.584, 1.737)
diseaselupus	−0.619*** (−0.684, −0.553)	−0.348*** (−0.426, −0.270)	0.378*** (0.287, 0.469)
diseasetype I diabetes	−0.640*** (−0.706, −0.574)	−0.457*** (−0.537, −0.377)	0.405*** (0.315, 0.496)
diseasetype II diabetes	−0.442*** (−0.506, −0.377)	−0.122*** (−0.197, −0.047)	0.854*** (0.770, 0.937)
diseaseulcerative colitis	0.371*** (0.310, 0.432)	0.616*** (0.547, 0.685)	0.432*** (0.342, 0.523)
diseaseunipolar depression	0.079** (0.018, 0.141)	0.385*** (0.315, 0.456)	1.060*** (0.978, 1.141)
Observations	8,250	8,250	8,250
Note:			*p<0.1; **p<0.05; ***p<0.01

Table 41: Predictions of the models rDA1, rDA2, rDA3, rDA4, rDA5, rDA6 (95% confidence intervals after averaging over disease). These models are adjusted on the drugs data, STRING network and block cross-validation, so the only independent variables are the method and the disease.

method	auroc	partial_auroc_0.10	partial_auroc_0.05	auprc	top_20_hits	top_100_hits
pr	(0.723, 0.734)	(0.095, 0.104)	(0.047, 0.054)	(0.019, 0.021)	(0.25, 0.33)	(1.73, 1.99)
randomraw	(0.624, 0.636)	(0.050, 0.057)	(0.021, 0.026)	(0.014, 0.016)	(0.05, 0.09)	(0.89, 1.08)
random	(0.494, 0.506)	(0.044, 0.050)	(0.020, 0.025)	(0.009, 0.011)	(0.12, 0.17)	(0.91, 1.10)
EGAD	(0.759, 0.770)	(0.295, 0.309)	(0.156, 0.167)	(0.056, 0.061)	(1.49, 1.66)	(6.20, 6.69)
ppr	(0.858, 0.867)	(0.401, 0.417)	(0.248, 0.262)	(0.094, 0.100)	(2.17, 2.37)	(9.16, 9.75)
raw	(0.855, 0.863)	(0.408, 0.424)	(0.258, 0.272)	(0.112, 0.119)	(2.62, 2.84)	(10.18, 10.80)
gm	(0.506, 0.518)	(0.069, 0.077)	(0.062, 0.070)	(0.045, 0.049)	(1.79, 1.98)	(3.07, 3.42)
mc	(0.836, 0.845)	(0.343, 0.358)	(0.190, 0.202)	(0.053, 0.057)	(0.79, 0.91)	(7.18, 7.70)
z	(0.837, 0.846)	(0.385, 0.400)	(0.244, 0.258)	(0.098, 0.105)	(2.33, 2.54)	(9.42, 10.02)
knn	(0.860, 0.869)	(0.422, 0.438)	(0.287, 0.301)	(0.120, 0.127)	(3.12, 3.37)	(12.25, 12.94)
wsld	(0.862, 0.870)	(0.425, 0.441)	(0.291, 0.305)	(0.122, 0.129)	(3.15, 3.40)	(12.88, 13.58)
COSNet	(0.404, 0.416)	(0.113, 0.123)	(0.092, 0.101)	(0.053, 0.057)	(2.18, 2.39)	(4.10, 4.49)
bagsvm	(0.844, 0.853)	(0.405, 0.420)	(0.257, 0.271)	(0.113, 0.119)	(2.64, 2.87)	(10.12, 10.74)
rf	(0.827, 0.836)	(0.439, 0.454)	(0.344, 0.359)	(0.156, 0.163)	(4.31, 4.61)	(13.60, 14.32)
svm	(0.770, 0.780)	(0.370, 0.385)	(0.287, 0.301)	(0.121, 0.128)	(3.51, 3.78)	(11.78, 12.45)

E.4.3 Genetic input

Table 42: Models for the metrics auroc, auprc, top_20_hits using the genetic input (model names GA1, GA2 and GA3)

	auroc	auprc	top_20_hits
Constant	0.793*** (0.776, 0.811)	−4.383*** (−4.423, −4.344)	−1.676*** (−1.776, −1.576)
methodrandomraw	−0.337*** (−0.352, −0.322)	−0.305*** (−0.333, −0.277)	−1.069*** (−1.145, −0.993)
methodrandom	−0.759*** (−0.774, −0.744)	−0.676*** (−0.707, −0.645)	−0.981*** (−1.055, −0.908)
methodEGAD	−0.392*** (−0.407, −0.377)	−0.264*** (−0.292, −0.236)	−0.860*** (−0.931, −0.789)
methodppr	−0.001 (−0.016, 0.015)	0.196*** (0.171, 0.221)	0.393*** (0.343, 0.443)
methodraw	−0.129*** (−0.145, −0.114)	0.102*** (0.077, 0.128)	0.135*** (0.082, 0.187)
methodgm	−1.249*** (−1.265, −1.234)	−0.479*** (−0.509, −0.449)	−0.355*** (−0.415, −0.295)
methodmc	−0.310*** (−0.325, −0.295)	0.005 (−0.021, 0.031)	0.403*** (0.353, 0.453)
methodz	−0.330*** (−0.345, −0.314)	0.042*** (0.017, 0.068)	−0.071** (−0.127, −0.016)
methodknn	−0.242*** (−0.257, −0.227)	−0.019 (−0.045, 0.007)	0.103*** (0.050, 0.156)
methodwsld	−0.243*** (−0.258, −0.228)	−0.060*** (−0.087, −0.034)	−0.055* (−0.110, 0.0003)
methodCOSNet	−0.692*** (−0.707, −0.677)	−0.067*** (−0.094, −0.041)	−0.052* (−0.108, 0.003)
methodbagsvm	−0.356*** (−0.371, −0.341)	−0.013 (−0.039, 0.013)	0.106*** (0.053, 0.159)
methodrf	−0.497*** (−0.512, −0.482)	−0.455*** (−0.484, −0.425)	−0.624*** (−0.689, −0.559)
methodsvm	−0.629*** (−0.644, −0.614)	−0.420*** (−0.449, −0.391)	−0.565*** (−0.629, −0.501)
cv_schemeblock	0.045*** (0.038, 0.051)	0.018*** (0.006, 0.030)	−0.172*** (−0.196, −0.147)
cv_schemerepresentative	0.118*** (0.111, 0.125)	−0.328*** (−0.341, −0.315)	−0.473*** (−0.500, −0.446)
diseaseAlzheimers disease	0.082*** (0.064, 0.101)	0.686*** (0.644, 0.729)	1.262*** (1.158, 1.366)
diseasearthritis	0.051*** (0.033, 0.070)	0.938*** (0.898, 0.979)	1.161*** (1.056, 1.266)
diseaseasthma	0.248*** (0.230, 0.267)	0.619*** (0.576, 0.662)	0.516*** (0.401, 0.632)
diseasebipolar disorder	−0.082*** (−0.100, −0.064)	0.587*** (0.544, 0.630)	0.424*** (0.306, 0.542)
diseasecardiac arrhythmia	−0.075*** (−0.093, −0.057)	1.625*** (1.587, 1.662)	2.476*** (2.381, 2.571)
diseaseCOPD	−0.111*** (−0.129, −0.093)	0.584*** (0.541, 0.627)	1.255*** (1.152, 1.359)
diseasecoronary heart disease	−0.243*** (−0.262, −0.225)	0.663*** (0.620, 0.705)	1.261*** (1.158, 1.365)
diseasedrug dependence	−0.036*** (−0.054, −0.018)	0.835*** (0.793, 0.876)	1.597*** (1.497, 1.698)
diseasehypertension	−0.083*** (−0.101, −0.065)	0.995*** (0.955, 1.035)	1.789*** (1.690, 1.888)
diseasemultiple sclerosis	0.041*** (0.023, 0.059)	0.952*** (0.912, 0.993)	1.080*** (0.974, 1.186)
diseaseobesity	−0.070*** (−0.088, −0.052)	0.937*** (0.897, 0.978)	1.147*** (1.042, 1.252)
diseaseParkinson's disease	−0.222*** (−0.240, −0.204)	0.418*** (0.374, 0.462)	0.213*** (0.090, 0.337)
diseasepsoriasis	0.154*** (0.136, 0.173)	0.831*** (0.790, 0.872)	1.494*** (1.393, 1.596)
diseaserheumatoid arthritis	0.183*** (0.165, 0.202)	0.643*** (0.601, 0.686)	1.140*** (1.035, 1.245)
diseaseschizophrenia	0.028*** (0.010, 0.046)	0.748*** (0.706, 0.790)	0.613*** (0.499, 0.727)
diseasestroke	−0.295*** (−0.314, −0.277)	0.427*** (0.383, 0.471)	0.443*** (0.325, 0.560)
diseaselupus	−0.144*** (−0.162, −0.126)	0.416*** (0.372, 0.460)	1.084*** (0.978, 1.190)
diseasetype I diabetes mellitus	−0.023** (−0.041, −0.005)	0.516*** (0.473, 0.559)	1.072*** (0.966, 1.179)
diseasetype II diabetes mellitus	−0.176*** (−0.194, −0.158)	0.494*** (0.450, 0.537)	0.448*** (0.330, 0.565)
diseaseulcerative colitis	0.620*** (0.601, 0.639)	0.829*** (0.788, 0.870)	1.311*** (1.208, 1.414)
diseaseunipolar depression	0.125*** (0.107, 0.143)	0.851*** (0.810, 0.892)	1.382*** (1.280, 1.485)
networkomnipath	−0.167*** (−0.172, −0.161)	−0.052*** (−0.062, −0.041)	0.202*** (0.181, 0.224)
Observations	49,500	49,500	49,500

Note: *p<0.1; **p<0.05; ***p<0.01

Table 43: Predictions of the models GA1, GA2, GA3 (95% confidence intervals after averaging over disease).

Input: genetic data		STRING			OmniPath		
metric	method	classic	block	representative	classic	block	representative
	pr	(0.686, 0.691)	(0.695, 0.700)	(0.710, 0.715)	(0.649, 0.654)	(0.659, 0.664)	(0.675, 0.680)
	randomraw	(0.609, 0.615)	(0.620, 0.625)	(0.637, 0.642)	(0.569, 0.574)	(0.580, 0.585)	(0.597, 0.603)
	random	(0.505, 0.511)	(0.517, 0.522)	(0.535, 0.541)	(0.464, 0.469)	(0.475, 0.481)	(0.493, 0.499)
	EGAD	(0.596, 0.601)	(0.607, 0.612)	(0.624, 0.629)	(0.555, 0.561)	(0.566, 0.572)	(0.584, 0.590)
	ppr	(0.685, 0.691)	(0.695, 0.700)	(0.710, 0.715)	(0.648, 0.654)	(0.659, 0.664)	(0.675, 0.680)
	raw	(0.657, 0.662)	(0.667, 0.672)	(0.683, 0.688)	(0.619, 0.624)	(0.629, 0.635)	(0.646, 0.651)
	gm	(0.385, 0.390)	(0.395, 0.401)	(0.413, 0.419)	(0.346, 0.351)	(0.356, 0.362)	(0.373, 0.379)
auroc	mc	(0.615, 0.621)	(0.626, 0.631)	(0.643, 0.648)	(0.575, 0.581)	(0.586, 0.592)	(0.604, 0.609)
	z	(0.611, 0.616)	(0.621, 0.627)	(0.638, 0.644)	(0.570, 0.576)	(0.581, 0.587)	(0.599, 0.605)
	knn	(0.631, 0.637)	(0.642, 0.647)	(0.658, 0.664)	(0.592, 0.597)	(0.602, 0.608)	(0.620, 0.625)
	wsld	(0.631, 0.637)	(0.641, 0.647)	(0.658, 0.663)	(0.591, 0.597)	(0.602, 0.608)	(0.620, 0.625)
	COSNet	(0.522, 0.528)	(0.533, 0.539)	(0.551, 0.557)	(0.480, 0.486)	(0.492, 0.497)	(0.510, 0.516)
	bagsvm	(0.605, 0.610)	(0.615, 0.621)	(0.632, 0.638)	(0.564, 0.570)	(0.575, 0.581)	(0.593, 0.598)
	rf	(0.570, 0.576)	(0.581, 0.587)	(0.599, 0.605)	(0.529, 0.535)	(0.540, 0.546)	(0.558, 0.564)
	svm	(0.538, 0.544)	(0.549, 0.555)	(0.567, 0.573)	(0.496, 0.502)	(0.507, 0.513)	(0.526, 0.531)
	pr	(0.024, 0.025)	(0.025, 0.026)	(0.018, 0.018)	(0.023, 0.024)	(0.023, 0.024)	(0.017, 0.017)
	randomraw	(0.018, 0.019)	(0.018, 0.019)	(0.013, 0.014)	(0.017, 0.018)	(0.017, 0.018)	(0.012, 0.013)
	random	(0.012, 0.013)	(0.013, 0.013)	(0.009, 0.009)	(0.012, 0.012)	(0.012, 0.013)	(0.009, 0.009)
	EGAD	(0.019, 0.020)	(0.019, 0.020)	(0.014, 0.014)	(0.018, 0.019)	(0.018, 0.019)	(0.013, 0.013)
	ppr	(0.029, 0.030)	(0.030, 0.031)	(0.021, 0.022)	(0.028, 0.029)	(0.028, 0.030)	(0.020, 0.021)
	raw	(0.027, 0.028)	(0.027, 0.028)	(0.019, 0.020)	(0.026, 0.026)	(0.026, 0.027)	(0.018, 0.019)
	gm	(0.015, 0.016)	(0.015, 0.016)	(0.011, 0.011)	(0.014, 0.015)	(0.015, 0.015)	(0.010, 0.011)
auprc	mc	(0.024, 0.025)	(0.025, 0.026)	(0.018, 0.018)	(0.023, 0.024)	(0.024, 0.025)	(0.017, 0.017)
	z	(0.025, 0.026)	(0.026, 0.027)	(0.018, 0.019)	(0.024, 0.025)	(0.024, 0.025)	(0.017, 0.018)
	knn	(0.024, 0.025)	(0.024, 0.025)	(0.017, 0.018)	(0.023, 0.024)	(0.023, 0.024)	(0.016, 0.017)
	wsld	(0.023, 0.024)	(0.023, 0.024)	(0.017, 0.017)	(0.022, 0.023)	(0.022, 0.023)	(0.016, 0.016)
	COSNet	(0.023, 0.024)	(0.023, 0.024)	(0.016, 0.017)	(0.022, 0.022)	(0.022, 0.023)	(0.016, 0.016)
	bagsvm	(0.024, 0.025)	(0.024, 0.025)	(0.017, 0.018)	(0.023, 0.024)	(0.023, 0.024)	(0.016, 0.017)
	rf	(0.015, 0.016)	(0.016, 0.017)	(0.011, 0.012)	(0.015, 0.015)	(0.015, 0.016)	(0.011, 0.011)
	svm	(0.016, 0.017)	(0.016, 0.017)	(0.012, 0.012)	(0.015, 0.016)	(0.015, 0.016)	(0.011, 0.012)
	pr	(0.51, 0.56)	(0.43, 0.47)	(0.32, 0.35)	(0.63, 0.68)	(0.53, 0.58)	(0.39, 0.43)
	randomraw	(0.17, 0.20)	(0.14, 0.17)	(0.11, 0.12)	(0.21, 0.24)	(0.18, 0.20)	(0.13, 0.15)
	random	(0.19, 0.21)	(0.16, 0.18)	(0.12, 0.13)	(0.23, 0.26)	(0.19, 0.22)	(0.14, 0.16)
	EGAD	(0.21, 0.24)	(0.18, 0.20)	(0.13, 0.15)	(0.26, 0.30)	(0.22, 0.25)	(0.16, 0.18)
	ppr	(0.77, 0.82)	(0.64, 0.69)	(0.48, 0.51)	(0.94, 1.01)	(0.79, 0.85)	(0.58, 0.63)
	raw	(0.59, 0.64)	(0.50, 0.54)	(0.37, 0.40)	(0.72, 0.78)	(0.61, 0.66)	(0.45, 0.49)
	gm	(0.36, 0.40)	(0.30, 0.33)	(0.22, 0.25)	(0.44, 0.48)	(0.37, 0.41)	(0.27, 0.30)
top_20_hits	mc	(0.77, 0.83)	(0.65, 0.70)	(0.48, 0.52)	(0.95, 1.02)	(0.80, 0.86)	(0.59, 0.64)
	z	(0.48, 0.52)	(0.40, 0.44)	(0.30, 0.33)	(0.59, 0.64)	(0.49, 0.54)	(0.36, 0.40)
	knn	(0.57, 0.62)	(0.48, 0.52)	(0.35, 0.39)	(0.70, 0.76)	(0.59, 0.64)	(0.43, 0.47)
	wsld	(0.49, 0.53)	(0.41, 0.45)	(0.30, 0.33)	(0.60, 0.65)	(0.50, 0.55)	(0.37, 0.40)
	COSNet	(0.49, 0.53)	(0.41, 0.45)	(0.30, 0.33)	(0.60, 0.65)	(0.50, 0.55)	(0.37, 0.41)
	bagsvm	(0.57, 0.62)	(0.48, 0.52)	(0.36, 0.39)	(0.70, 0.76)	(0.59, 0.64)	(0.44, 0.47)
	rf	(0.27, 0.30)	(0.23, 0.26)	(0.17, 0.19)	(0.33, 0.37)	(0.28, 0.31)	(0.21, 0.23)
	svm	(0.29, 0.32)	(0.24, 0.27)	(0.18, 0.20)	(0.35, 0.39)	(0.30, 0.33)	(0.22, 0.25)

Table 44: Models for the metrics partial_auroc_0.10, partial_auroc_0.05, top_100_hits using the genetic input (model names GA4, GA5 and GA6)

	partial_auroc_0.10	partial_auroc_0.05	top_100_hits
Constant	−2.173*** (−2.207, −2.138)	−2.758*** (−2.801, −2.715)	0.249*** (0.195, 0.302)
methodrandomraw	−0.622*** (−0.654, −0.589)	−0.782*** (−0.824, −0.740)	−0.514*** (−0.555, −0.473)
methodrandom	−0.871*** (−0.905, −0.836)	−0.999*** (−1.044, −0.954)	−0.784*** (−0.828, −0.739)
methodEGAD	−0.040*** (−0.069, −0.012)	−0.349*** (−0.386, −0.312)	−0.267*** (−0.305, −0.229)
methodppr	0.395*** (0.369, 0.421)	0.398*** (0.366, 0.430)	0.377*** (0.345, 0.409)
methodraw	0.170*** (0.143, 0.197)	0.030* (−0.004, 0.065)	0.085*** (0.051, 0.120)
methodgm	−1.096*** (−1.133, −1.059)	−0.931*** (−0.976, −0.887)	−0.553*** (−0.595, −0.512)
methodmc	0.127*** (0.099, 0.154)	0.163*** (0.130, 0.197)	0.196*** (0.162, 0.230)
methodz	0.141*** (0.114, 0.169)	0.112*** (0.078, 0.145)	0.162*** (0.128, 0.196)
methodknn	−0.188*** (−0.217, −0.159)	−0.244*** (−0.280, −0.207)	−0.106*** (−0.142, −0.070)
methodwsld	−0.233*** (−0.263, −0.204)	−0.271*** (−0.308, −0.234)	−0.122*** (−0.158, −0.085)
methodCOSNet	0.009 (−0.019, 0.037)	0.090*** (0.056, 0.123)	0.103*** (0.069, 0.138)
methodbagsvm	0.013 (−0.015, 0.041)	−0.070*** (−0.105, −0.035)	−0.004 (−0.040, 0.031)
methodrf	−0.473*** (−0.504, −0.442)	−0.630*** (−0.670, −0.590)	−0.476*** (−0.516, −0.436)
methodsvm	−0.475*** (−0.507, −0.444)	−0.569*** (−0.608, −0.529)	−0.465*** (−0.505, −0.424)
cv_schemeblock	−0.024*** (−0.038, −0.011)	−0.043*** (−0.060, −0.026)	−0.087*** (−0.103, −0.071)
cv_schemerepresentative	0.005 (−0.008, 0.019)	−0.028*** (−0.044, −0.011)	−0.433*** (−0.451, −0.416)
diseaseAlzheimers disease	0.174*** (0.137, 0.211)	0.255*** (0.209, 0.301)	0.739*** (0.683, 0.796)
diseasearthritis	−0.020 (−0.059, 0.019)	−0.102*** (−0.152, −0.052)	0.930*** (0.875, 0.985)
diseaseasthma	0.339*** (0.303, 0.375)	0.073*** (0.025, 0.121)	0.350*** (0.290, 0.411)
diseasebipolar disorder	−0.075*** (−0.114, −0.036)	−0.307*** (−0.360, −0.254)	0.327*** (0.266, 0.388)
diseasecardiac arrhythmia	0.684*** (0.650, 0.718)	1.035*** (0.994, 1.077)	1.750*** (1.700, 1.801)
diseaseCOPD	−0.073*** (−0.112, −0.034)	−0.140*** (−0.191, −0.090)	0.382*** (0.322, 0.442)
diseasecoronary heart disease	−0.160*** (−0.199, −0.120)	−0.134*** (−0.184, −0.084)	0.701*** (0.644, 0.758)
diseasedrug dependence	0.048** (0.010, 0.086)	0.070*** (0.022, 0.118)	0.759*** (0.702, 0.815)
diseasehypertension	0.148*** (0.110, 0.185)	0.196*** (0.149, 0.243)	1.104*** (1.051, 1.158)
diseasemultiple sclerosis	0.126*** (0.088, 0.164)	−0.039 (−0.089, 0.010)	0.865*** (0.810, 0.921)
diseaseobesity	0.144*** (0.107, 0.182)	0.064*** (0.016, 0.113)	0.988*** (0.934, 1.043)
diseaseParkinson's disease	−0.744*** (−0.791, −0.698)	−0.811*** (−0.872, −0.750)	0.057* (−0.008, 0.121)
diseasepsoriasis	0.407*** (0.371, 0.443)	0.435*** (0.391, 0.480)	0.925*** (0.870, 0.980)
diseaserheumatoid arthritis	0.402*** (0.366, 0.438)	0.385*** (0.340, 0.430)	0.772*** (0.716, 0.828)
diseaseschizophrenia	−0.022 (−0.061, 0.016)	−0.136*** (−0.187, −0.086)	0.700*** (0.643, 0.757)
diseasestroke	−0.509*** (−0.552, −0.465)	−0.535*** (−0.591, −0.479)	0.334*** (0.273, 0.395)
diseaselupus	0.177*** (0.140, 0.214)	0.261*** (0.215, 0.307)	0.558*** (0.499, 0.616)
diseasetype I diabetes mellitus	0.424*** (0.388, 0.459)	0.423*** (0.378, 0.468)	0.658*** (0.601, 0.715)
diseasetype II diabetes mellitus	−0.274*** (−0.315, −0.233)	−0.381*** (−0.434, −0.327)	0.314*** (0.252, 0.375)
diseaseulcerative colitis	1.008*** (0.974, 1.041)	0.977*** (0.936, 1.019)	0.744*** (0.687, 0.800)
diseaseunipolar depression	0.334*** (0.298, 0.370)	0.320*** (0.275, 0.366)	0.943*** (0.888, 0.997)
networkomnipath	−0.145*** (−0.156, −0.134)	−0.105*** (−0.119, −0.091)	0.013* (−0.001, 0.026)
Observations	49,500	49,500	49,500

Note: *p<0.1; **p<0.05; ***p<0.01

Table 45: Predictions of the models GA4, GA5 and GA6 (95% confidence intervals after averaging over disease).

Input: genetic data		STRING			OmniPath		
metric	method	classic	block	representative	classic	block	representative
	pr	(0.111, 0.116)	(0.109, 0.113)	(0.112, 0.116)	(0.098, 0.102)	(0.095, 0.099)	(0.098, 0.102)
	randomraw	(0.063, 0.066)	(0.061, 0.064)	(0.063, 0.066)	(0.055, 0.058)	(0.053, 0.056)	(0.055, 0.058)
	random	(0.049, 0.052)	(0.048, 0.051)	(0.050, 0.053)	(0.043, 0.046)	(0.042, 0.044)	(0.043, 0.046)
	EGAD	(0.107, 0.112)	(0.105, 0.109)	(0.108, 0.112)	(0.094, 0.098)	(0.092, 0.096)	(0.095, 0.098)
	ppr	(0.157, 0.162)	(0.154, 0.159)	(0.158, 0.163)	(0.139, 0.143)	(0.136, 0.140)	(0.139, 0.144)
	raw	(0.129, 0.134)	(0.126, 0.131)	(0.130, 0.135)	(0.114, 0.118)	(0.111, 0.116)	(0.114, 0.119)
	gm	(0.040, 0.042)	(0.039, 0.041)	(0.040, 0.042)	(0.035, 0.037)	(0.034, 0.036)	(0.035, 0.037)
partial_auroc_0.10	mc	(0.124, 0.129)	(0.122, 0.126)	(0.125, 0.130)	(0.109, 0.114)	(0.107, 0.111)	(0.110, 0.114)
	z	(0.126, 0.131)	(0.123, 0.128)	(0.127, 0.131)	(0.111, 0.115)	(0.108, 0.113)	(0.111, 0.116)
	knn	(0.094, 0.098)	(0.092, 0.096)	(0.094, 0.098)	(0.082, 0.086)	(0.080, 0.084)	(0.083, 0.086)
	wsld	(0.090, 0.094)	(0.088, 0.092)	(0.090, 0.094)	(0.079, 0.082)	(0.077, 0.080)	(0.079, 0.083)
	COSNet	(0.112, 0.116)	(0.110, 0.114)	(0.113, 0.117)	(0.098, 0.102)	(0.096, 0.100)	(0.099, 0.103)
	bagsvm	(0.112, 0.117)	(0.110, 0.114)	(0.113, 0.117)	(0.099, 0.103)	(0.097, 0.101)	(0.099, 0.103)
	rf	(0.072, 0.076)	(0.070, 0.074)	(0.072, 0.076)	(0.063, 0.066)	(0.061, 0.065)	(0.063, 0.066)
	svm	(0.072, 0.075)	(0.070, 0.074)	(0.072, 0.076)	(0.063, 0.066)	(0.061, 0.064)	(0.063, 0.066)
	pr	(0.063, 0.066)	(0.061, 0.064)	(0.061, 0.065)	(0.057, 0.060)	(0.055, 0.058)	(0.056, 0.059)
	randomraw	(0.030, 0.032)	(0.028, 0.030)	(0.029, 0.031)	(0.027, 0.029)	(0.026, 0.028)	(0.026, 0.028)
	random	(0.024, 0.026)	(0.023, 0.025)	(0.023, 0.025)	(0.022, 0.023)	(0.021, 0.022)	(0.021, 0.023)
	EGAD	(0.045, 0.048)	(0.043, 0.046)	(0.044, 0.047)	(0.041, 0.043)	(0.039, 0.042)	(0.040, 0.042)
	ppr	(0.091, 0.095)	(0.088, 0.092)	(0.089, 0.093)	(0.083, 0.087)	(0.080, 0.083)	(0.081, 0.085)
	raw	(0.065, 0.068)	(0.062, 0.066)	(0.063, 0.067)	(0.059, 0.062)	(0.056, 0.059)	(0.057, 0.060)
	gm	(0.026, 0.028)	(0.024, 0.026)	(0.025, 0.027)	(0.023, 0.025)	(0.022, 0.024)	(0.022, 0.024)
partial_auroc_0.05	mc	(0.074, 0.077)	(0.071, 0.074)	(0.072, 0.075)	(0.067, 0.070)	(0.064, 0.067)	(0.065, 0.068)
	z	(0.070, 0.074)	(0.067, 0.071)	(0.068, 0.072)	(0.064, 0.067)	(0.061, 0.064)	(0.062, 0.065)
	knn	(0.050, 0.053)	(0.048, 0.051)	(0.049, 0.052)	(0.045, 0.048)	(0.043, 0.046)	(0.044, 0.047)
	wsld	(0.049, 0.052)	(0.047, 0.049)	(0.047, 0.050)	(0.044, 0.047)	(0.042, 0.045)	(0.043, 0.045)
	COSNet	(0.069, 0.072)	(0.066, 0.069)	(0.067, 0.070)	(0.062, 0.065)	(0.060, 0.063)	(0.061, 0.064)
	bagsvm	(0.059, 0.062)	(0.057, 0.060)	(0.057, 0.061)	(0.053, 0.056)	(0.051, 0.054)	(0.052, 0.055)
	rf	(0.034, 0.037)	(0.033, 0.035)	(0.033, 0.036)	(0.031, 0.033)	(0.030, 0.032)	(0.030, 0.032)
	svm	(0.036, 0.039)	(0.035, 0.037)	(0.036, 0.038)	(0.033, 0.035)	(0.032, 0.034)	(0.032, 0.034)
	pr	(2.46, 2.59)	(2.25, 2.38)	(1.59, 1.68)	(2.49, 2.63)	(2.28, 2.41)	(1.61, 1.71)
	randomraw	(1.46, 1.56)	(1.34, 1.43)	(0.95, 1.01)	(1.48, 1.58)	(1.35, 1.45)	(0.96, 1.03)
	random	(1.11, 1.20)	(1.02, 1.10)	(0.72, 0.78)	(1.12, 1.21)	(1.03, 1.11)	(0.73, 0.79)
	EGAD	(1.87, 1.99)	(1.72, 1.83)	(1.21, 1.29)	(1.90, 2.02)	(1.74, 1.85)	(1.23, 1.31)
	ppr	(3.59, 3.77)	(3.29, 3.45)	(2.33, 2.45)	(3.64, 3.81)	(3.34, 3.50)	(2.36, 2.48)
	raw	(2.68, 2.82)	(2.45, 2.59)	(1.73, 1.83)	(2.71, 2.86)	(2.48, 2.62)	(1.76, 1.86)
	gm	(1.40, 1.50)	(1.28, 1.38)	(0.91, 0.98)	(1.42, 1.52)	(1.30, 1.40)	(0.92, 0.99)
top_100_hits	mc	(2.99, 3.15)	(2.74, 2.89)	(1.94, 2.04)	(3.03, 3.19)	(2.78, 2.92)	(1.96, 2.07)
	z	(2.89, 3.04)	(2.65, 2.79)	(1.87, 1.98)	(2.93, 3.08)	(2.68, 2.83)	(1.90, 2.00)
	knn	(2.21, 2.34)	(2.02, 2.14)	(1.43, 1.52)	(2.23, 2.36)	(2.05, 2.17)	(1.45, 1.54)
	wsld	(2.17, 2.30)	(1.99, 2.11)	(1.41, 1.49)	(2.20, 2.33)	(2.02, 2.14)	(1.42, 1.51)
	COSNet	(2.73, 2.87)	(2.50, 2.63)	(1.77, 1.87)	(2.76, 2.91)	(2.53, 2.67)	(1.79, 1.89)
	bagsvm	(2.45, 2.58)	(2.24, 2.37)	(1.58, 1.68)	(2.48, 2.62)	(2.27, 2.40)	(1.60, 1.70)
	rf	(1.52, 1.62)	(1.39, 1.49)	(0.98, 1.05)	(1.54, 1.64)	(1.41, 1.51)	(0.99, 1.07)
	svm	(1.53, 1.64)	(1.41, 1.50)	(0.99, 1.06)	(1.55, 1.66)	(1.42, 1.52)	(1.01, 1.08)

E.4.4 Reference streams

In order to evaluate the extent to which using networks for predicting disease genes is of use compared against not using networks at all, we have also checked the extent to which the gene scores from other data streams in Open Targets could be used to recover known drug targets. To that end, we have computed the metrics between all the remaining streams and the drug targets stream, reusing the partitions from the cross validation folds (see main text).

These metrics are therefore not directly comparable to those presented above for the network-based approaches, as in this case, the concept of cross-validation does not apply. It is rather a data subsetting strategy to compute the estimates, to which the additive models also apply (see table 46).

The genes scores from the Open Targets *literature* data stream result in the best alignment with the scores from known drug targets.

There may be some circularity explaining this as the *literature* data stream uses publications mentioning known drug targets and their relation to diseases, and also as a gene with a lot of *literature* describing its relationship to disease may be more likely to be picked as a potential drug target. The *genetic association* data stream is second best in terms of correlation with the known drug target scores, thereby justifying its usage for finding potential targets a posteriori.

Table 46: Predictions of the models SA1, SA2, SA3, SA4, SA5 and SA6 (95% confidence intervals after averaging over disease).

Input: streams		STRING			OmniPath		
metric	method (stream)	classic	block	representative	classic	block	representative
auroc	affected_pathway	(0.494, 0.496)	(0.496, 0.498)	(0.505, 0.508)	(0.495, 0.498)	(0.497, 0.500)	(0.506, 0.509)
	animal_model	(0.503, 0.506)	(0.505, 0.508)	(0.514, 0.517)	(0.505, 0.507)	(0.507, 0.509)	(0.516, 0.518)
	genetic_association	(0.516, 0.519)	(0.518, 0.521)	(0.527, 0.530)	(0.517, 0.520)	(0.520, 0.522)	(0.529, 0.531)
	literature	(0.692, 0.694)	(0.693, 0.696)	(0.701, 0.703)	(0.693, 0.695)	(0.695, 0.697)	(0.702, 0.705)
	rna_expression	(0.511, 0.513)	(0.513, 0.515)	(0.522, 0.524)	(0.512, 0.515)	(0.514, 0.517)	(0.523, 0.526)
	somatic_mutation	(0.493, 0.496)	(0.496, 0.498)	(0.505, 0.507)	(0.495, 0.498)	(0.497, 0.500)	(0.506, 0.509)
partial_auroc_0.10	affected_pathway	(0.046, 0.047)	(0.047, 0.049)	(0.054, 0.055)	(0.045, 0.047)	(0.047, 0.048)	(0.053, 0.055)
	animal_model	(0.062, 0.063)	(0.064, 0.066)	(0.072, 0.074)	(0.061, 0.063)	(0.063, 0.065)	(0.072, 0.074)
	genetic_association	(0.081, 0.083)	(0.084, 0.086)	(0.095, 0.097)	(0.080, 0.082)	(0.083, 0.085)	(0.094, 0.096)
	literature	(0.275, 0.279)	(0.282, 0.286)	(0.310, 0.315)	(0.273, 0.277)	(0.280, 0.284)	(0.308, 0.313)
	rna_expression	(0.066, 0.067)	(0.068, 0.070)	(0.077, 0.079)	(0.065, 0.067)	(0.067, 0.069)	(0.076, 0.078)
	somatic_mutation	(0.045, 0.047)	(0.047, 0.048)	(0.053, 0.055)	(0.045, 0.046)	(0.046, 0.048)	(0.053, 0.054)
partial_auroc_0.05	affected_pathway	(0.023, 0.023)	(0.023, 0.024)	(0.027, 0.028)	(0.022, 0.023)	(0.023, 0.024)	(0.027, 0.028)
	animal_model	(0.038, 0.039)	(0.040, 0.041)	(0.046, 0.048)	(0.037, 0.038)	(0.039, 0.040)	(0.045, 0.046)
	genetic_association	(0.054, 0.056)	(0.056, 0.058)	(0.065, 0.067)	(0.053, 0.054)	(0.055, 0.057)	(0.064, 0.066)
	literature	(0.183, 0.186)	(0.189, 0.193)	(0.214, 0.218)	(0.179, 0.182)	(0.185, 0.189)	(0.210, 0.214)
	rna_expression	(0.037, 0.038)	(0.039, 0.040)	(0.045, 0.046)	(0.036, 0.037)	(0.038, 0.039)	(0.044, 0.045)
	somatic_mutation	(0.022, 0.023)	(0.023, 0.024)	(0.027, 0.028)	(0.022, 0.022)	(0.022, 0.023)	(0.026, 0.027)
auprc	affected_pathway	(0.010, 0.010)	(0.010, 0.011)	(0.009, 0.009)	(0.012, 0.012)	(0.012, 0.012)	(0.010, 0.010)
	animal_model	(0.018, 0.018)	(0.018, 0.019)	(0.015, 0.016)	(0.021, 0.021)	(0.021, 0.022)	(0.018, 0.018)
	genetic_association	(0.019, 0.020)	(0.020, 0.020)	(0.016, 0.017)	(0.022, 0.023)	(0.023, 0.024)	(0.019, 0.020)
	literature	(0.050, 0.052)	(0.052, 0.053)	(0.044, 0.045)	(0.059, 0.060)	(0.060, 0.062)	(0.051, 0.052)
	rna_expression	(0.013, 0.013)	(0.013, 0.014)	(0.011, 0.011)	(0.015, 0.016)	(0.016, 0.016)	(0.013, 0.013)
	somatic_mutation	(0.010, 0.010)	(0.010, 0.011)	(0.008, 0.009)	(0.011, 0.012)	(0.012, 0.012)	(0.010, 0.010)
top_20_hits	affected_pathway	(0.21, 0.23)	(0.21, 0.23)	(0.15, 0.17)	(0.20, 0.22)	(0.20, 0.22)	(0.14, 0.16)
	animal_model	(0.75, 0.79)	(0.74, 0.79)	(0.54, 0.58)	(0.70, 0.74)	(0.69, 0.73)	(0.50, 0.54)
	genetic_association	(1.06, 1.12)	(1.05, 1.11)	(0.77, 0.81)	(0.99, 1.05)	(0.98, 1.04)	(0.71, 0.76)
	literature	(1.94, 2.03)	(1.92, 2.01)	(1.40, 1.47)	(1.81, 1.89)	(1.79, 1.88)	(1.30, 1.37)
	rna_expression	(0.44, 0.47)	(0.43, 0.47)	(0.32, 0.34)	(0.41, 0.44)	(0.40, 0.44)	(0.29, 0.32)
	somatic_mutation	(0.20, 0.22)	(0.20, 0.22)	(0.15, 0.16)	(0.19, 0.21)	(0.19, 0.21)	(0.14, 0.15)
top_100_hits	affected_pathway	(1.46, 1.52)	(1.45, 1.51)	(1.09, 1.13)	(1.84, 1.91)	(1.83, 1.90)	(1.37, 1.43)
	animal_model	(1.90, 1.96)	(1.89, 1.95)	(1.41, 1.46)	(2.39, 2.47)	(2.38, 2.46)	(1.78, 1.85)
	genetic_association	(2.15, 2.22)	(2.14, 2.21)	(1.60, 1.65)	(2.71, 2.80)	(2.70, 2.78)	(2.02, 2.09)
	literature	(6.85, 7.01)	(6.82, 6.98)	(5.10, 5.23)	(8.64, 8.84)	(8.61, 8.81)	(6.44, 6.60)
	rna_expression	(1.63, 1.69)	(1.62, 1.68)	(1.21, 1.26)	(2.06, 2.13)	(2.05, 2.12)	(1.53, 1.59)
	somatic_mutation	(1.45, 1.51)	(1.45, 1.50)	(1.08, 1.12)	(1.83, 1.90)	(1.82, 1.89)	(1.36, 1.42)

E.4.5 Interaction effects

As mentioned in the main text, an illustrated by the above tables, in addition to a main effects model, we also considered how model performance could vary with other parameters such as choice of network and disease. In particular, we formally asked whether the differences – or interactions – we observed were in line or greater than those that would be expected by chance.

Interaction effects were omitted from our main analysis to avoid overfitting the data and a corresponding underestimation of the residual error and inflation in statistical significance. Given the large number of combinations possible, this was a risk even where the majority of interactions were not significant. This scenario was however in contrast to the current exploration were the sizes of any such effects were of interest per se. On the other hand, this exploratory analysis shows that the interaction terms that would improve the model involve poorly performing methods. Given their lack of interest in the final recommendations, including such terms would make the formal comparisons cumbersome, without providing any added value.

To simplify our analysis, motivated by standard statistical theory for multifactorial statistical screening designs (Montgomery, 2017), high order interactions, such as those between say CV scheme, network and disease, were omitted from our calculations and assumed to be little different to statistical noise. In contrast to standard screening methods, however, which typically address all binary or two level factors, all factor levels where considered adding to the complexity of the plots. In a further deviation, two-way interactions, say, were considered independently from all lower order terms, here one way or main effects, which were removed from the signal prior to analysis. This, done at the cost of over counting by one the total degrees of freedom in our data, served to improve interpretability since otherwise we would have one fewer interaction terms than combinations in the looked for effects. See figure 112 for an example and the html viewer in the supplementary file `interaction_html_viewer.zip` for other models and interaction terms.

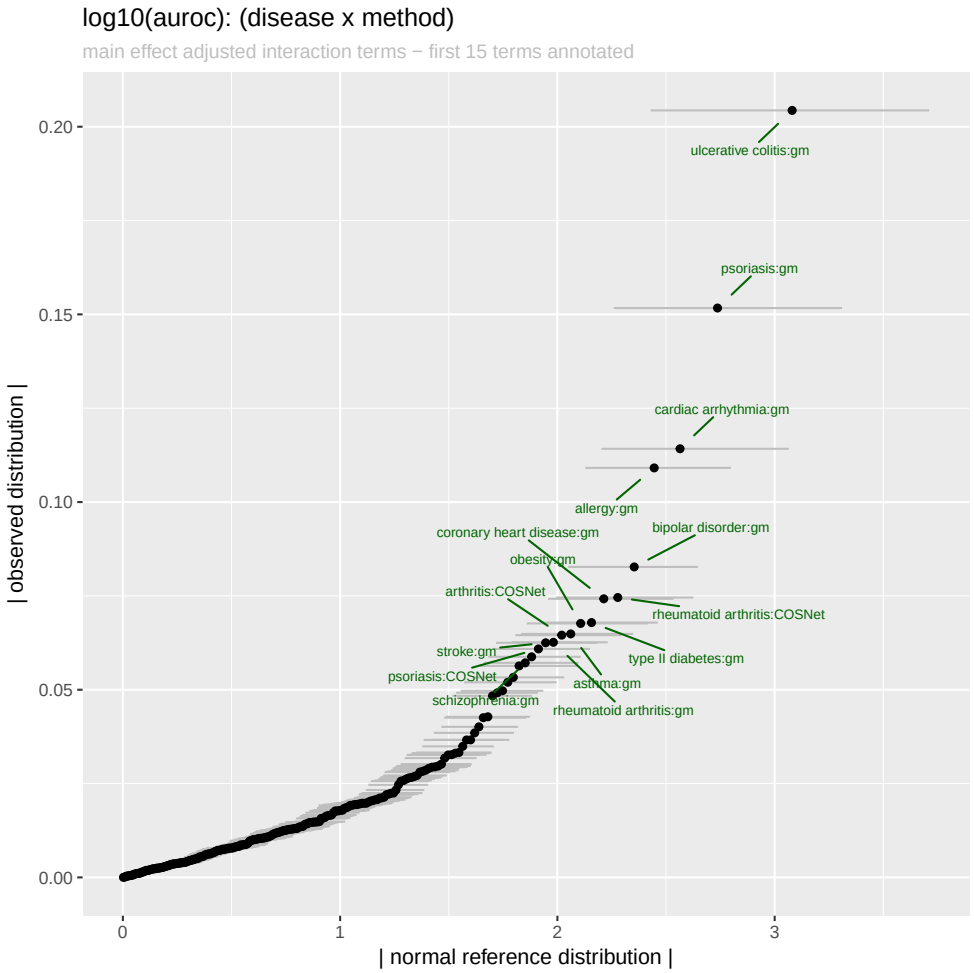


Figure 112: Graphical representation of the disease × method interaction. To investigate the possible effects of disease on individual method performance under the log10(AUROC), the absolute values of the the observed deviations from those predicted by a simple main effects model combination are plotted against those values that would be expected by chance alone under an assumption on normally distributed random noise. Deviations upwards from a straight line trend suggest interactions that are larger than would be expected by chance. Due to the use of absolute values -signs of interactions are difficult to interpret and can confuse comparisons- to 'fold over' the two distributions this is typically referred to as a *half normal plot*. To maintain a one to one correspondence between observed deviations and the set of two-way combinations of disease and method which would otherwise be lost by accounting degrees of freedom, main effect contributions for disease and method were removed prior to this analysis. As a guide to the underlying variability, the plot also includes 95% confidence intervals for the distribution of each absolute value normal reference value under re-sampling.

E.5 PACKAGE VERSIONS

Table 47: summary of the package versions used in this work and their source of download

Number	Package	Version	Source
1	acepack	1.4.1	CRAN
2	affy	1.54.0	Bioconductor
3	affyio	1.46.0	Bioconductor
4	affyPLM	1.52.1	Bioconductor
5	annotate	1.54.0	Bioconductor
6	AnnotationDbi	1.38.2	Bioconductor
7	arrayQualityMetrics	3.32.0	Bioconductor
8	assertthat	0.2.0	CRAN
9	backports	1.0.5	CRAN
10	base64	2.0	CRAN
11	base64enc	0.1-3	CRAN
12	BBmisc	1.11	CRAN
13	beadarray	2.26.1	Bioconductor
14	BeadDataPackR	1.28.0	Bioconductor
15	BH	1.65.0-1	CRAN
16	Biobase	2.36.2	Bioconductor
17	BiocGenerics	0.22.1	Bioconductor
18	BiocInstaller	1.26.1	Bioconductor
19	biomaRt	2.32.1	Bioconductor
20	Biostrings	2.44.2	Bioconductor
21	bitops	1.0-6	CRAN
22	broom	0.4.3	CRAN
23	Cairo	1.5-9	CRAN
24	caret	6.0-78	CRAN
25	caTools	1.17.1	CRAN
26	checkmate	1.8.2	CRAN
27	chron	2.3-50	CRAN
28	coda	0.19-1	CRAN
29	colorspace	1.3-2	CRAN
30	corrplot	0.84	CRAN
31	COSNet	1.10.0	Bioconductor
32	crayon	1.3.4	CRAN
33	curl	2.8.1	CRAN
34	CVST	0.2-1	CRAN
35	data.table	1.10.4	CRAN
36	DBI	0.6-1	CRAN
37	ddalpha	1.3.1	CRAN
38	DEoptimR	1.0-8	CRAN
39	devtools	1.13.4	CRAN
40	dichromat	2.0-0	CRAN

41	diffuStats	0.101.1	github
42	digest	0.6.12	CRAN
43	dimRed	0.1.0	CRAN
44	doMC	1.3.5	CRAN
45	dplyr	0.5.0	CRAN
46	DRR	0.0.2	CRAN
47	e1071	1.6-8	CRAN
48	EGAD	1.4.1	Bioconductor
49	emmeans	1.1.2	CRAN
50	estimability	1.3	CRAN
51	evaluate	0.10	CRAN
52	expm	0.999-2	CRAN
53	foreach	1.4.3	CRAN
54	formatR	1.5	CRAN
55	Formula	1.2-1	CRAN
56	gcrma	2.48.0	Bioconductor
57	gdata	2.17.0	CRAN
58	gdtools	0.1.7	CRAN
59	genefilter	1.58.1	Bioconductor
60	GenomeInfoDb	1.12.3	Bioconductor
61	GenomeInfoDbData	0.99.0	Bioconductor
62	GenomicRanges	1.28.6	Bioconductor
63	GEOquery	2.42.0	Bioconductor
64	GGally	1.4.0	CRAN
65	ggdendro	0.1-20	CRAN
66	ggplot2	2.2.1	CRAN
67	ggsci	2.8	CRAN
68	git2r	0.20.0	CRAN
69	glue	1.2.0	CRAN
70	gower	0.1.2	CRAN
71	gplots	3.0.1	CRAN
72	graph	1.54.0	Bioconductor
73	gridExtra	2.2.1	CRAN
74	gridSVG	1.6-0	CRAN
75	gsubfn	0.6-6	CRAN
76	gtable	0.2.0	CRAN
77	gtools	3.5.0	CRAN
78	hash	2.2.6	CRAN
79	hexbin	1.27.1	CRAN
80	highr	0.6	CRAN
81	Hmisc	4.0-3	CRAN
82	hms	0.4.1	CRAN
83	htmlTable	1.9	CRAN
84	htmltools	0.3.6	CRAN
85	htmlwidgets	0.9	CRAN

86	httr	1.3.1	CRAN
87	hwriter	1.3.2	CRAN
88	igraph	1.1.2	CRAN
89	illuminaio	0.18.0	Bioconductor
90	impute	1.50.1	Bioconductor
91	ipred	0.9-6	CRAN
92	IRanges	2.10.5	Bioconductor
93	irlba	2.2.1	CRAN
94	irr	0.84	CRAN
95	iterators	1.0.8	CRAN
96	jsonlite	1.5	CRAN
97	kableExtra	0.7.0	CRAN
98	kernlab	0.9-25	CRAN
99	knitr	1.16	CRAN
100	labeling	0.3	CRAN
101	latticeExtra	0.6-28	CRAN
102	lava	1.5.1	CRAN
103	lazyeval	0.2.0	CRAN
104	limma	3.32.10	Bioconductor
105	lpSolve	5.6.13	CRAN
106	lsmeans	2.27-61	CRAN
107	lubridate	1.7.1	CRAN
108	magrittr	1.5	CRAN
109	markdown	0.8	CRAN
110	memoise	1.1.0	CRAN
111	Metrics	0.1.3	CRAN
112	mime	0.5	CRAN
113	MLmetrics	1.1.1	CRAN
114	mlr	2.11	CRAN
115	mnormt	1.5-5	CRAN
116	ModelMetrics	1.1.0	CRAN
117	multcomp	1.4-8	CRAN
118	munsell	0.4.3	CRAN
119	mvtnorm	1.0-6	CRAN
120	NetPreProc	1.1	CRAN
121	numDeriv	2016.8-1	CRAN
122	openssl	0.9.7	CRAN
123	packrat	0.4.8-1	CRAN
124	parallelMap	1.3	CRAN
125	ParamHelpers	1.10	CRAN
126	PerfMeas	1.2.1	CRAN
127	pkgconfig	2.0.1	CRAN
128	plogr	0.1-1	CRAN
129	plotrix	3.7	CRAN
130	plyr	1.8.4	CRAN

131	png	0.1-7	CRAN
132	precrec	0.9.1	CRAN
133	preprocessCore	1.38.1	Bioconductor
134	prettyunits	1.0.2	CRAN
135	pROC	1.10.0	CRAN
136	prodlim	1.6.1	CRAN
137	progress	1.2.0	CRAN
138	proto	1.0.0	CRAN
139	PRROC	1.3	CRAN
140	psych	1.7.8	CRAN
141	purrr	0.2.4	CRAN
142	R6	2.2.2	CRAN
143	randomForest	4.6-12	CRAN
144	RANKS	1.0	CRAN
145	RBGL	1.52.0	Bioconductor
146	RColorBrewer	1.1-2	CRAN
147	Rcpp	0.12.14	CRAN
148	RcppArmadillo	0.8.300.1.0	CRAN
149	RcppParallel	4.3.20	CRAN
150	RcppRoll	0.2.2	CRAN
151	RCurl	1.95-4.8	CRAN
152	readr	1.1.1	CRAN
153	recipes	0.1.1	CRAN
154	reshape	0.8.7	CRAN
155	reshape2	1.4.2	CRAN
156	rlang	0.1.1	CRAN
157	rmarkdown	1.8	CRAN
158	robustbase	0.92-7	CRAN
159	ROCR	1.0-7	CRAN
160	rprojroot	1.3-1	CRAN
161	RSQLite	1.1-2	CRAN
162	rstudioapi	0.7	CRAN
163	rvest	0.3.2	CRAN
164	S4Vectors	0.14.7	Bioconductor
165	sandwich	2.3-4	CRAN
166	scales	0.4.1	CRAN
167	selectr	0.3-1	CRAN
168	setRNG	2013.9-1	CRAN
169	sfsmisc	1.1-1	CRAN
170	sqldf	0.4-11	CRAN
171	stargazer	5.2.1	CRAN
172	STRINGdb	1.16.0	Bioconductor
173	stringi	1.1.5	CRAN
174	stringr	1.2.0	CRAN
175	SVGAnnotation	0.93-2	github

176	svglite	1.2.1	CRAN
177	TH.data	1.0-8	CRAN
178	tibble	1.3.1	CRAN
179	tidyr	0.6.3	CRAN
180	tidyselect	0.2.3	CRAN
181	timeDate	3012.100	CRAN
182	TopKLists	1.0.6	CRAN
183	viridis	0.4.0	CRAN
184	viridisLite	0.2.0	CRAN
185	vsn	3.44.0	Bioconductor
186	whisker	0.3-2	CRAN
187	withr	2.1.1	CRAN
188	XML	3.98-1.12	CRAN
189	xml2	1.2.0	CRAN
190	xtable	1.8-2	CRAN
191	XVector	0.16.0	Bioconductor
192	yaml	2.1.14	CRAN
193	zlibbioc	1.22.0	Bioconductor
194	zoo	1.8-0	CRAN

REFERENCES

Ballouz, Sara, Melanie Weber, Paul Pavlidis, and Jesse Gillis

2017 "EGAD: ultra-fast functional analysis of gene networks", *Bioinformatics*, 33, 4, pp. 612-614, DOI: 10.18129/B9.bioc.EGAD.

Bersanelli, Matteo, Ettore Mosca, Daniel Remondini, Gastone Castellani, and Luciano Milanesi

2016 "Network diffusion-based analysis of high-throughput data for the detection of differentially enriched modules", *Sci. Rep.*, 6, 34841, pp. 1-12.

Bertoni, Alberto, Marco Frasca, and Giorgio Valentini

2011 "COSNet: a Cost Sensitive Neural Network for Semi-supervised Learning in Graphs", *Lect. Notes Comput. Sc.*, 6911, 4, pp. 219-234, DOI: 10.18129/B9.bioc.COSNet.

Bischl, Bernd, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, and Zachary M. Jones

2016 "mlr: Machine Learning in R", *J. Mach. Learn. Res.*, 17, 170, pp. 1-5.

Cho, Hyunghoon, Bonnie Berger, and Jian Peng

2016 "Compact integration of multi-network topology for functional analysis of genes", *Cell Syst.*, 3, 6, pp. 540-548.

Csardi, Gabor and Tamas Nepusz

2006 "The igraph software package for complex network research", *InterJournal*, Complex Systems, p. 1695, http://igraph.org.

Elkan, Charles and Keith Noto

2008 "Learning classifiers from only positive and unlabeled data", in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 213-220.

Erten, Sinan, Gurkan Bebek, Rob M Ewing, and Mehmet Koyutürk

2011 "DADA: degree-aware algorithms for network-based disease gene prioritization", *BioData mining*, 4, 1, p. 19.

Frasca, Marco, Alberto Bertoni, Matteo Re, and Giorgio Valentini

2013 "A neural network algorithm for semi-supervised node label learning from unbalanced data", *Bioinformatics*, 43, C, pp. 84-98.

Gronau, Ilan and Shlomo Moran

2007 "Optimal implementations of UPGMA and other common clustering algorithms", *Information Processing Letters*, 104, 6, pp. 205-210.

Jiang, Biaobin, Kyle Kloster, David F Gleich, and Michael Gribskov

2017 "AptRank: an adaptive PageRank model for protein function prediction on bi-relational graphs", *Bioinformatics*, 33, 12, pp. 1829-1836.

Kunn, Max

2008 "Building Predictive Models in R Using the caret Package", *J. Stat. Softwe.*, 28, 5, pp. 1-26.

Montgomery, Douglas C

2017 *Design and analysis of experiments*, John wiley & sons.

Mordelet, Fantine and Jean-Philippe Vert

2011 "ProDiGe: Prioritization Of Disease Genes with multitask machine learning from positive and unlabeled examples", *BMC Bioinformatics*, 12, 1, p. 389.

Mostafavi, Sara, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris

2008 "Genemania: a real-time multiple association network integration algorithm for predicting gene function", *Genome Biol.*, 9, S4, pp. 1-15.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd

1999 *The PageRank citation ranking: Bringing order to the web.* Tech. rep., Stanford InfoLab.

Picart-Armada, Sergio, Francesc Fernández-Albert, Maria Vinaixa, Miguel A Rodríguez, Suvi Aivio, Travis H Stracker, Oscar Yanes, and Alexandre Perera-Lluna

2017 "Null diffusion-based enrichment for metabolomics data", *PloS one*, 12, 12, e0189012.

Picart-Armada, Sergio, Wesley K Thompson, Alfonso Buil, and Alexandre Perera-Lluna

2017 "diffuStats: an R package to compute diffusion-based scores on biological networks", *Bioinformatics*, 34, 3, pp. 533-534, DOI: 10.1093/bioinformatics/btx632.

R Core Team

2016 *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Szklarczyk, Damian, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi P Tsafou, et al.

2014 "STRING v10: protein–protein interaction networks, integrated over the tree of life", *Nucleic Acids Res.*, 43, D1, pp. D447-D452.

TheUniProtConsortium

2017 "UniProt: the universal protein knowledgebase", *Nucleic Acids Res.*, 45, D1, pp. D158-D169, DOI: 10.1093/nar/gkw1099, eprint: /oup/backfile/content_public/journal/nar/45/d1/10.1093_nar_gkw1099/4/gkw1099.pdf, http://dx.doi.org/10.1093/nar/gkw1099.

Valentini, Giorgio, Giuliano Armano, Marco Frasca, Jianyi Lin, Marco Mesiti, and Matteo Re

2016 "RANKS: a flexible tool for node label ranking and classification in biological networks", *Bioinformatics*, 32, 18, pp. 2872-2874.

Valentini, Giorgio, Alberto Paccanaro, Horacio Caniza, Alfonso E. Romero, and Matteo Re

2014 "RANKS: a flexible tool for node label ranking and classification in biological networks", *Artif. Intell. Med.*, 61, 2, pp. 63-78.

Vandin, Fabio, Eli Upfal, and Benjamin J. Raphael

2011 "Algorithms for detecting significantly mutated pathways in cancer", *J. Comput. Biol.*, 18, 3, pp. 507-22.

Yang, Peng, Xiao-Li Li, Jian-Ping Mei, Chee-Keong Kwoh, and See-Kiong Ng

2012 "Positive-unlabeled learning for disease gene identification", *Bioinformatics*, 28, 20, pp. 2640-2647.

Zerbino, Daniel R, Premanand Achuthan, Wasiu Akanni, MRidwan Amode, Daniel Barrell, Jyothish Bhai, Konstantinos Billis, Cummins, et al.

2018 "Ensembl 2018", *Nucleic Acids Res.*, 46, D1, pp. D754-D761, DOI: 10.1093/nar/gkx1098, eprint: /oup/backfile/content_public/journal/nar/46/d1/10.1093_nar_gkx1098/2/gkx1098.pdf, http://dx.doi.org/10.1093/nar/gkx1098.

Zuur, Alain F and Ieno, Elena N and Walker, Neil J and Saveliev, Anatoly A and Smith, Graham M

2009 "GLM and GAM for count data", in *Mixed effects models and extensions in ecology with R*, Springer, pp. 209-243.