UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

Development of a machine learning-based pipeline able to predict genes associated with diseases and cell processes using interpretable network embeddings

Alexandre Filipe dos Reis Coelho

Mestrado em Bioinformática e Biologia Computacional

Dissertação orientada por:

Prof. Dr. Francisco Rodrigues Pinto

2022

Acknowledgments

First and foremost, I would like to thank my supervisor, Professor Francisco Pinto, not only for accepting to supervise me in this project but also for his guidance, relentless support to answer any questions, and for trying to fulfill all my curiosities about different research topics. Thank you for all the experiences, stories, and fun moments shared throughout this last year.

To BioISI and the Foundation of Science and Technology, for funding my project through the UIDB/04046/2020 and UIDP/04046/2020 Centre grants, and also through the awarding of the BioISI Junior Programme Research Grant.

To the team of the RNA Systems Biology Lab, for making me feel welcome since day one, for all the fun times we had throughout the last year, and for everything that I have learned from each member of the team. It was truly great to be involved in such a dynamic, diverse team, where everyone is always trying to learn more, even if it is outside of their domain, leading to fun moments such as explaining machine learning with apples and bananas. A special thanks to Cláudia, Tânia, Carolina, João, and Lara, you guys made this last year fly by, thank you for all our lunch break talks, all our mocking sessions, and for always being here whenever anyone needed a friendly word.

To all my college friends, looking back in time, I cannot believe how quickly these last 5 years have gone by. Thank you for always being present in the happiest and toughest times. My university experience would not be the same without you.

To all my childhood friends, that despite all our lives having taken different paths, try to maintain contact and never stop caring about each other.

A special thanks to my girlfriend, Marta, for always supporting me and being available to hear about my interests and frustrations. Thank you for always pushing me to want more and believe in my capabilities.

Finally, to all my family, but especially to my parents, Luísa and Paulo, and to my sister, Catarina. Without you, I would not be the person I am today. Thank you for all the education you gave me throughout all these years, for always pushing me to follow my dreams, for motivating me to achieve what I am capable of, and more importantly, for all the unconditional support I have always had.

Abstract

The rapid growth of genomic sequences has expanded the number of known proteins, however, their annotation mapping to known diseases and cell processes is still trailing. Protein mapping relies on experimental methods, such as linkage mapping studies, that are both expensive and time-consuming, so computational methods have emerged as alternatives for candidate prioritization.

Network-based algorithms are one kind of algorithm that has been developed for this purpose. Diseases and cell processes are resultant of the coordination of multiple physically interacting proteins, thus, biological networks can be used to search for new proteins that frequently interact with other disease or process associated proteins. Although several algorithms have been developed to tackle this problem, most of them do not use the full extent of available information within the network for their predictions, only relying on the known proteins associated with the disease/cell process of interest, or only using additional information from phenotypically similar diseases.

Here we propose GAP-MINE, a network-based algorithm with module-based interpretable embeddings, that uses additional modules to improve the prediction of new gene annotations. GAP-MINE is an adaptable algorithm with diverse possibilities in each of its several steps, such as the use of different classification algorithms or different protein interaction networks. We applied GAP-MINE in the discovery of newly associated genes for a total of 429 processes and 301 diseases. Using Random Walks with Restart as the scoring function, GAP-MINE shows median F-Measure scores consistently above 0.9. Compared to baseline and literature algorithms, GAP-MINE not only shows significantly better results but is also more precise and robust to the addition of noise, with its candidates showing biologically relevant annotations. GAP-MINE is therefore a suitable algorithm for gene annotation prediction and could be used to narrow down the number of genes to validate experimentally.

Keywords: Machine Learning; Network Biology; Gene-Disease Association; Gene-Process Association; Network Embeddings.

Resumo Alargado

A revolução tecnológica no mundo da sequenciação observada nas últimas duas décadas levou a um grande aumento no número de proteínas conhecidas. Porém, este aumento não foi correspondido com o aumento no seu número de anotações proteicas, em particular acerca do seu envolvimento em processos celulares e doenças. Atualmente, apenas cerca de 25% das proteínas humanas é conhecida por ter uma associação a uma doença. A lenta expansão de conhecimento destas associações deve-se essencialmente às técnicas experimentais necessárias para as descobrir, como por exemplo, estudos de ligação genética e *genome-wide association studies*, uma vez que falham quando aplicados a doenças heterogéneas, ou produzem números elevados de falsos positivos, respetivamente. Isto leva a um complexo processo de validação de resultados, que inevitavelmente desacelera o processo de anotação.

O desenvolvimento de métodos capazes de produzir um número mais restrito de candidatos surge então como uma necessidade para a mais eficaz descoberta de associações, com vários tipos de métodos computacionais a terem sido desenvolvidos nas últimas décadas. Uma fração destes métodos foca-se no uso de redes. Os mecanismos de processos celulares e doenças surgem da coordenação de múltiplas proteínas que interagem fisicamente, formando módulos de doenças e de processos celulares numa rede de interações biológicas. As redes biológicas podem ser representadas sob a forma de grafos, objectos matemáticos que representam como um conjunto de entidades interage entre si. Os grafos são formados por um conjuntos de nós (ou vértices), ligados entre si por arestas, permitindo a fácil representação e análise de redes, como as de interação de proteínas. Estas redes podem então ser exploradas de forma a encontrar padrões de interação que caracterizem as proteínas que fazem parte de um determinado módulo, de modo a mais tarde expandir este conhecimento e encontrar novas proteínas candidatas que possam eventualmente estar associadas a esse mesmo módulo e ser experimentalmente validadas. A deteção dos padrões que caracterizam as proteínas associadas a cada módulo depende do uso de métricas capazes de discriminar as relações de interesse que cada proteína apresenta, podendo estas métricas ir desde a medição da distância de cada proteína ao módulo, ao uso de métodos mais complexos de difusão, tais como *Random Walks with Restart*.

Muitos dos algoritmos já desenvolvidos focam-se no uso de métricas de proximidade, como o *Closeness*, que mede a centralidade de um determinado nó na rede, ou realizando um teste hipergeométrico de modo a analisar o enriquecimento de um nó em ligações com nós do módulo de interesse. A maioria dos algoritmos disponíveis na literatura baseia-se apenas na informação dada pela relação de cada nó com os nós do módulo em estudo, com uma minoria destes algoritmos a usar informação adicional de doenças fenotipicamente semelhantes. O mapeamento do interactoma humano ainda está por concluir, e, portanto, as redes de interação proteica usadas estão incompletas, faltando nós e arestas aos grafos contruídos. Para além disso, os processos de deteção de associações estão expostos à presença de falsos positivos. Tanto a incompletude das redes da interação como a presença de falsos positivos são fatores

que podem afetar em larga escala as previsões de algoritmos que apenas se baseiam no próprio módulo, dificultando então o processo de seleção de novos candidatos. Será, portanto, interessante o desenvolvimento de um algoritmo capaz de usar uma maior fração dos dados ao seu dispor, sem depender do uso de informação, como a semelhança fenotípica, que permita uma maior precisão aquando da previsão de novos candidatos, mas também uma maior robustez sob a presença de alterações na rede de interação ou de anotações incorretas.

Neste trabalho, é então proposto o desenvolvimento de um novo algoritmo para a previsão de novos candidatos associados com doenças ou processos celulares designado de *Gene Annotation Prediction using Module-based Interpretable Network Embeddings (GAP-MINE)*. A maior contribuição deste algoritmo é o uso de *network embeddings* facilmente interpretáveis num contexto biológico. *Network embeddings*, são vetores usados para explicar a relação de cada nó da rede com os restantes através de um espaço multidimensional. Estes podem ser adaptados ao contexto de nosso problema, e assim explicar a relação que cada nó tem com cada módulo, criando, portanto, uma representação multidimensional que pode ser usada para descobrir os padrões que caracterizam as proteínas associadas a um determinado módulo, usando informação adicional contida no restante vetor. Para além disso, ao exprimirem a relação de cada nó com os diferentes módulos, estes vetores permitem uma melhor interpretação dos resultados, uma vez que permitem a análise dos módulos escolhidos.

O algoritmo desenvolvido é composto por 6 passos que podem ser facilmente adaptados consoante a natureza do problema. Primeiramente, a rede de interação proteica foi construída utilizando as interações disponíveis na bases de dados APID e HuRI, juntamente com as anotações de associação de proteínas a processos e doenças, provenientes das bases de dados REACTOME e DisGeNET, respetivamente. A rede criada apresenta um total de 17 204 nós, ligados por 260 960 arestas. Foram criados três tipos de módulos: um de processos celulares, num total de 429, e dois do doença, que variam consoante a conectividade do módulo em si, porém tendo origem nos mesmo dados, totalizando um total de 203 aquando da utilização de módulos conectados, e de 301 aquando da utilização de módulos mais dispersos na rede. Cinco diferentes métricas foram, de seguida, aplicadas à rede (*Hypergeometric Test*, *Closeness*, *Betweenness*, *Fraction Betweenness* e *Random Walks with Restart*) sendo modificadas das suas formas normais de forma a explicar como cada nó se relaciona com os nós associados a um determinado módulo. Ao serem aplicadas aos diferentes módulos, é então formado um *embedding* para cada nó de dimensão igual ao número de módulos presentes na rede, resultando na matriz de *embeddings* de *N* vs. *M* dimensões, onde *N* é o número de nós da rede e *M* o número de módulos. À matriz é depois aplicado um passo de seleção, onde, para a classificação de um determinado módulo são selecionados os módulos que mais contribuem para a discriminação das proteínas associadas e não associadas ao mesmo. Tendo os módulos mais relevantes selecionados, os *embeddings* são fornecidos a um modelo de regressão logística, um algoritmo de classificação, que é treinado e otimizado com uma validação cruzada de 10 passos. Este algoritmo de classificação é depois avaliado usando um conjunto de teste, e aplicado para a totalidade dos dados de modo a prever as novas associações. Por fim, as associações previstas são validadas através comparação dos termos da *Gene Ontology* e da *Human Phenotype Ontology* (este último exclusivamente aplicado a proteínas de doença) em comum com os termos enriquecidos das proteínas do módulo alvo, e pela procura do identificador da proteína e do nome do módulo em títulos e resumos da literatura.

O algoritmo GAP-MINE foi primeiramente comparado com um modelo padrão que apenas utiliza os valores obtidos para o módulo que se pretende classificar. Verificou-se que as *Random Walks with Restart* são as melhores métricas a ser usadas para a previsão de novas proteínas associadas aos módulos, obtendo valores medianos de *F-Measure* acima de 0.9 utilizando tanto o nosso algoritmo, como os modelos padrão. Comparando o nosso algoritmo com os modelos padrão, foi possível observar que foram obtidos resultados significativamente melhores em 2 dos 3 tipos de módulos aquando da utilização

de tanto as *Random Walks with Restart*, como do *Closeness* como métricas, obtendo, no entanto, piores resultados usando *Betweenness* e *Fraction Betweenness*.

Analisando as *Random Walks with Restart* em pormenor, foi possível verificar que a melhoria dos resultados obtidos se deveu a um aumento da precisão em todos os módulos, à custa de capturar um menor número de positivos. O mesmo comportamento foi verificado em testes feitos onde a rede utilizada foi alterada para simular casos de falta de informação ou da inclusão de falsos positivos.

A combinação GAP-MINE com *Random Walks with Restart* foi também comparada com outros algoritmos já estabelecidos (*GenePANDA*, *Raw* e *MaxLink*), tendo sido observado que o nosso algoritmo é capaz de obter resultados significativamente melhores do que qualquer um dos três algoritmos.

De forma geral, as previsões feitas pelo nosso algoritmo mostram-se enriquecidas em termos relevantes e relacionados com os termos associados aos diferentes processos e doenças, tendo também sido possível verificar a presença na literatura de algumas das novas associações.

Concluindo, o nosso algoritmo mostra-se ser uma alternativa capaz de prever novas associações entre proteínas e processos celulares/doenças, com uma melhoria de precisão, o que deverá facilitar o processo de validação experimental, e acelerar a descoberta de novas associações.

Palavras-chave: Aprendizagem automática; Redes Biológicas; Associações Doença-Proteína; Associações Processo-Proteína; *Network Embeddings*.

Contents

List of Figures

List of Tables

Acronyms

GO	Gene Ontology
GWAS	Genome-wide Association Studies
CPS	Common Pathway Scanning
GenePANDA	Gene Prioritizing Approach using Network Distance Analysis
DIAMOnD	Disease Module Detection
DiaBLE	DIAMOnD Background Local Expansion
S2B	Specific Betweenness
RWR	Random Walks with Restart
PRINCE	Prioritization and Complex Elucidation
ORIENT	Neighbor-favoring Weight Reinforcement
Cardigan	Charting Disease Gene Associations
ProDiGe	Prioritization of Disease Genes
GAP-MINE	Gene Annotation Prediction using Module-based Interpretable Network Embeddings
OMIM	Online Mendelian Inheritance in Man
CTD	Comparative Toxicogenomics Database
ClinGen	Clinical Genome Resource
CGI	The Cancer Genome Interpreter
PsyGeNET	Psychiatric Disorders Gene Association Network
MGD	Mouse Genome Database
RGD	Rat Genome Database
HPO	Human Phenotype Ontology
PPI	Protein-Protein Interactions
Y2H	Yeast two-hybrid
APID	Agile Protein Interactomes DataServer
HuRI	The Human Reference Interactome
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
OPLS-DA	Orthogonal Partial Least Squares Discriminant Analysis

VIP	Variable Importance in Projection
OBO	Open Biomedical Ontology
GOAE	Gene Ontology Enrichment Analysis
HGNC	HUGO Gene Nomenclature Committee
SCA	Seed Connector Algorithm
API	Application Programming Interface
STRING	Search Tool for the Retrieval of Interacting Genes/Proteins
HT	Hypergeometric Test
C	Closeness
B	Betweenness
FB	Fraction Betweenness
F	F-Measure
P	Precision
R	Recall
RA	Rheumatoid Arthritis

Chapter 1

Introduction

In the last decades, sequencing improvements have led to a huge increase in the number of known proteins. However, the number of proteins with known association with diseases and cell processes is still trailing, with only about 25% of human proteins having a known disease association, 21% having a curated process annotation (REACTOME), and 85% having an associated biological process (GO) [1]. Besides gene-wise experiments, these annotations are usually found by genome-wide association studies (GWAS) [2] or linkage mapping studies [3]. However, these methods either produce a very large number of candidates (and inevitably many false positives) for experimental validation or fail when applied to complex heterogeneous diseases [4]. This leads to an expensive and arduous process of experimental validation that slows down the annotation task, and so, other methods able to present a more stringent number of candidates have been developed.

The use of network-based methods has been one of the major approaches [5]. Cell Processes and Disease mechanisms emerge from the coordinated function of multiple physically interacting proteins, that constitute Process or Disease network modules [6]. Biological networks, created with experimental interactome information, together with the known protein annotations, allow for the search of patterns that characterize the module proteins. Candidate proteins that present similar patterns to the ones observed for known module proteins can be found and identified as suitable candidates for experimental validation. The patterns that characterize the module proteins can be defined by a set of different network metrics that go from measuring a protein distance to the module, to more complex propagation methods, such as random walks with restart.

Several algorithms already use biological networks for the prediction of functional annotations. George et al. [7] and Yin et al. [8], developed algorithms based on the protein network distance and their shortest paths; Gaula et al. [9], Ghiassian et al. [10], and Petti et al. [11] use implementations of the hypergeometric test to select proteins enriched in interactions with known module members; Guney et al. developed NetShort [12], a closeness-based algorithm that scores proteins by the inverse of their distance to the module; Garcia-Vaquero et al. [13] and Maiorino et al. [14] developed methods based on how a node connects two modules of interest; finally, diffusion-based algorithms score proteins by performing random walks or simulating diffusion processes throughout the network. Vandin et al. developed Raw [15], a diffusion-based algorithm. Köhler et al. developed the Random Walks with Restart (RWR) [16]. Due to its good performance, the RWR has been incorporated into several prediction algorithms like PRINCE [17], VAVIEN [18], and ORIENT [19].

Table 1.1 - Network-based methods for the prediction of new gene-disease associations, representing the use of different network proximity or connectivity metrics.

Name	Year	Type	Author
CPS	2006	Shortest Paths	George et al. [7]
GenePANDA	2017		Yin et al. [8]
MaxLink	2014	Hypergeometric Test	Gaula et al. [9]
DIAMOnD	2015		Ghiassian et al. [10]
DiaBLE	2020		Petti et al.[11]
NetShort	2012	Closeness	Guney et al. [12]
S2B	2018	Betweenness	Garcia-Vaquero et al. [13]
Flow Centrality	2020		Maiorino et al. [14]
RWR	2008	Random Walks with Restart/Diffusion	Köhler et al. [16]
PRINCE	2010		Vanunu et al. [17]
Raw	2011		Vandin et al. [15]
VAVIEN	2011		Erten et al. [18]
ORIENT	2013		Le et al. [19]

Additionally, some authors have developed network-based prioritization algorithms like Cardigan [20] and ProDiGe [21] that use additional information from phenotypically similar diseases to complement the disease module information and consequently improve the scoring of candidate proteins. These approaches are useful to find candidates for less-characterized diseases. However, they still require previous knowledge of the similarities between the diseases which might not be available. Not only that, but it is also possible that relevant information for the discrimination of candidate proteins might be found in other diseases and processes that are not phenotypically similar.

Another relevant factor lies in the fact that current protein interaction networks are known to be incomplete, lacking both nodes and edges that have not yet been experimentally detected [22]. Proteins annotated with a given process or disease are also incomplete and contaminated with false positive associations. When making predictions of new proteins associated with a given process or disease, those predictions can be severely affected by the referred incompleteness around the corresponding network module. Therefore, it might be useful to use additional modules when predicting proteins associated with a specific module.

1.1 Objectives

This work's principal objective is the development of an algorithm for the prediction of genes associated with diseases and cell processes with the use of targeted network embeddings, entitled Gene Annotation Prediction using Module-based Interpretable Network Embeddings (GAP-MINE). Network embeddings should be a suitable representation of how each candidate interacts with additional modules and thus provide additional information for the candidate classification without the need for previous phenotypic similarity knowledge.

Furthermore, this project can be divided into three different goals:

1. Define what proximity/connectivity metrics provide more precise results when used as predictors for new gene-module associations.
2. Establish a network embedding-based pipeline able to surpass the base metrics performances, not only in a complete network but also under different network conditions.
3. Compare the developed pipeline's performance with already established methods and validate the new predictions.

1.2 Document Structure

This document is structured as follows:

- Chapter 2 includes the theory fundamentals required to understand the methodology, as well as their adaptation to the problem.
- Chapter 3 focuses on the methodology taken for the development of this work, mainly, the different steps of GAP-MINE.
- Chapter 4 presents the obtained results in the multiple performance tests performed.
- Chapter 5 presents the discussion of the obtained results.

Chapter 2

Background

2.1 Gene-Process and Gene-Disease Associations

The completion of the human genome project [23] together with the technological revolution in the sequencing world has provided remarkable opportunities for a better understanding of human diseases [24]. While Mendelian diseases have particular genes directly causing the disease, common disease-associated genes explain only a fraction of the disease behavior (and are not strictly causal), as these diseases result from the coordination of multiple interacting genes [25]. Additionally, while monogenic diseases exist, many disease phenotypes are modified by additional genes, so it is rare to have diseases with a single associated gene [6,26].

Over the last decades, several initiatives were created to identify and unify the knowledge of disease-associated genes. The Online Mendelian Inheritance in Man (OMIM) [27] is a knowledge base of human genes associated with genetic disorders with over 26 000 annotated genes. DisGeNET [28–30] is a database with the collection of more than 620 000 associations between more than 17 500 genes and 24 000 diseases (Mendelian, complex, and environmental), along with 117 000 genomic variants. DisGeNET comprises the gene-disease associations from four different database types:

- Curated: gene-disease associations from curated resources. Includes data from UniProt [31], CTD [32], Orphanet, ClinGen [33], Genomics England, CGI [34], and PsyGeNET [35].
- Animal Models: associations from animal models mapped into human genes. Includes data from CTD, MGD [36], and RGD [37].
- Inferred: associations inferred from HPO [38] and variant-disease associations. Includes data from HPO and ClinVar [39].
- Literature: associations extracted from textual sources of interest.

Life at a cellular level is controlled by a network of molecular reactions, with the annotation of the involved proteins being required to try to understand the full extent of the cellular life. Several resources annotate the genes involved in the several processes occurring in the cell. One of these resources is the Reactome Knowledgebase [40], a literature-based database for the description of the human biological processes that characterize the genes, proteins, and molecules that take part in more than 11 000 reactions. Reactome has over 10 000 annotated protein-coding genes and 25 000 proteins that interact with 1 856 molecules.

2.2 Protein-Protein Interactions

Protein-protein interactions (PPI) are central to the basic functioning of cellular life [41], with proteins constituting more than half of the dry mass of the cell [42] and with more than 650 000 estimated interactions [43]. Binary and complex (composed of more than 2 proteins) protein-protein interactions are involved in core processes for cell life such as transcription, translation, cell-cell adhesion and communication, protein synthesis, and degradation, among others. Due to the large scale of biological processes dependent on PPIs, it is easy to correlate disruptions in the cell life with changes in PPIs. While most of these disruptions can be suppressed by the cell systems, some might be directly linked to diseases, being causal or being directly linked to the disease phenotype or progression. Additionally, as previously stated, diseases and cell processes emerge from the coordination of multiple interacting proteins, and so, PPIs can provide suitable ways to find new involved proteins in the neighborhood of the already known associated proteins [41].

The map of PPIs is an ongoing issue, with different experimental techniques being employed depending on the problem. One of the principal mapping techniques is the yeast two-hybrid (Y2H) [44], the predecessor to most experimental techniques [45]. Y2H splits a transcription factor into its DNA binding domain and its transcription activation domain and connects each one to one of the two proteins to be tested; in case of interaction, the domains are close enough to present activity and thus activate the transcription of a reporter protein, that produces a signal indicating the occurred interaction.

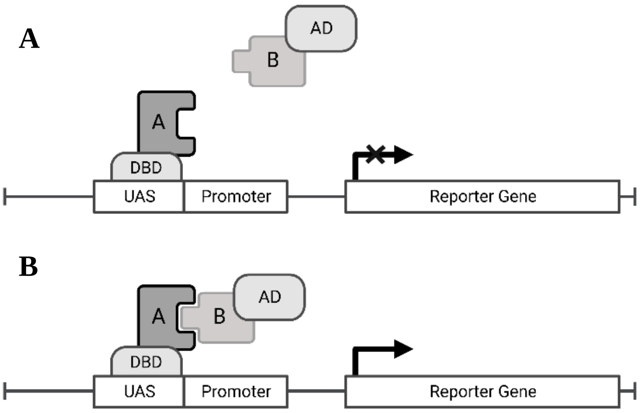


Figure 2.1 - Y2H system overview. A transcription factor is split into its DNA binding domain and activation domain. Each domain is coupled to one of the testing proteins, if they interact, the transcription factor activates the transcription of a reporter protein (B). Otherwise, no protein is transcribed (A). Created with BioRender.com.

However, Y2H techniques have certain biases, such as the fact that only proteins that can reach the nucleus and that cannot independently activate the used reported gene can be used, increasing the number of false negatives. Membrane proteins are of particular interest for drug development, however, their mapping is particularly challenging due to their hydrophobic characteristics, and therefore requires the adaptation of the Y2H technique to enable their detection. Co-complex methods can also be used and allow for the detection of protein complexes and more physiological-like results as no tagging is required [46].

Several initiatives have been published in the last years with different mapping approaches and coverages of the human interactome. The Agile Protein Interactomes DataServer (APID) [47,48] is a public resource with the collection of interactomes for more than 1100 organisms. APID integrates the information of five primary databases (BioGRID [49], DIP [50], HPRD [51], IntAct [52], and MINT [53]) together with experimentally resolved 3D structures (tested between protein pairs to find protein-protein contacts that allow the interaction), making APID database a combination of literature curated interactions, experimentally assessed interactions and inferred interactions. As of March 2021, APID has a total of 667805 different PPIs (binary or complex), covering 46.7% of the known human proteome. APID defines three different confidence thresholds for the interactions, depending on whether all interactions (inferred or not) are included (level 0), all interactions have 2 or more experimental pieces of evidence (level 1), or all interactions were detected by at least one binary method (instead of co-complex detection methods – level 2).

HuRI [54] is an ongoing initiative for the mapping of the human binary interactome. HuRI's goal consists of the systematic mapping of the human interactome, performing pairwise assessments for more than 90% of the human protein-coding genome, using three different Y2H techniques in 9 independent screenings of the search space. Version 9.1 of the human ORFeome covers 17 408 protein-coding genes, forming more than 150 million pairwise combinations to be searched. HuRI's latest version (HI-III-20) encompasses 8 275 proteins, containing a total of 52 569 interactions.

APID and HuRI databases can be combined to try to maximize the interactome coverage. The APID's dependence on literature curated PPIs makes this database biased towards having proteins known to have important roles in cell processes and diseases with more interactions (and more dense networks around them) as more experiments are performed with them. On the other hand, HuRI's Y2H techniques, search the human proteome in an unbiased way, however, have difficulties finding membrane PPIs, something that APID should not have as membrane proteins are common targets for the development of therapeutic solutions.

2.3 Graphs

A graph is a mathematical object used to represent pairwise relationships between a set of entities. Graphs are composed of a set of nodes (also called vertices), connected by a set of edges [55]. Graphs have a variety of applications, ranging from representing chemical molecules (where nodes represent atoms and edges represent chemical bonds) to representing social networks (where nodes represent different people and edges represent whether they are friends) [56]. In biology, graph implementations are used for single-cell transcriptome analysis, to represent neurons and their connections, or to represent physical or regulatory interactions between biomolecules within a cell, for example [57].

Graphs can be grouped into two main categories, depending on whether their edges are directed or undirected. Directed graphs are used to represent relationships where one node's interaction with another does not guarantee the opposite. These graphs are often used in gene regulatory networks where a given transcription factor interacts with a target gene, but the contrary does not necessarily occur. On the other hand, edges in an undirected graph simply state that the connected nodes bilaterally interact with each other. Undirected graphs can be used to represent physically interacting proteins, for example.

A second categorization of graphs can be done depending on whether these include weights or not. Weighted graphs require additional information than just the interactions but can provide more accurate insights. In biological terms, these weights can be seen as the expression level of each gene (weighted nodes) or the interaction probability of two proteins (weighted edges).

Graphs can be further customized with the inclusion of auxiliary information, annotating nodes, edges, or the whole graph. The added information can have a variety of characteristics, ranging from labels and discrete or continuous attributes as node or edge features [58].

Graphs can also be used for the application of the Guilt-by-Association principle. In a graph with N nodes and M edges, additional information can be used to label the nodes accordingly to some class, such as their association to a disease, which allows for inference of the class of the remaining nodes of a graph. The labeled nodes can positively or negatively influence the presence of positive labeled nodes in their neighborhood. In cases where nodes positively influence the presence of positive labeled nodes, nodes with an enriched number of edges between them and the positive labeled ones can be predicted to be of the same class [59]. Translating this principle into our problem, genes enriched with interactions with module-associated genes can be predicted to be associated with that same module. As an example, let's take a set of 7 genes, connected by 9 edges, with 3 of those genes being known to be associated with disease X (Figure 2.2). By direct analysis of this graph, we can infer that gene D might also be associated with disease X, as 3 of the 4 interactions it has link it to disease-associated genes.

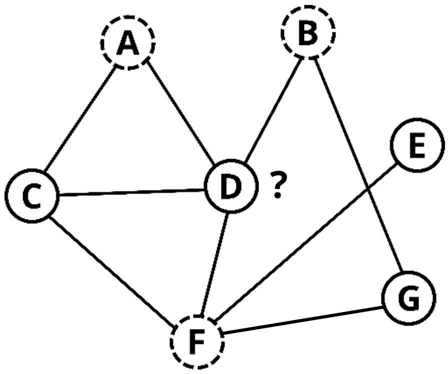


Figure 2.2 - Guilt-by-association example. In a network with only 7 nodes and with 3 of them being associated with disease X (nodes A, B, and F), it is possible to infer that node D might also be associated with disease X. Figure created with Cytoscape [60].

2.3.1 Network Proximity Metrics

Graphs can range from having a small set of nodes and edges (as seen in Figure 2.2), to having hundreds of thousands of nodes and edges. While it is easy to infer graph properties in small graphs, it is impossible to do the same in bigger graphs, and so, alternatives to represent the graph's information are required. Proximity metrics are useful ways to characterize each node and its neighborhood and can be later given to models that effectively find new information within the thousands of existing nodes. Different metrics, with varying complexity, can be used depending on the purpose of the study.

One of the simpler metrics to compute is the shortest path between two nodes. The shortest path is defined as the least number of edges required in order to get from one node to another. The shortest paths are in fact a base metric used on a large scale on other metrics.

Closeness is the measure of how central to the network a given node is. Scores are computed by doing the inverse of the average length of the shortest paths from a node to all other nodes in the network. Closeness can be computed by equation 2.1, where N is the number of nodes in the network and $d(y,x)$ is the length of the shortest path that connects nodes x and y. High closeness values are associated with nodes more closely connected to the other nodes, and thus, more central.

$$C(x) = \frac{N}{\sum_y d(y,x)} \tag{2.1}$$

Betweenness is a centrality measure that scores a node presence in the shortest paths that connect two given nodes. This metric is calculated by equation 2.2, where σ_{yz} is the number of shortest paths between nodes y and z, $\sigma_{yz}(x)$ are the number of shortest paths between y and z that pass through x, and y and z are any two nodes in the network that are not x.

$$g(x) = \sum_{x \neq y \neq z} \frac{\sigma_{yz}(x)}{\sigma_{yz}} \tag{2.2}$$

Random Walks with Restart can also be applied to a graph. Starting from a specific node, nodes are scored by randomly walking the network, i.e., going from one node to another following the available edges that connect them. At each step, there is a probability of the walk restarting from the initial nodes. At the end of n iterations, nodes are scored according to the number of times they were visited. The number of iterations is large enough to ensure that node visiting probabilities are stable.

2.4 Graph Embeddings

Embeddings are low-dimensional vectors that translate high-dimensional vectors, and consequently, ease the use of machine learning models. Embeddings are used in a variety of cases, such as linguistics, where Google's Word2vec [61], one of the most popular embedding algorithms, is used to compute embeddings that individually describe the words present in a text corpus, with words that share similar contexts having similar embedding vectors.

A special use case for embeddings is their application to graphs. Graph embeddings are low-dimensional vector representations of the graph that try to preserve certain properties of it. In general, graph embeddings can be divided into four different categories [58]:

- *Node Embeddings*: the most common type of embedding. Represent each node in a low dimensional space, with nodes that are close to each other in the graph having similar vectors. Node embeddings are commonly computed by first- or second-order proximity. First-order proximity gets the closer nodes by the weight of the edge that connects them (higher weight, more similar). Second-order proximity is computed by the similarity of the neighborhood of the two compared nodes (more common neighbors, more similar).
- *Edge Embeddings*: represent edges as a low-dimensional vector. Edge embeddings can typically be represented by the embedding of the node pair, allowing it to make it comparable to other nodes or predict the existence of a link between two nodes.
- *Hybrid Embeddings*: represent the combination of different types of graph components. Hybrid embeddings can combine the embedding of nodes and edges or the embedding of nodes and communities. Node and edge embeddings can be referred to as substructure embeddings and have been used in the representation of subgraphs. Community embeddings take into account the structure of heavily connected regions in a graph (communities) for the computation of the embedding [62].
- *Whole-Graph Embeddings*: typically used to represent a small graph into a single vector and to compare the similarities of two different graphs (with similar vectors). Whole-graph embeddings are computationally expensive as it is required to capture all properties of a graph. Thus, compromises between the computational time and the embedding quality need to be made. One solution for this problem is the hierarchical embedding of a graph, with the processing of graph structures at different levels and their later concatenation into one vector.

2.5 Machine Learning

Machine Learning is a field of computer science focused on the development of algorithms able to learn patterns from data in order to make predictions about new observations. Machine Learning algorithms can be categorized into three different kinds: supervised, semi-supervised, and unsupervised [63,64].

2.5.1 Types of Machine Learning

Supervised Learning models are used to make predictions about unseen unlabeled data. Supervised learning requires labeled data as input and creates a model able to, first, discriminate the own training data labels, to then predict the labels of new unseen data. Supervised learning problems can be further separated into two categories: *Classification* and *Regression*. The difference between these two categories relies on the type of the predicted label. Classification models predict categorical values, with these either being binary values (where there are only two possible outcomes) or multiclass values (where there are > 2 possible outcomes). Regression models, on the other hand, try to find a relationship between the training data and a continuous response variable, meaning that the predicted value will be in a continuous range.

Semi-Supervised Learning has the same goal as the supervised learning models, however, here, the training data is composed of both labeled and unlabeled observations in order to produce a better model, as unlabeled examples allow for the addition of information to the model.

In **Unsupervised Learning** models, only unlabeled data is used, i.e., no knowledge about the data distribution is previously known, and so, the model is responsible to learn and categorize the data into different groups, for example, clusters.

For the proposed problem of this work, we want to predict whether a gene is associated with a specific module or not. Thanks to the already known gene-disease and gene-process associations, we can create a labeled dataset that can be used to train and test a supervised machine learning model, performing a classification task to predict if a gene is associated with the module.

2.5.2 Supervised Learning – Logistic Regression

Logistic regression is one simple, but powerful algorithm, that despite its name, is used for classification proposes [63–65]. This linear model was first built when computers were not around, but there was already a need for linear classification models. When in presence of a binary classification problem, we can model the two possible outputs as 0 and 1, where $y = 1$ is the label to be predicted. Thus, a function whose range is from 0 to 1, can be defined to predict whether an input is negative or positive. The function used in logistic regression is the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

Where x is the combination of the sample predictor variables:

$$x = w_0 x_0 + w_1 x_1 + \cdots + w_m x_m \tag{2.4}$$

The sigmoid function output can be interpreted as the probability of a given sample belonging to $y = 1$, with the probability of a sample belonging to $y = 0$, being consequently given by $1 - f(x)$. In a standard case, it is considered that any sample whose probability is above 0.5 belongs to the positive class, however, the probability threshold can be adjusted to the problem. The logistic regression can also be adapted to multiclass classification.

The building of the logistic regression model lies in the optimization of its maximum likelihood function, i.e., maximizing the likelihood of the training data according to the model. The maximum likelihood function is given by:

$$L = \prod_{i=1 \ldots N} f(x_i)^{y_i} \left(1 - f(x_i)\right)^{(1-y_i)} \tag{2.5}$$

Where i identifies an individual in a set of N individuals. The optimization of the maximum likelihood can be seen as the minimization of the logarithm of the likelihood function (cost function):

$$C = -\sum_{i=1}^{N} y_i \ln f(x) + (1 - y_i) \ln\left(1 - f(x)\right) \tag{2.6}$$

2.5.3 Performance Metrics

When creating and applying a model, it is necessary to evaluate its performance. As previously stated, binary classification problems only have two possible outcomes (typically 0 or 1), and so, a classifier's predictions can be grouped into four categories, depending on whether the predicted label was in conformance with the true label or not.

A confusion matrix can be used to summarize the algorithm performance, as it shows the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Table 2.1 - Confusion matrix of a binary classification problem. Each label is classified by the comparison of its true and predicted class, being classified as a TP, FP, TN, or FN.

		Predicted Label	
		Positive	Negative
True Label	Positive	TP	FN
	Negative	FP	TN

In our problem, a TP is a module node predicted to be part of the module, a TN is any node not known to be part of the module to be classified as not being part of it, and a FP is any node not known to be part of the module classified as being part of it. Finally, FN represents the module nodes that were classified as not being part of it.

Frequently, classification algorithms are evaluated by their **accuracy**. The accuracy can be interpreted as the fraction of the correct predictions, i.e., the sum of TP and TN divided by the total number of labels, as shown in equation 2.7:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

However, this metric should not be applied to imbalanced datasets. Imbalanced datasets are datasets that have an abnormal proportion of positive or negative labels, i.e., instead of having a close to 50/50 proportion of positive and negative examples, some datasets might have distributions of 10/90 or 1/99 positive to negative samples. In our case, process and disease modules have between 50 and 300 genes, whereas the biological network has more than 17 000 genes, leading to a very imbalanced dataset. This is a problem for accuracy as high accuracy scores can be achieved even if no labels are predicted to be of the positive class. Taking our dataset example, a disease module of 300 genes that has no genes to be predicted to be part of it will still achieve an approximate accuracy of 98%, not reflecting the poor model performance.

Other metrics can thus be computed to better evaluate a model's performance, especially on imbalanced datasets. Precision is the fraction of true positives in the total number of positive predictions (equation 2.8):

$$Precision = \frac{TP}{TP + FP} \tag{2.8}$$

Recall is the fraction of positives that were predicted as such (equation 2.9):

$$Recall = \frac{TP}{TP + FN} \tag{2.9}$$

In theory, a good machine learning model has high precision and recall scores, however, combinations of high precision scores with low recall ones, or vice-versa are possible. As a result, it is not always easy to assert a model performance by the direct analysis of these two metrics. The F-Measure (equation

2.10) is a combination of precision and recall scores through their harmonic mean that allows for the evaluation of the model performance with a single value.

$$F - Measure = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.10}$$

Nonetheless, it is important to analyze the precision and recall scores obtained by the model, as depending on the model goal, it might be more useful to have higher precision scores at the expense of recall, for example.

2.5.4 Cross-Validation

A standard way to evaluate a model's performance is to simply use a data partition to train it and another to test its performance. However, a model can suffer from underfitting or overfitting if it is too simple or too complex for the used training data. A possible solution to this problem is to use an additional set for the validation of the model, prior to its application on the test data. However, this implies a third data split, which can greatly reduce the number of samples used to learn the model.

Cross-Validation is a technique that allows for the validation of the model, without requiring a validation set. A basic approach to cross-validation is k-fold cross-validation, where the training set is split into k folds, with $k - 1$ folds being used to train the model, and the remaining fold used to evaluate the model performance. The process is repeated k times, with each different fold being used once to evaluate the model. Figure 2.3 summarizes a 5-fold cross-validation procedure. In the end, the k different models with k different performances are built. If the performance is satisfactory, the model is then trained with the entire training dataset and applied to the testing data.

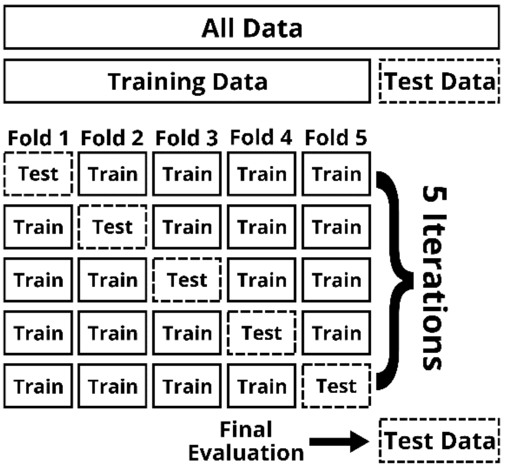


Figure 2.3 - Standard cross-validation processing. The available data is split into a train and a test set. The train set is split into *n* sets and a model is trained in *n* iterations, using a different set in each iteration for its evaluation. After the *n* iterations, the model is evaluated with the original, unseen, test set.

2.5.5 Hyperparameter Tuning

Machine Learning models have two types of parameters that influence the model performance: the ones that are learned from the training data (as in the case of the logistic regression's coefficients), and the ones that can be finely tuned by the user, called the hyperparameters.

Hyperparameters cannot be estimated from the data as their values are set before the learning process. These parameters have a direct impact on the model's performance but it is not possible to know the best value for each hyperparameter, beforehand, so these need to be optimized.

The most popular method for hyperparameter tunning is grid search. This method consists of an exhaustive search for the best combination of hyperparameter scores. In order to apply it, it requires a list of the hyperparameters and the values to be tested. This method could be time-consuming as a different model is trained for each value combination of each hyperparameter (if we want to optimize two hyperparameters with 5 different values each, grid search will create 10 different models), and so, some compromises might be required, such as decreasing the number of hyperparameters to tune, or the range of values to test for each hyperparameter.

Random Search is another tuning method, that differs from grid search as instead of the set of values to test for each hyperparameter, a statistical distribution to randomly choose from is given along with the number of combinations to try.

Halving grid search is a technique based on grid search, that searches for the best parameter value combination using successive halving. In successive halving, the parameter combinations are all initially tested with a small number of resources, i.e., training samples. The best performing combinations survive this iteration and are again tested, but with a larger number of training samples. At the last iteration, only one parameter combination will have survived and will be chosen as the best-performing one. As all the possible combinations are only trained once with a small set of samples, this approach allows for a more time-effective parameter search than the grid search.

Cross-validation is often coupled with (halving) grid search techniques. This way, the returned best parameter combination is asserted to provide consistent results on unseen data and can later be fitted to the entire training data and then predict the outcomes of the test set.

2.5.5.1 Logistic Regression Hyperparameters

Logistic Regression has three main hyperparameters that can be tuned: penalty, solver, and C [63,64,66]. The penalty parameter defines the inclusion of a regularization term in the cost function of the logistic regression (equation 2.6), with the cost function of the logistic regression being defined by:

$$Cost = \sum_{i=1}^{N} (-y_i \ln f(x) - (1 - y_i) \ln\big(1 - f(x)\big)) + r(w) \tag{2.11}$$

There are four different penalty values to be considered: None, L_1, $L_{2,}$ and ElasticNet. L_1 regularization can be seen as a form of feature selection, as it produces a sparse matrix, with some features having a value of 0. On the other hand, L_2 regularization only forces the weights of each feature to be small, but never zero. ElasticNet method is an ensemble of L_1 and L_2 regularizations, having a ratio that influences the contribution of each regularization. None simply considers $r(w) = 0$.

The C parameter is directly linked to the penalty parameter as it represents the inverse of the regularization strength, with smaller C values leading to a stronger regularization.

The cost function described in equation 2.11 has no closed form solution, thus, its optimization usually depends on gradient descent algorithms. Python’s scikit learn library [66] presents five different algorithms for the optimization of the cost function, that can be controlled by the solver parameter. Of the five available solvers, “liblinear” for example, uses a coordinate descent algorithm, and “sag” uses a Stochastic Average Gradient descent algorithm. The solver selection not only must take into account their computational time, but also consider the possible regularization penalties, since not all solvers are compatible with the presented penalties.

2.6 Feature Selection

In the previous section, a basic notion of machine learning models and how they work was presented. One critical part of building a successful model lies in the quality of the given data. Not only can datasets present a high dimensionality, i.e., a high number of features, as some of these features might be highly correlated with each other or just not provide any relevant information for our model, so the model can benefit from their removal. The removal of these features could possibly be performed manually, but datasets can have hundreds of features with not all of them being correctly labeled so it can be very time-consuming.

Feature Selection algorithms try to remove features that do not provide useful additional information to the model. Basic feature selection algorithms simply remove the features with the lowest variance. However, there are some more complex algorithms, based on machine learning model information, taking advantage of the L_1 regularization (that as previously discussed weights some features as 0) or using the feature importance originated from a tree model [64,67].

2.6.1 OPLS-DA

OPLS-DA (Orthogonal Partial Least Squares Discriminant Analysis) [68,69] is a regression model based on PLS (Partial Least Squares), a latent variable regression module based on the covariance between the data and the labels. OPLS-DA projects the data into one component whose variation is maximally correlated with the data labels and into several other orthogonal components whose variation is uncorrelated with the data labels, making it easy to interpret.

Although OPLS-DA's main use is to work as a classification algorithm, it can also be used as a feature selection algorithm, as it scores the contribution of each feature for the separation of the labels with a Variable Importance in Projection (VIP) score [70]. Each feature is scored regarding its contribution to the variance explained by each component of the OPLS model, depending on the weight of the component itself (orthogonal components have lower weights). Higher VIP scores ($VIP \in [0; \infty[$) are associated with features (in our case process or disease modules) more important for the discrimination of the positive and negative labels of the target module. Usually, features with VIP scores above 1 are considered to be relevant for the model, however, this value can be tuned depending on the data [71].

2.7 Ontologies

Ontologies provide a standardized form to represent, characterize and analyze data through a set of formalized entities and rules that describe their relationship [72]. In biology, ontologies can be used to define a structured, precisely defined vocabulary that is able to describe cell processes, disease, phenotypes, and gene functions across multiple systems [73].

The Open Biomedical Ontology (OBO) is the most popular ontology language for the description of biomedical and biological terms. Despite being an abstract formulation, ontologies are easily represented as graphs, where nodes represent the entities and edges their relationships. Biological ontology graphs are simple hierarchies, with child nodes having increasing levels of detail. In OBO ontologies, terms are connected by relations like "*is_a*" or "*part_of*" that form assertions between the connected nodes, e.g., "cytokinesis part_of cell proliferation".

2.7.1 Gene Ontology

Gene Ontology (GO) is the most used ontology in biology, describing biological processes, molecular functions, and cell components in three independent hierarchies. The biological process ontologies comprise the larger processes, composed of multiple molecular activities. Cellular component ontologies encompass the locations relative to cellular structures in which a gene product performs a function. Finally, molecular function ontologies relate to molecular-level activities performed by gene products, describing activities such as catalysis and transport.

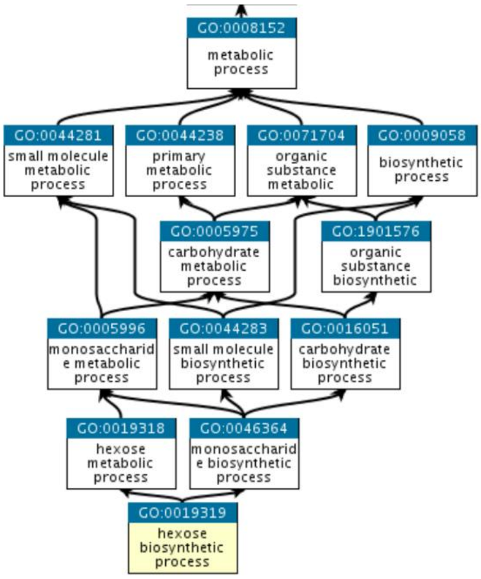

GO:0008152
metabolic process
GO:0044281
small molecule metabolic process
GO:0044238
primary metabolic process
GO:0071704
organic substance metabolic
GO:0009058
biosynthetic process
GO:0005975
carbohydrate metabolic process
GO:1901576
organic substance biosynthetic
GO:0005996
monosaccharide metabolic process
GO:0044283
small molecule biosynthetic process
GO:0016051
carbohydrate biosynthetic process
GO:0019318
hexose metabolic process
GO:0046364
monosaccharide biosynthetic process
GO:0019319
hexose biosynthetic process

Figure 2.4 - Gene Ontology example structure. Nodes are organized in a hierarchical structure, where each node represents a GO term. Nodes have one or more parent nodes (upstream connections) [73].

GO unifies the knowledge of gene products and gene functions for almost 1.5 million gene products, across 5 213 species, 185 of which with more than 1 000 annotations. As of July 2022, GO comprises more than 7 million annotations across the three different aspects. Overall, GO is composed of 43 558 terms, of which 28 140 are related to biological processes, 11 238 are related to molecular functions, and 4 180 are related to cellular components.

The characterization of cell processes and diseases can be performed with a Gene Ontology Enrichment Analysis (GOEA). This analysis allows the discovery of the GO terms over-represented in a particular gene set (for example genes associated with a particular disease) against a particular background. Over-represented terms can be interpreted as being correlated with the chosen set of genes, and thus allow for the interpretation of the biological relevance of such genes [74].

2.7.2 Human Phenotype Ontology

The Human Phenotype Ontology (HPO) [38] is an ontology that systematically defines phenotypes found in human diseases. HPO can be divided into five different sub ontologies: phenotypic abnormality, mode of inheritance, clinical modifier, clinical course, and frequency.

HPO provides annotations to diseases defined by the Online Mendelian Inheritance in Man (OMIM) database. Currently, HPO contains 108 580 annotations across 7 801 diseases, with the majority of the annotations being derived from the disease entry in OMIM.

HPO enables the association between diseases and phenotypes and between genes and phenotypes, thus allowing for the indirect comparison of shared phenotypes between genes and diseases.

Chapter 3

Materials and Methods

This chapter presents the methodology for the development of GAP-MINE, together with the methodology taken in all performance tests and comparisons.

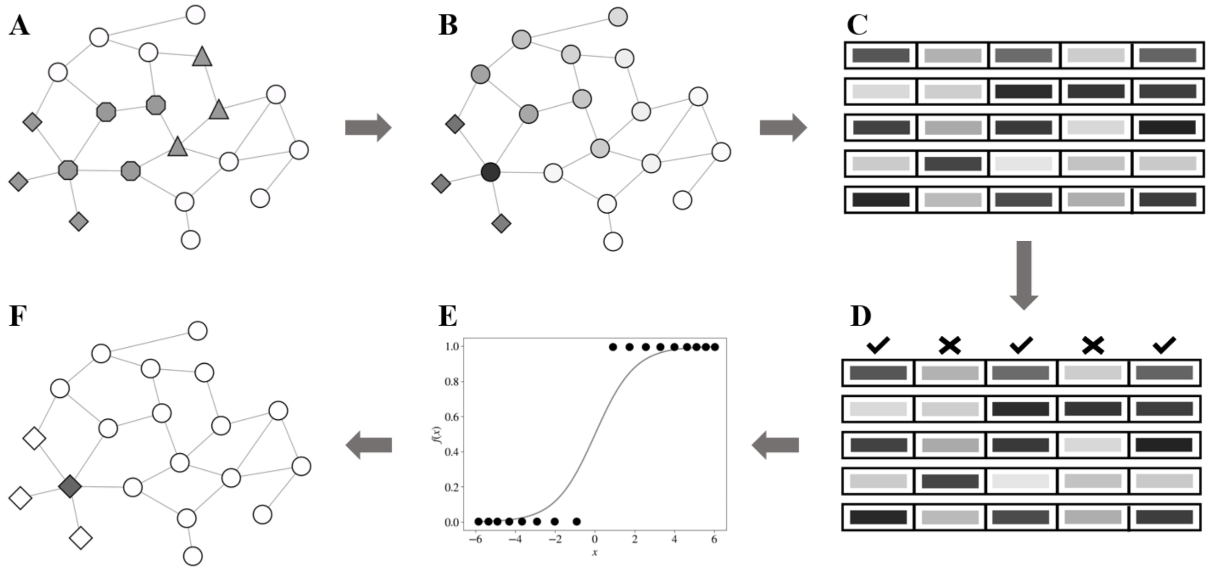


Figure 3.1 - Overview of GAP-MINE’s methodology: (A) Creation of the biological network with module-association annotations (diamond, octagon, and triangle nodes); (B) Scoring of each node regarding its proximity with the module nodes; (C) Creation of the embedding; (D) Feature Selection; (E) Logistic Regression Classifier Training; (F) Prediction and validation of new module-associated nodes. Figures A, B, and F were created with Cytoscape [60]. Figure E was adapted from [63].

GAP-MINE’s methodology can be explained in 6 different steps (Figure 3.1). First of all, the biological network is built using the information from protein-protein interactions and the gene-disease and gene-process associations. Each node is then scored regarding its proximity with the nodes associated with a target module, with this process being repeated for all modules. The node scores form a vector embedding representing its proximity across all modules. A feature selection algorithm is then applied to these embeddings, selecting the modules that best optimize the discrimination of the labels for a particular target module. The filtered embeddings are then used to train and test a logistic regression classifier. The classifier is then applied to the entire set of data, predicting new node-module associations, that later are validated. All steps of GAP-MINE’s methodology are described in detail across the remaining of this chapter.

Furthermore, alterations to GAP-MINE’s standard pipeline were performed to compare the algorithm performance against baseline and established methods and to evaluate its robustness to protein interaction network-related problems, such as network incompleteness and false positive annotations. All alterations to the standard GAP-MINE pipeline and the inclusion of additional methods are described in the later stages of this chapter.

3.1 Network Construction

3.1.1 Protein-Protein Interactions

Physical protein-protein interactions were obtained from two different data sources: APID and HuRI.

In APID's case, we selected interactions from the *Homo sapiens* interactome with quality level 1 (proven by 2 or more experimental pieces of evidence). The selected interactome comprises a total of 265 217 interactions. HuRI's HI-III-22 version was selected, containing 52 569 interactions and covering around 77% of a search space of more than 17 500 genes.

APID and HuRI use different protein identifier formats, causing a merging problem since the two databases have proteins in common and need to have the same identifier to properly merge the interactions and identify the proteins in common. UniProt IDs and Ensembl IDs are used to identify the proteins in APID and HuRI databases, respectively. Both ID types were translated into their HGNC Gene Name [75] synonyms. HGNC Gene Names provide good coverage results for the translation of UniProt and Ensembl IDs while presenting a single active identifier for the identification of each gene (the identifier might change but there are no multiple active identifiers to refer to the same gene at the same time). When combined, the resultant interactome has 284 177 interactions and 17 222 genes.

The igraph's package [76] was used to build the protein interaction network. The obtained interactome was simplified to remove self-loops, repeated edges, and to keep only the main component. The final network comprises a total of 17 204 genes connected by 260 960 interactions.

3.1.2 Gene-Process and Gene-Disease Associations

Gene-process associations were retrieved from Reactome, while gene-disease associations were retrieved from DisGeNET. Both Reactome and DisGeNET modules were filtered according to their number of genes, only keeping those that range from 50 to 300 genes, resulting in a total of 429 processes and 301 diseases. This range guarantees an adequate minimum number of positive cases for model training and avoids very large and heterogeneous modules.

Reactome and DisGeNET protein IDs were translated into their HGNC Gene Name in order to properly annotate the nodes in the created biological network.

3.1.2.1 Disease Modules

One factor that heavily impacts the prediction quality of network-based modules is their dispersion throughout the network. As nodes are scored regarding their proximity with module-associated nodes, disperse modules will have a broader distribution of values and module-associated nodes will not be able to achieve such high scores, increasing the probability of wrong nodes ranking amongst the higher-scored ones.

The Seed Connector algorithm (SCA) [77] performs a search throughout the network for linking nodes (seed connectors) able to connect seed module nodes into a single large component. SCA iteratively adds to the module the node that maximally increases the size of the current largest component. SCA stops when there are no nodes that simultaneously neighbor the largest component and one of the unconnected seed nodes. When having multiple neighboring nodes to choose from, SCA selects the one with the highest fraction of neighboring seed nodes compared to the total number of neighbors.

As SCA does not have any stop threshold for the number of added nodes, in our implementation of the algorithm we discarded disease modules with an increase of more than 40% of nodes, leaving us with 203 different diseases.

A second approach was taken where the SCA algorithm was again applied to the network, however, no nodes were added to the module, and only retaining the original module nodes included in the final largest component formed by SCA, thus retaining the 301 different diseases (Conservative approach) (Figure 3.2).

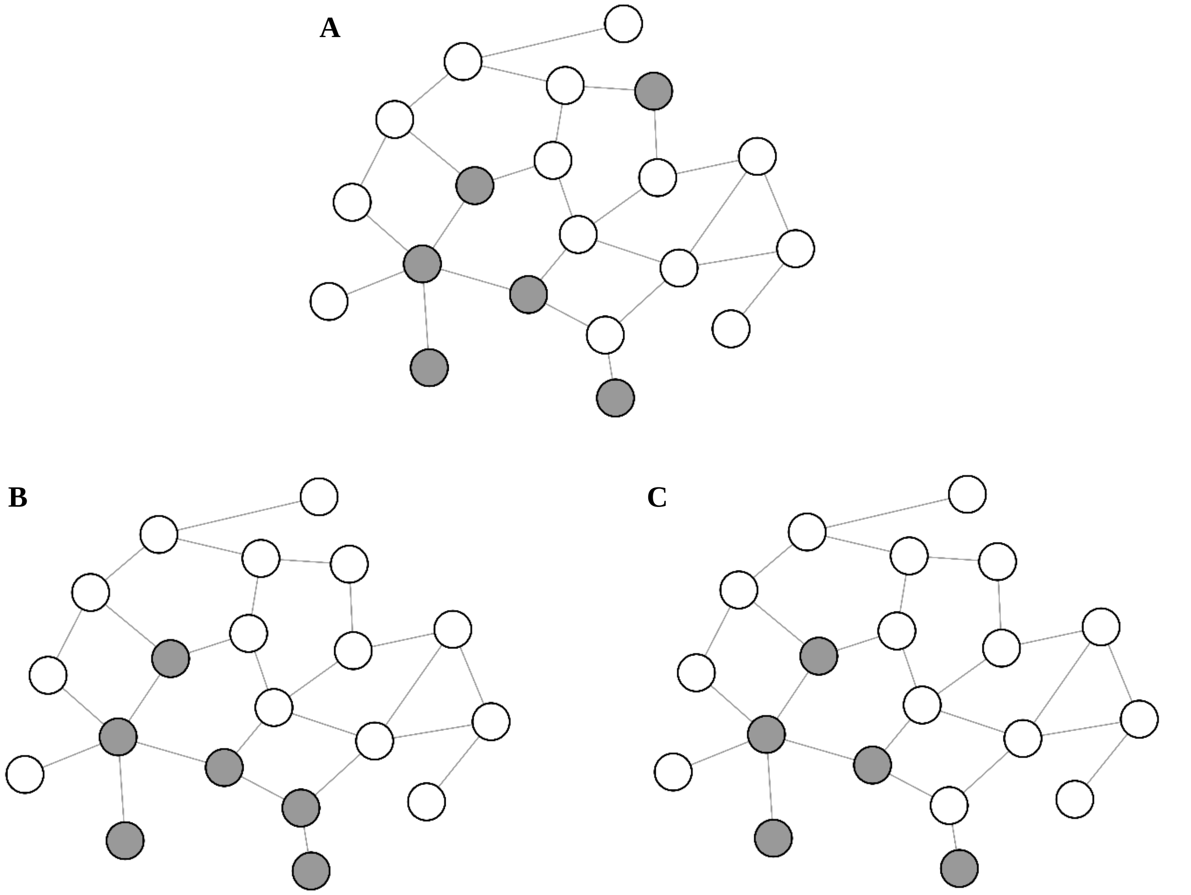


Figure 3.2 - Approaches taken for disease modules completion. The different networks represent how the same disease module is originally represented (A), and how SCA (B) and the Conservative approach (C) influence the nodes present in the disease module. Figures were created with Cytoscape [60].

3.1.3 Network Proximity metrics for gene-module association prediction

While standard proximity metrics score a node against all other nodes, a different approach should be taken when scoring nodes for the prediction of new gene associations with a specific process or disease module. In our case, the scoring of each individual node following the metrics described in chapter 2.3.1 would provide insights about how central a given node is or how it connects node pairs in the network. However, process and disease modules are spread across the entire graph, meaning that the centrality of a node in the entire graph will not provide any information as to how this node relates to the target module, making it impossible to apply the Guilt-by-Association principle. Furthermore, these metrics should be specific to each module since a gene associated with a disease or process is not necessarily associated with all other processes and diseases. In this work, five different module-specific proximity metrics were used to estimate each node's proximity to module nodes.

The Hypergeometric Test computes the probability of a given node being enriched with neighbors known to be associated with the target module. Equation 3.1 can be used to compute this metric for each candidate node in the network, where k is the number of neighbors of the candidate node associated with the target module, M is the total number of possible nodes to interact with

$(total\,number\,of\;genes - 1)$, n is the number of nodes known to be associated with the target module, and N is the number of neighbors of the candidate node.

$$P(k, M, n, N\,) = \frac{\binom{n}{k}\binom{M-n}{N-k}}{\binom{M}{N}} \tag{3.1}$$

Closeness can be altered to measure how a given node is related to a given module, instead of the entire network. To do so, equation 2.1 can still be used, however, y should only be a module-associated node (instead of any node) and consequently, N will be the number of nodes associated with the target module.

Betweenness can also be altered in order to follow the same rationale as the previous metrics, however, two different metrics can be implemented, Betweenness and Fraction Betweenness. Betweenness counts the fraction of module-associated node pairs that have the candidate node in their shortest paths. Equation 3.2 is used to compute this metric, where $\sigma_{yz}(x)$ is the number of module-associated node pairs that have the candidate node x in one or more of their shortest paths, and n is the number of module-associated nodes.

$$Betweenness(x) = \frac{\sum_{x \neq y \neq z} \sigma_{yz}(x)}{\frac{n(n-1)}{2}} \tag{3.2}$$

Fraction Betweenness follows a similar approach to Betweenness, however, here the number of different shortest paths that have the candidate node is accounted for, in order to give higher weights to nodes present in a higher fraction of the connections between the two module nodes. Fraction Betweenness can be computed by equation 2.2, where σ_{yz} is the number of shortest paths between nodes y and z, $\sigma_{yz}(x)$ is the number of shortest paths between y and z that pass through x, being y and z two nodes associated with the target module.

Finally, the Random Walks with Restart can be altered too, instead of starting from one specific node, start with equal probability from any node corresponding to a module-associated gene (Figure 3.3).

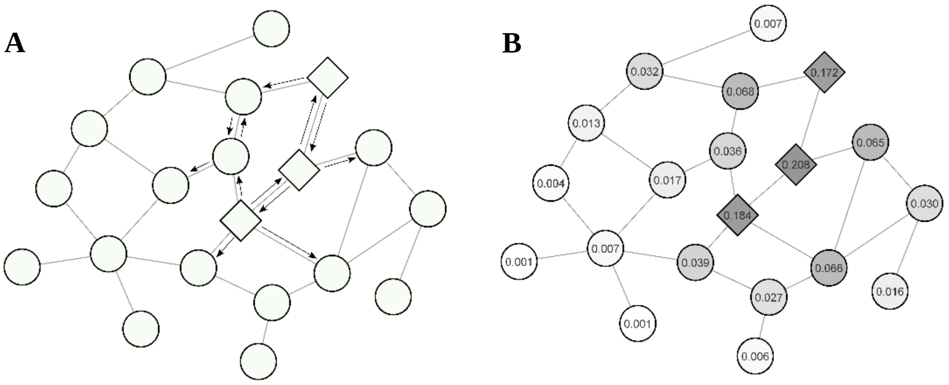


Figure 3.3 - RWR implementation for gene-module association prediction. Instead of restarting from a specific node, the RWR has a set of nodes from which the restart is allowed to restart, correspondent to the module nodes (diamond nodes) (A). The walk can thus start from any module-associated node and go to any other nodes, including those associated with the module. At the end of the iterations, a score is given based on the number of times each node was visited (B). Figures were created with Cytoscape [60].

3.1.4 Network Embedding Construction

The scoring of every node's proximity to the nodes associated with every module allows for the creation of a vector embedding that uniquely represents each node.

The vector embedding can thus be seen as a vector representing a node, where the j^{th} element is a network proximity metric explaining the proximity of that node with the j^{th} network module, composed by a set of nodes X associated with that module. The combination of all vector embeddings forms a matrix of N vs. M dimensions, where N is the number of nodes in the network and M is the number of disease or process modules used to build the embedding.

3.2 Feature Selection

The OPLS package was used to perform an Orthogonal Partial Least Squares Discriminant Analysis. OPLS-DA was individually applied to each module in order to build a model to discriminate that module's labels. In each model, the whole embedding matrix is used, with OPLS-DA scoring each module contribution for the discrimination of the target module's labels using a VIP score. There are several methods to define the threshold above which the VIP scores are considered relevant. Here three different approaches were taken: (1) selection of the 11 highest VIP scores (predicted module VIP + 10 modules with highest VIP scores), (2) selection of the upper outlier VIP scores (above $Q3 + 1.5xIQR$, where $Q3$ is the 3rd quartile and IQR is the interquartile range), and (3) selection of the n highest VIP scores where n is the mean between 11 and the number of upper outliers. In any of the three cases, a new embedding that explains the proximity of a given node across the selected modules is created and can later be analyzed to better understand how each module affects the classification.

3.3 Logistic Regression Classifier

The logistic regression uses the network embedding matrix as an input for a classification algorithm aiming to predict proteins associated with a particular module. Modules in the input matrix are selected through OPLS-DA's VIP score analysis, considering the three feature selection options described in chapter 3.2. Data is split into a stratified 80/20 partition, the same as the data partition used when applying the OPLS-DA model to avoid data leakage and overfitting.

The logistic regression models were optimized using a Halving Grid Search together with a 10-fold cross-validation using the following hyperparameters and set of values:

- Penalty: L_1, $L_{2,}$ or None
- C: [0.01, 0.1, 1, 10, 100]
- Solver: Liblinear, Sag, Saga or Newton-cg
- Maximum iterations: [10, 50, 100].

For each module, three different logistic regression models are trained and optimized with halving grid search, using one of the three different feature selection methods. The model with the best performance across the 10 different validation sets used is retained. The optimized logistic regression model is refitted to the used data and tested with the final 20% of (unseen) data. F-Measure is used to compare and evaluate the different models' performance, both upon training and after testing. Precision and Recall scores are also used for a more detailed analysis of the performance of the classifiers.

Different training and testing tasks are performed for each distinct module, thus obtaining k different classifiers, where k is the number of modules to classify.

3.4 False Positive Analysis

3.4.1 Gene Annotations

To evaluate if predicted False Positives could in fact be associated with the target module, additional gene annotations were downloaded from Gene Ontology and the Human Phenotype Ontology databases.

When using GO annotations, Gene Names were translated into UniProt IDs to get each GO term associated with each protein. For the three Gene Ontology categories (Biological Processes, Cellular Component, and Molecular Function) associations were found for a total of 15 314, 16 101, and 15 990 proteins, respectively. A GOEA was performed for each process and disease module by analyzing their known proteins and comparing them to all proteins present in the biological network. A confidence of 95% was used as the threshold for the identification of enriched terms.

The presence of enriched GO terms was searched for across the newly predicted proteins. Proteins with at least one GO term in common with that module's enriched set were considered to be a good prediction of the model. As only modules that have had at least one new prediction can be analyzed, only modules with at least one false positive were kept, resulting in a total of 218 processes and 101 and 218 diseases (SCA and Conservative Modules, respectively).

In the case of HPO annotations, DisGeNET disease IDs were translated into OMIM IDs to get the phenotypes associated with each disease. Due to ID incompatibilities, only 29 diseases from the SCA modules and 39 diseases from the Conservative modules were kept. As HPO is composed of a set of gene-phenotype and disease-phenotype associations, newly predicted genes were searched for having at least one phenotype in common with the disease of the target module.

In both approaches, a coverage metric that represents the fraction of predicted proteins with a GO term/HPO phenotype in common with the target module (equation 3.3) was computed. These coverages were compared with expected coverages when using a set of random proteins as new predictions, chosen from a set of proteins that excluded the proteins from the target module and the newly predicted proteins.

$$Coverage = \frac{\#\ New\ Proteins\ with\ enriched\ GO\ terms}{\#\ New\ Proteins} \tag{3.3}$$

3.4.2 Abstract Text Mining

Newly predicted module-associated nodes were searched to be present in titles and abstracts of published articles together with the name of the associated disease/process. PubMed's API [78] was used to search for titles/abstracts with both terms in common. The number of hits returned by PubMed's API was counted for the analysis of the results.

3.5 Performance Tests

3.5.1 Network Reductions

To simulate network incompleteness two different methods of network reductions were implemented.

The first one was performed by simply removing at random 20% of the protein-protein interactions (PPI 80%). A second one was performed by randomly removing 20% of the proteins and their interactions where each protein has a probability of being removed inversely proportional to their degree (Protein 80%).

3.5.2 Addition of Noise

To simulate annotation errors, 10% of wrongly annotated proteins were added to each process or disease module. The proteins were chosen at random, with each one having a probability for their selection given by equation 3.4:

$$p_{i,j} = \frac{log_{10}(degree_i)}{10^{sp_{i,j}}} \tag{3.4}$$

Where i identifies the protein and j the module, $degree_i$ is the degree of the queried protein and $sp_{i,j}$ is the smallest shortest path between the protein and any protein of module j. This promotes the selection of proteins with higher degrees that are closer in the network to other module proteins. Proteins with these properties are more likely to be associated with the module even in the absence of a real association.

3.6 Performance Comparisons

GAP-MINE performance was compared at two different levels. As GAP-MINE's method heavily depends on the quality of the proximity metric used, and as its main novelty is the inclusion of the information from different modules in an interpretable network embedding, it was first compared to a baseline model that solely depends on the metric quality to differentiate the labeled proteins.

In a later stage, GAP-MINE was compared with three well-established algorithms selected from the literature (Raw, GenePANDA, and MaxLink).

3.6.1 Baseline Classification

Previous network-based methods for gene annotation prediction, such as the ones mentioned in Table 1, have effectively been able to produce good results in the prediction of new disease-associated genes.

All these algorithms end up ranking proteins based on a single score produced by a combination of factors, such as their connectivity to known disease proteins, or the similarity between their associated diseases. In order to have a baseline prediction model with which to compare the results of our algorithm, we followed a similar approach by building a classifier that ranks each protein according to its proximity to known disease/process proteins.

The baseline classifier uses the scores that describe the proteins' proximity with the proteins from the target module. The same 80/20 stratified data split used in GAP-MINE is used here in the baseline model to ensure the direct comparison of the results, together with a 10-fold cross-validation. The baseline model uses training data to select the best metric value threshold above which nodes can be classified as associated with the module. On each fold, the set of selected proteins was ranked and a score threshold that maximizes the F-measure was selected, with its results being evaluated on the validation set. At the end of the cross-validation step, the average threshold score was applied to the test set. Thanks to this implementation, we can effectively test whether the use of a network embedding including the protein's network metric scores relative to other process/disease modules results in a more accurate and robust prediction algorithm.

3.6.2 Raw

Raw is a diffusion-based algorithm for node prioritization. Raw imposes a diffusion process on the interaction network to measure the influence between all pairs of nodes in an unbiased way, not discarding highly connected nodes or distant interactors. The Raw application can be seen as a constant flow running through the network during time *t*, with each of the module-associated nodes being the source of the flow. At $t \to \infty$, the flow values through each node are constant and can be used as node scores.

3.6.3 GenePANDA

GenePANDA is a network-based algorithm for the prioritization of candidate genes, that ranks nodes according to their distance to the known module nodes. GenePANDA's fundamental steps are the following:

1. Starting from the protein interaction network, compute an adjusted distance matrix. A distance network can be obtained by computing the length of the shortest path between two nodes (D_{ab}). The adjusted distance between a and b is given by equation 3.5:

$$D_{ab}^{adj} = \frac{D_{ab}}{\sqrt{\mu_a \times \mu_b}} \tag{3.5}$$

   where,

$$\mu_a = \frac{\sum_{j=1}^{N} D_{aj}}{N} \tag{3.6}$$

   Where $\sum_{j=1}^{N} D_{aj}$ is the sum of the length of the shortest paths between node a and every other node, and N is the total number of nodes in the network.

2. Given a list of module-associated nodes, compute a module-specific node weight, w_p. Candidate nodes should have stronger functional interactions with known module nodes than with random genes, thus w_p is defined by:

$$w_i = \frac{\sum_{j=1}^{N} D_{ij}^{adj}}{N} - \frac{\sum_{j=1}^{K} D_{ij}^{adj}}{K} \tag{3.7}$$

   Where N is the total number of nodes in the network, K is the total number of module nodes, and D_{ij}^{adj} is the adjusted distance between nodes i and j, with j being any network node in the first component of the equation, and any module node in the second component.

3. Weight conversion. To compare the scores between the different modules, weights are converted to probabilities. For every module, nodes are sorted by weight in descending order, and at each w_p the corresponding precision (equation 2.8) is computed, where TP and P are the total number of module nodes and the total number of nodes above w_p, respectively. The precision score is interpreted as the probability of a node with a weight above w_i being associated with the target module.

GenePANDA originally uses a STRING [79] network that includes the confidence of the interaction between two genes, which allows for the definition of raw distance between two neighbor nodes to be $D = 1000/S$, with S being the confidence on the interaction and ranging from 0 to 1 000. As our network does not present confidence values for the protein interactions, this step was simplified, and so neighboring nodes were considered to have the standard distance of 1.

3.6.4 MaxLink

MaxLink is a node prioritization algorithm that relies on the Guilt-by-association paradigm. MaxLink scores each node i based on the number of interactions with the module nodes. Additionally, to search for nodes enriched with interactions with the module, eliminating nodes with a high number of interactions to the module solely based on their high degree, a connectivity function (equation 3.8) is used.

$$connectivity(i) = \frac{\binom{K}{ML}\binom{N-K}{\deg(i)-ML}}{\binom{N}{\deg(i)}} \tag{3.8}$$

Where N is the number of nodes in the network, K is the number of module-associated nodes, ML is the number of links node i has with module-associated nodes, and deg (i) is the degree of node i.

MaxLink uses a connectivity filter of 0.5, discarding candidates with $connectivity(i) \geq 0.5$.

Chapter 4

Results

4.1 Random Walks with Restart is the best performing metric

To test if the GAP-MINE pipeline can improve the predictive performance of the Baseline classifiers, we compared both approaches using different types of network metrics to predict two types of annotations: biological processes (based on the Reactome Pathways) and disease associations (Figure 4.1). In all three testing scenarios (Process, Disease SCA, and Disease Conservative modules) we can observe that the RWR show the best performance, with F-Measure scores consistently above 0.9 on the test sets (Figure 4.1, Table S1), and showing similar results on the validation steps of the cross-validation procedure (Figure S1). Of the remaining four metrics, median F-Measure values never go above 0.5, with the Hypergeometric Test showing the second-best results in the Process and Conservative modules and the Fraction Betweenness having the second-best performance in the SCA modules. Closeness follows a similar behavior to both the Hypergeometric Test and the Fraction Betweenness. On the contrary, the Betweenness presents F-Measure values close to 0 in all three cases.

Comparing the performance of GAP-MINE with the baseline models, it is possible to observe that significantly better results are obtained in 2 out of the 3 modules in both the RWR and the Closeness metrics. However, significantly worse results are obtained in the betweenness-based metrics (Betweenness and Fraction Betweenness), but this does not significantly impact the evaluation of GAP-MINE as these are the worst-performing metrics.

Regarding the selection of modules for the embeddings, as stated in chapter 3.2, three different selection methods were tested with the highest performing one in the training set being kept for that module. Across the different modules, with the RWR, it is interesting to observe that process prediction tends to have a larger fraction of classifiers with only 11 modules (35%) than disease modules, where this selection type only comprises 20% of the classifiers (22% in Disease SCA and 20% in Disease Conservative). Regarding the other two types of feature selection, outlier selection is the preferred method, especially in the Disease Conservative Classification where it is selected in 59% of the classifiers (compared to 38% and 48% in Process and Disease SCA modules, respectively).

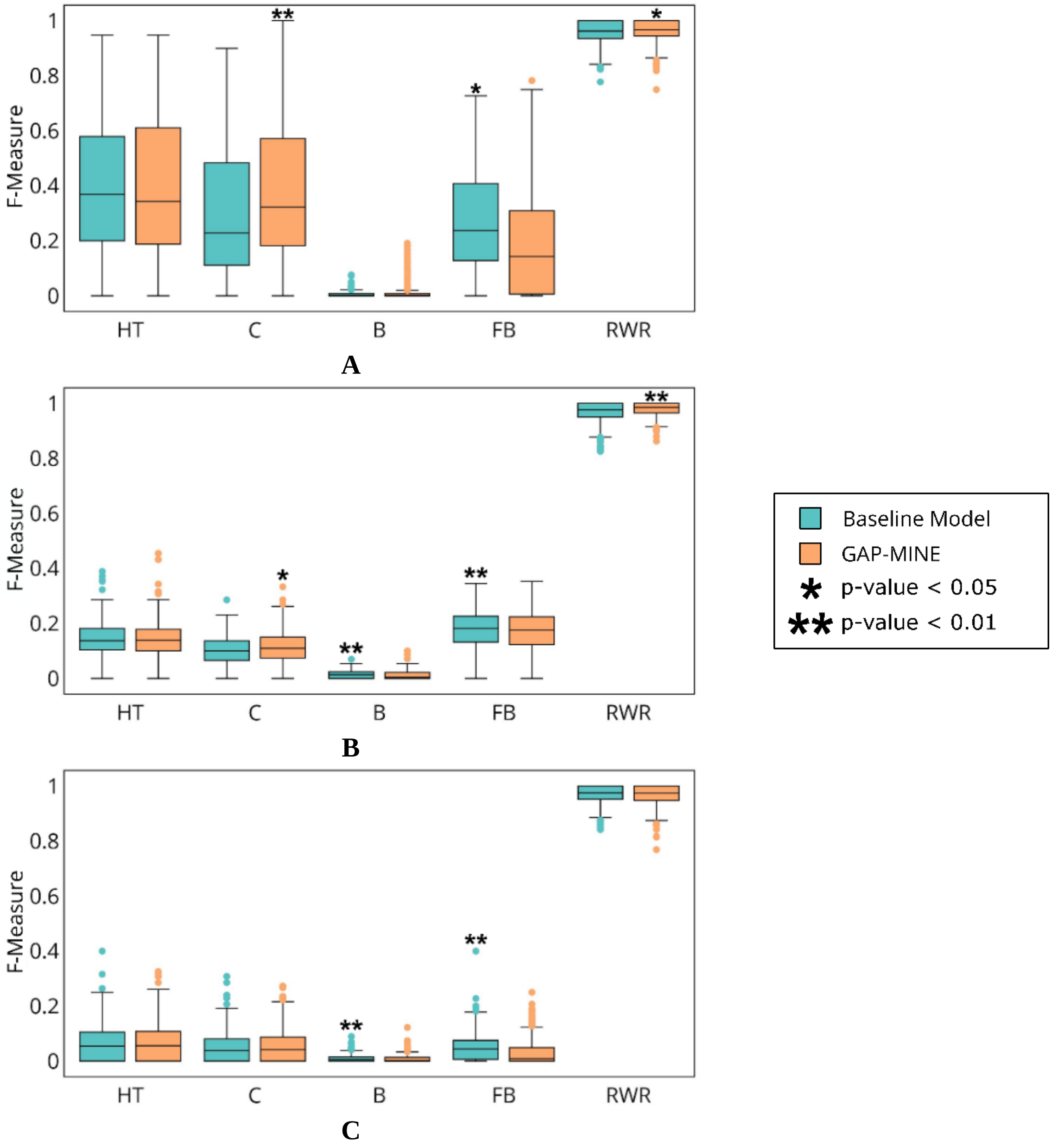


Figure 4.1 - GAP-MINE and Baseline performances show RWR as the best scoring metric. Both algorithms were applied to the complete biological network using the five different scoring metrics (Hypergeometric Test (HT), Closeness (C), Betweenness (B), Fraction Betweenness (FB) and Random Walks with Restart (RWR)) and the three different module types: (A) Cell Process; (B) Disease SCA; (C) Disease Conservative. Results show the algorithm's performance under a 20% testing set, after being trained in a 10-fold cross-validation. The same training and testing sets were used in both algorithms. Each boxplot represents the distribution of F-Measure values obtained for all the modules tested. * Represents statistical significance (* p<0.05, ** p<0.01) of the difference between Baseline and GAP-MINE distributions, evaluated by a paired Wilcoxon bilateral test. The * or ** are above the boxplot corresponding to the algorithm with the higher median. Figure created with Plotly [80].

4.2 GAP-MINE has higher precision but lower recall in comparison with the Baseline models

RWR has proven to be the best scoring metric with both GAP-MINE and the Baseline models, with F-Measure scores consistently above 0.9. Therefore, we used RWR to make a more detailed comparison of the Baseline and GAP-MINE performances. To evaluate thoroughly both methods, we used not only the F-Measure but also precision and recall. With the complete network (Figure 4.2), GAP-MINE obtains a significantly better F-Measure in both Process and Disease SCA modules. In all three module types, the Baseline models produce significantly better recall scores. On the other hand, GAP-MINE is always more precise. Both precision and recall metrics follow the same behavior as the remaining metrics (Table S1). The Baseline models have higher recall but lower precision, implying that they classify more candidate nodes as associated with the module, but with less certainty that they are true positives. Considering that these annotation predictions should be experimentally validated, we find it more advantageous to have the GAP-MINE performance, with higher precision and lower recall. The total number of annotation predictions is lower, but there is higher confidence that the predictions are true positives.

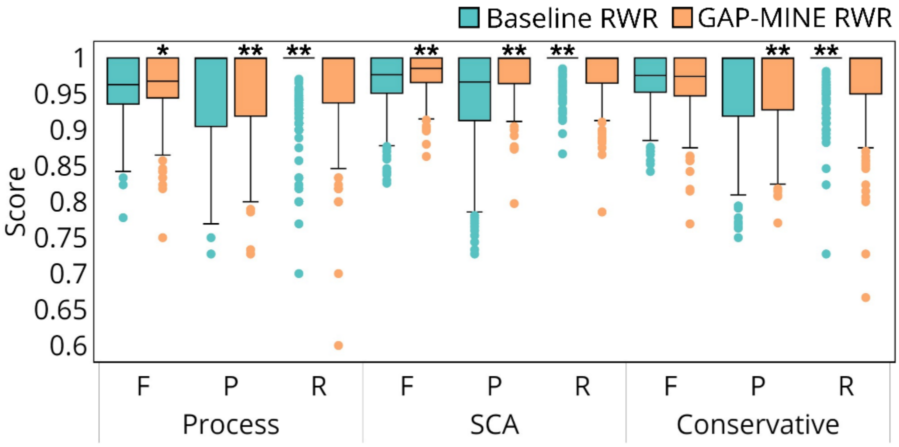


Figure 4.2 - GAP-MINE produces significantly better F-Measure scores (F) thanks to a significantly better precision (P), at the expense of a lower recall (R), in the three different modules (Cell Process, Disease SCA, and Disease Conservative). Boxplots represent the distributions of F, P, or R values for all the evaluated modules. * and ** represent the statistical significances ($p<0.05$ and $p<0.01$, respectively) of the difference between Baseline and GAP-MINE distributions, evaluated with a paired Wilcoxon bilateral test. The * or ** are above the boxplot corresponding to the algorithm with the higher median. Figure created with Plotly [80].

4.3 GAP-MINE outperforms the Baseline models in the presence of false annotations

The previously shown performance evaluations intrinsically assume that the network is complete and that all protein annotations are correct. However, as we previously mentioned, we cannot assume that our network is complete and does not have any wrong information. The mapping of the human interactome is a project that has been developed in the past 20 years [40], with the interactome growing from around 3 000 interactions to more than 650 thousand interactions [22,23]. A recent study analyzed the interactome of *Saccharomyces cerevisiae* under different environmental conditions and found a three-fold increase in the number of known interactions, as a considerable fraction of the interactions were found only under specific conditions [21]. Therefore, it is important to verify if the observations made with the complete network are still valid in conditions that mimic network incompleteness or the inclusion of false protein annotations.

Figure 4.3A and Table S2 show that, when using Protein 80% incomplete networks (lacking 20% of proteins and their interactions), both GAP-MINE and Baseline models have lower performances when applied to both types of disease modules. This is mainly due to a smaller number of positives being captured, as is shown by the decrease in recall scores. The fact that this decrease is only observed in the disease modules is probably related to the characteristics of these modules. Their smaller connectivity makes them more sensitive to the removal of interactions. Comparing GAP-MINE and the Baseline models' performances, it is possible to observe that both algorithms present similar results and thus are equally robust to the lack of interactions and proteins in the network. Similar results are obtained when using networks that had a 20% random removal of edges (Table S3).

The module annotations also play a relevant role in the quality of our predictions. Wrong annotations will have an impact on both the scoring function and the classification step. Both algorithms were tested under the addition of 10% false annotations. Figure 4.3B shows the biggest observed differences between GAP-MINE and the Baseline models. As previously stated, the Baseline models lack precision when compared to GAP-MINE. This is even clearer with the addition of noise as the Baseline models suffer a performance loss of 10-15%, whereas GAP-MINE only loses about 5% of its precision (Table S4). This shows that the added module proteins are not being classified as negatives in the baseline models, while GAP-MINE is able to filter a significant part of them (as the decrease in precision is smaller than 10%). Regarding the number of captured positives, we observe that the Baseline models are still able to capture a larger fraction of positives, however, the decrease in precision is so considerable that GAP-MINE obtains significantly better F-measure scores in all three module types.

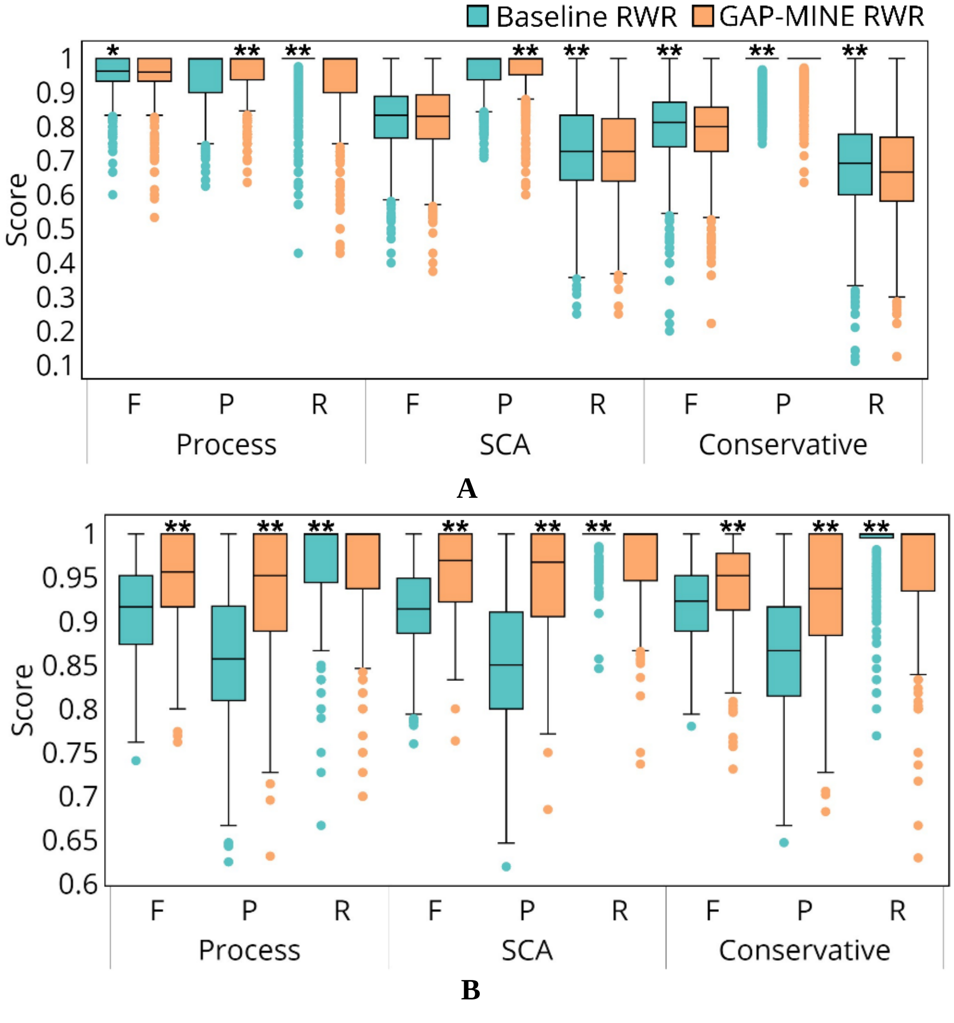
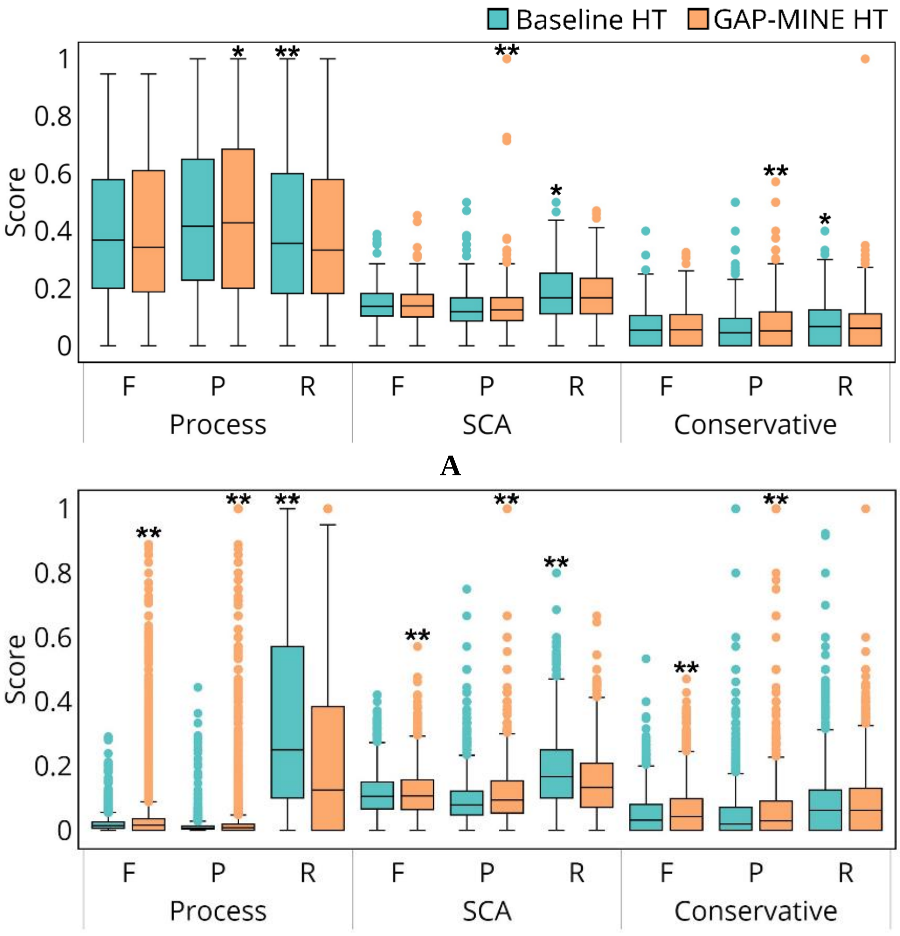


Figure 4.3 - GAP-MINE outperforms Baseline models in the presence of false annotations while presenting similar robustness to network incompleteness. Both algorithms with the RWR were tested in two different network conditions: (A) random removal of 20% of the proteins (with removal probability proportional to the protein degree); and (B) random addition of 10% of wrong annotations based on protein proximity to module (see methods). Both classifiers in (B) are trained considering the wrong information as true, and the performance scores are then corrected given the known truth. Boxplots represent the distributions of either F-Measure (F), Precision (P), or Recall (R) values for all the evaluated modules (Process, disease SCA and disease Conservative). * and ** represent the statistical significances ($p<0.05$ and $p<0.01$, respectively) of the difference between Baseline and GAP-MINE distributions, evaluated with a paired Wilcoxon bilateral test. The * or ** are above the boxplot corresponding to the algorithm with the higher median. Figure created with Plotly [80].

4.4 The Hypergeometric Test suffers a performance loss upon the removal of interactions

The Hypergeometric Test scores nodes according to their number of interactions with module nodes, and so, it is interesting to observe how the module connectivity impacts this metric performance on both the Baseline models and GAP-MINE.

As already stated, disease modules are much less connected than process modules. A poorly connected module should impact the performance of the Hypergeometric Test as it is more difficult for a node to be connected to another node from that module. Figure 4.4A confirms this hypothesis, as the Hypergeometric Test produces significantly better results when classifying process annotations. Furthermore, within the disease modules, we also see a decrease in quality when applying this metric to the Conservative modules, which goes in line with our hypothesis, since these modules do not have the additional nodes used to connect them, as is the case of the SCA modules.



A

B

Figure 4.4 - Module connectivity impacts the Hypergeometric Test performance. (A) The Hypergeometric Test in the complete network presents a performance loss when applied to disease modules, due to their poor connectivity. (B) When applying the Hypergeometric Test to the Protein 80% incomplete network, the process classification suffers a process loss, that is more dampened in the disease classification. Boxplots represent the distributions of either F-Measure (F), Precision (P), or Recall (R) values for all the evaluated modules (Process, disease SCA and disease Conservative). * and ** represent the statistical significances ($p<0.05$ and $p<0.01$, respectively) of the difference between Baseline and GAP-MINE distributions, evaluated with a paired Wilcoxon bilateral test. The * or ** are above the boxplot corresponding to the algorithm with the higher median. Figure created with Plotly [80].

Another way to decrease the connectivity of a module is to remove interactions. Contrary to what was previously observed, both types of disease modules show better performances than the process modules (Figure 4.4B). The performance loss in the process modules is in fact quite staggering but it is consistent with our hypothesis. Process modules are very well connected, which benefits the hypergeometric test as module proteins will be enriched with connections between themselves. However, when we remove 20% of the interactions, it is likely that we are also removing the specific interactions that allowed for that enrichment and so, it becomes easier for any node to have similar scores to module nodes. This marked performance loss with incomplete networks does not occur for disease modules as their smaller connectivity is not as severely impacted.

4.5 The Betweenness metric achieves better performances with the Baseline models using incomplete networks

The Betweenness metric was observed to be the worst metric when predicting annotations not only when using the complete network (Figure 4.5A), but also under several different performance tests (Table S1 and Table S4). However, the Betweenness metric shows significantly better results when incomplete networks are used (Figure 4.5B, Table S2, and Table S3). Furthermore, this increase in performance only happens when using the Baseline models. Betweenness is a metric that scores a given node if it is part of the shortest paths that connect two nodes of a given module. In this implementation of Betweenness, it only accounts for whether a node is present or not in the shortest paths, and not the fraction of shortest paths they are part of. The fact that we have this binary version of Betweenness makes it easy for a given uncorrelated node to have a similar score to the module nodes in a highly connected network. By removing 20% of the interactions, it is expected that the interactions with noisy nodes might disappear and thus Betweenness becomes more specific and has a performance gain.

The reason why this performance improvement only happens with the Baseline models might be explained if we consider the meaning of Betweenness. By scoring the nodes that are in the shortest paths of two module nodes, we are effectively scoring nodes that are within the target module, with any node that does not connect any two module nodes having a score of 0. Therefore, the scores obtained will only serve to identify whether a node is inside the module. Nodes outside the module will not be scored, independently of them being neighbors of module nodes, or being distant in the network. The Betweenness/GAP-MINE approach tries to improve the prediction for a given module by adding as predictor variables the betweenness of the candidates relative to other modules. However, the betweenness of other modules will only be informative if these modules are either highly overlapping or completely distinct. In the former case, little or no information is added when compared with the betweenness of the target module. In the latter case, these modules will only be informative for a small fraction of the candidates.

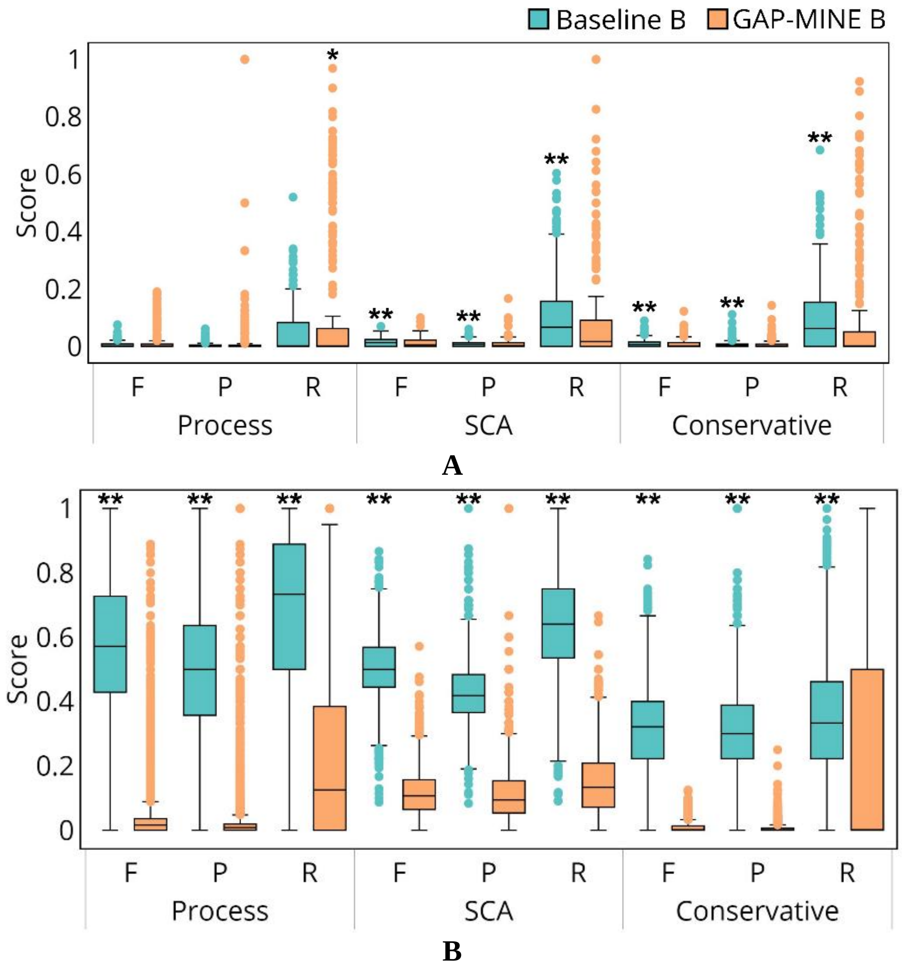


Figure 4.5 - The performance of the Betweenness metric improves with the Baseline models when applied to incomplete networks. (A) Results were obtained when applying the Betweenness centrality to the complete network using both GAP-MINE and Baseline models. (B) Using incomplete networks (Protein 80%), the Betweenness metric has significantly better results, but only when applied with the Baseline models. Boxplots represent the distributions of either F-Measure (F), Precision (P), or Recall (R) values for all the evaluated modules (Process, disease SCA, disease Conservative). * and ** represent the statistical significances ($p<0.05$ and $p<0.01$, respectively) of the difference between Baseline and GAP-MINE distributions, evaluated with a paired Wilcoxon bilateral test. The * or ** are above the boxplot corresponding to the algorithm with the higher median. Figure created with Plotly [80].

4.6 GAP-MINE outperforms established literature methods

Previous results showed that GAP-MINE outperforms the baseline methods solely based on the metrics used to describe the relationship that each node has with module nodes. To further evaluate the performance of GAP-MINE, three established literature methods (Raw, GenePANDA, and MaxLink) were applied to our network for the prediction of gene-disease and gene-process associations.

GAP-MINE with the RWR is shown to have significantly better performassnce results than all three algorithms in all module types (Figure 4.6). Of the three additional algorithms, Raw is the one showing more similar results to RWR, which is not a striking behavior as both algorithms are diffusion-based methods. GenePANDA and MaxLink produce similar results to the ones observed when Closeness and the Hypergeometric Test metrics were applied to the baseline, which goes in line with what was observed with the Raw algorithm, and with both producing worse results than the remaining two metrics.

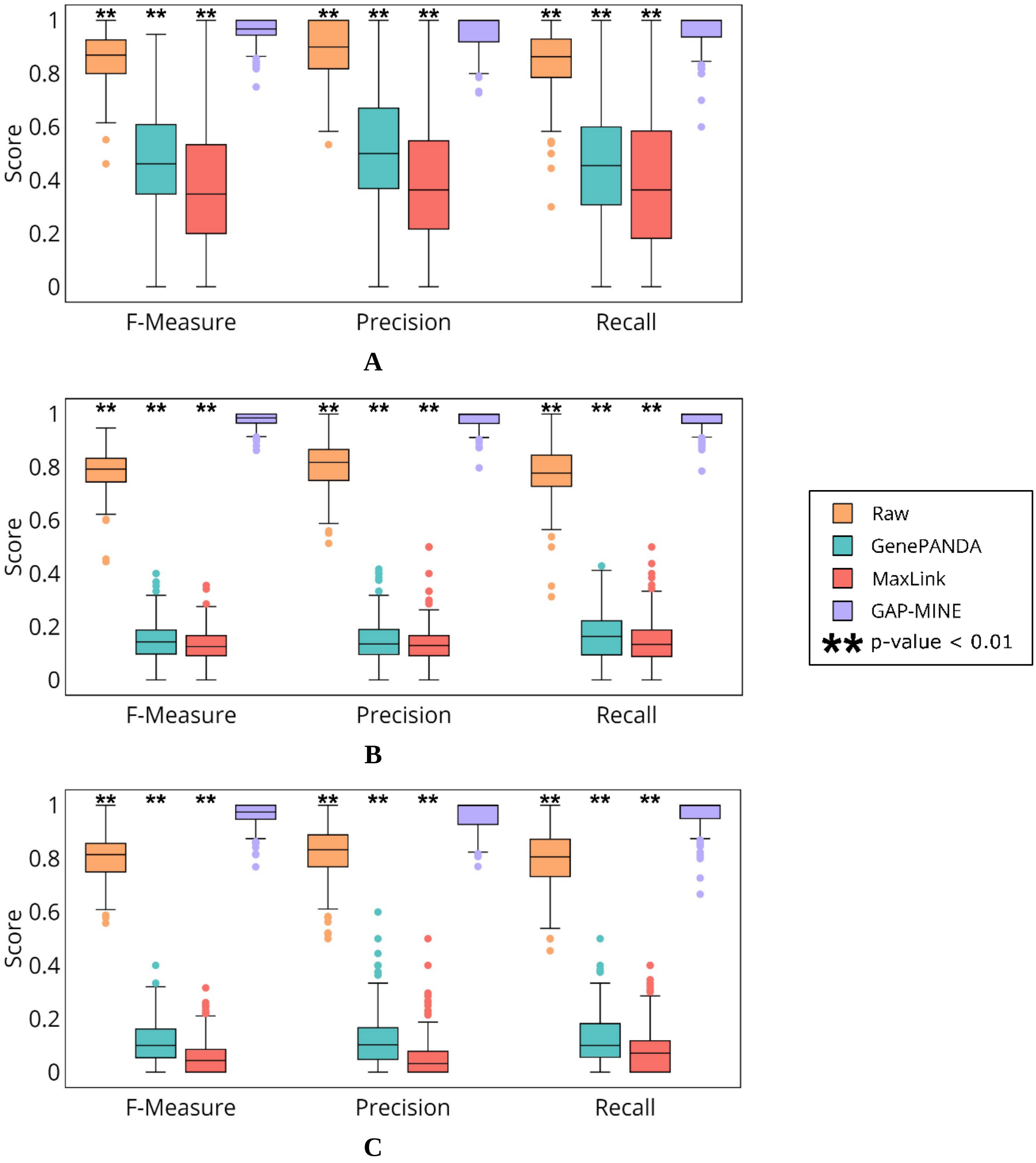


Figure 4.6 - GAP-MINE shows significantly better results than established literature methods. All four algorithms were applied to the complete biological network and applied to the three different module types: (A) Cell Process; (B) Disease SCA; (C) Disease Conservative. Results show the algorithm's performance under a 20% testing set, after being trained in a 10-fold cross-validation. The same training and testing sets were used in both algorithms. Boxplots represent the distributions of F-Measure, Precision, or Recall values for all the evaluated modules. ** Represents a statistical significance of 99% of GAP-MINE scores being significantly higher than the compared algorithm, evaluated by a paired Wilcoxon bilateral test. Figure created with Plotly [80].

4.7 GAP-MINE module scores provide Biological Explainability for the predictions

The selection of a certain number of processes or diseases to aid in the prediction of a given process or disease module not only contributes to the algorithm's robustness and precision but can also provide interesting insights from a biological point of view, as possible patterns can be inferred from the chosen modules. Rheumatoid Arthritis (RA), or C0003873 is a DisGeNet disease composed of 228 proteins in the SCA modules. RA is an autoimmune disease characterized by synovial inflammation, swelling, cancer autoantibody production, and cartilage and bone destruction. Furthermore, as an autoimmune disease, individuals also tend to present a higher mortality rate than healthy individuals, especially due to cardiovascular diseases, and infections, such as pneumonia [81].

The RA classification model was built with 11 different disease modules, one of which was the RA module. By analyzing the selected modules, we can find several groups of diseases that differ from each other but that are related to RA's different phenotypes. One notable group is Atherogenesis, Atherosclerosis, and Inflammation. Not only Atherosclerosis is more prevalent in RA patients, but both diseases also share genetic and environmental risk factors. Furthermore, Inflammation is closely linked to Atherosclerosis in RA patients, as patients with both RA and Atherosclerosis show aggravated signs of inflammation [82].

Table 4.1 - Disease Modules selected for the logistic regression model that predicts new associations to Rheumatoid Arthritis.

Disease ID	Disease Name
C0003873	Rheumatoid Arthritis
C0020517	Allergy
C1527304	Allergic Reaction
C0021368	Inflammation
C0032285	Pneumonia
C0887898	Experimental Lung Inflammation
C3714636	Pneumonitis
C0032300	Lobar Pneumonia
C1563937	Atherogenesis
C0004153	Atherosclerosis
C0345967	Malignant Mesothelioma

Disease-associated genes are not necessarily causal genes as they can have other functions that are disrupted upon disease. Thus, it is expected that these genes show different behaviors throughout the embedding. A Principal Component Analysis was applied to the set of known and predicted RA proteins to find how the proteins group. RA proteins show two main clusters, with a fraction of proteins being dispersed along the first component (Figure 4.7A). More importantly, the predicted proteins cluster along the known proteins thus showing similar patterns to these.

The selected modules can also provide insights on their importance to discriminate the candidate proteins. Figure 4.7B presents a clustergram of the candidate proteins together with their RWR scores for the chosen modules. We can observe three different disease clusters and three different protein clusters. Regarding the disease clusters, it is worth pointing out that the lung-associated diseases cluster together, as well as the two allergy disease modules. As for the predicted proteins, MEOX2, COPS5, and HTT are proteins with especially high scores under the lung-related disease modules. These results suggest that, besides making new disease association predictions, GAP-MINE results can discriminate between different subtypes of associations.

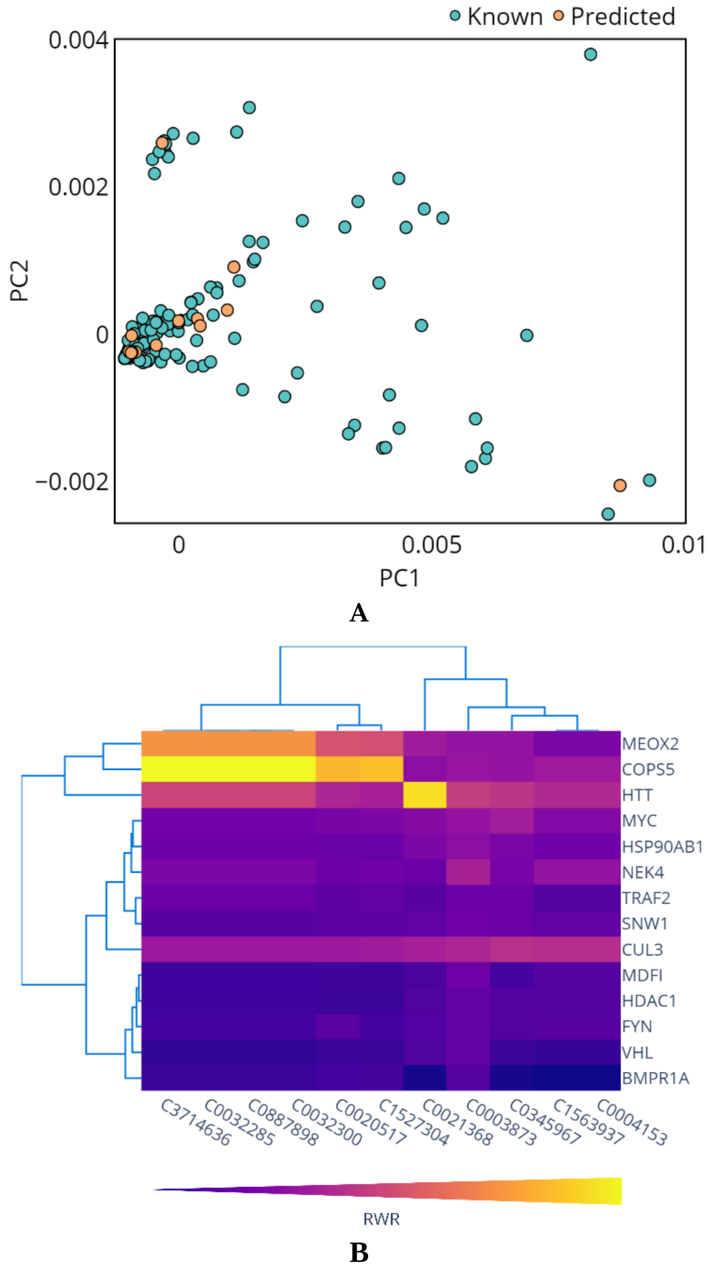


Figure 4.7 - Module proteins present different scoring behaviors. (A) Principal Component Analysis using the known Rheumatoid Arthritis proteins and the newly predicted one's data across the 11 selected modules. 87.9% of the variance in the data is explained by the first two principal components. (B) Clustergram analysis with the 14 new predicted proteins of Rheumatoid Arthritis and the 11 selected modules to see the importance of the protein relationship with each of the chosen modules. Figure created with Plotly [80].

4.8 Novel predictions are significantly associated with concordant GO and HPO annotations

Our pipeline has shown promising results when applied with the RWR, with precision scores consistently above 0.9. This precision was evaluated with known annotations, but the utility of GAP-MINE depends on its ability to make novel predictions. To evaluate these novel predictions, we applied GAP-MINE with RWR to each of the different modules using the full extent of the data (without train/test/validation splits). To check whether our new predictions are biologically plausible, we applied two distinct approaches using GO terms or HPO phenotypes. We performed a GOEA for each process and disease module using their known proteins. For every newly predicted protein, we checked whether it was annotated with at least one of the enriched GO terms. In the HPO phenotypes approach, for each new prediction, we checked for annotations with at least one phenotype in common with known disease phenotypes. In both approaches, we computed a coverage metric that represents the fraction of predicted proteins with a GO term/HPO phenotype in common with the target module. These coverages were then compared with expected coverages when using a set of randomly chosen proteins as new predictions.

For each set of modules, a probability density function of coverage values was estimated for both the real predictions and the random ones. To find whether the real predictions were skewed to higher coverage values, density values were compared through their log ratio (Figure 4.8).

As we can observe, in both the GO and HPO analysis, we get positive fold values for coverage values above 0.8 and 0.7, respectively. This shows that our predictions are enriched with relevant annotations when compared to a random classifier, and thus, are biologically relevant.

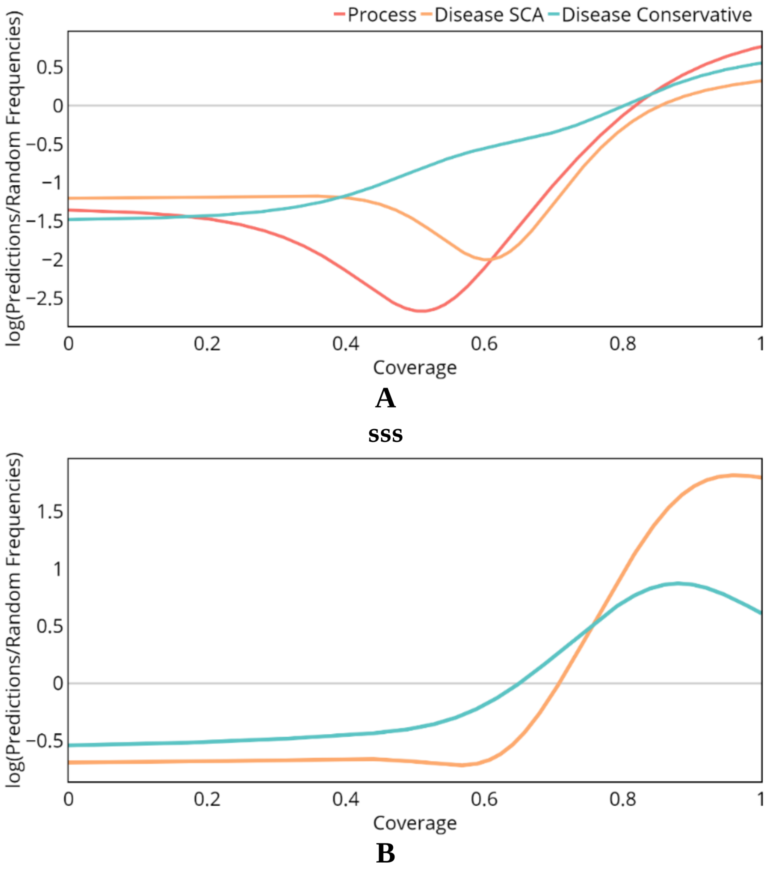


Figure 4.8 - GAP-MINE predictions are enriched with biologically relevant GO terms and HPO phenotypes when compared to a random classifier. (a) GO enrichment analysis was performed for the different process and disease modules. The predicted proteins GO terms were compared to the process/disease enriched GO terms. (b) HPO database provides gene-phenotype and disease-phenotype associations. New predictions were searched for the presence of phenotypes in common with the target disease. In (a) and (b) coverage values were computed as the fraction of new predictions with terms/phenotypes in common with the target module. Plotted curves represent the log ratio of the coverage frequency distributions of Predictions over random selection. A positive log ratio means that the corresponding coverage value is more frequent in real predictions compared with the random selection of predictions. Figure created with Plotly [80].

The discovery of gene associations with diseases and cell processes is an ongoing task, and so, potential new associations might be discovered every day. This reality differs from the databases that record these associations which have only periodic updates (e.g., DisGeNET's last update dates from May 2020). Therefore, new associations found by our algorithm might in fact have already been documented in the literature. As seen in chapter 4.7, 14 new genes were predicted to be associated with RA disease. To check whether these genes were already described to be associated with RA, we searched for

Titles/Abstracts with the presence of "Rheumatoid Arthritis" together with each one of the predicted genes. As Figure 4.9 shows, only 5 of the 14 predicted genes were not found in any paper title/abstract. Of the remaining 11 genes, all of them have at least 2 different papers that relate them to the predicted disease, which validate the predictions of our algorithm.

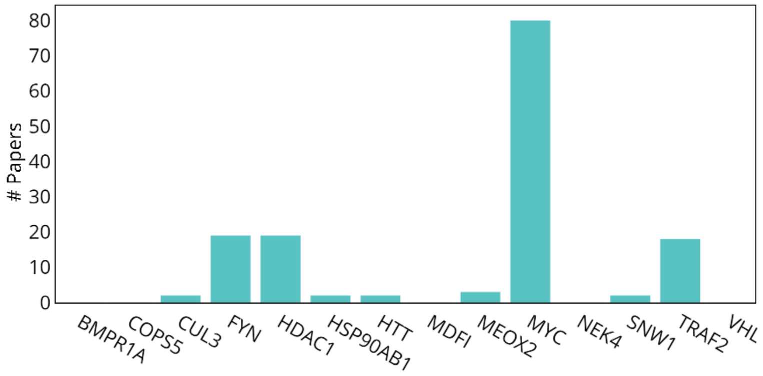


Figure 4.9 - GAP-MINE predictions for Rheumatoid Arthritis are already documented in the literature. PubMed's API was used to perform a literature search of titles and abstracts that had the predicted gene and "Rheumatoid Arthritis" in the text. Figure created with Plotly [80].

Chapter 5

Discussion

GAP-MINE proposes a flexible and modular framework to improve the predictions of network-based gene prioritization and classification methods. Most of these methods use a list of proteins with a given annotation as input and compute a score for all proteins in the interaction network. The higher the score, the more likely it is for the protein to be associated with the annotation of the input set. GAP-MINE framework can apply any of these methods to a large set of annotations, each defining a network module and a corresponding column in the embedding matrix. Then, a feature selection step selects the most informative network modules to predict a target annotation. Finally, a machine learning model is trained with the known proteins with the target annotation as positive examples. There is a large choice of algorithms available for the two last steps. The interaction network used to compute the individual module scores can also be adjusted, by using different sources of information or by including different types of interaction. In this work, we only used five implementations of GAP-MINE. All five use a protein physical interaction network, perform feature selection through OPLS-DA, and make predictions with a logistic regression. The five distinct implementations differ in the network-based method to compute module scores. One of our main aims was to test if the implementation of GAP-MINE with network-based method “A” could in fact improve the predictive performance compared to method “A” alone. Therefore, we selected five general network metrics that could represent the variety of network-based gene prioritization methods. It was not our goal to pursue the absolute optimization of GAP-MINE performance, as that would imply testing much more variations of its implementation.

Our results confirmed that the GAP-MINE approach can achieve higher performance, measured by the F-measure, than the simple Baseline models with the same network metric, although that improvement is not homogeneous across the five metrics tested. The two variants of Betweenness perform better alone than within the GAP-MINE framework. This means that the Betweenness scores for additional modules are not informative for the prediction of the target annotation. On the other hand, Closeness and RWR improve with the GAP-MINE framework. Both metrics measure network proximity between the candidates and the modules, which may contribute to the utility of the embedding approach.

In most cases, the improvements are related to an increase in precision, but at the expense of a lower recall. Even in some situations where the GAP-MINE F-Measure is not significantly higher, we observe increases in precision. The higher precision scores make GAP-MINE a suitable algorithm for candidate prediction: although a smaller number of candidates is generated at each classification task, the higher precision increases the likelihood of success during the experimental validation task.

The fact that the GAP-MINE framework uses additional information when compared with the Baseline models led us to expect larger performance increases in conditions mimicking network incompleteness or the existence of false annotations. We implemented two scenarios of network incompleteness, but in both, the performances of GAP-MINE were very similar to the corresponding Baseline models, and no clear pattern of significant differences between the two approaches was observed. Our expectations were confirmed when we introduced false annotations. Here, we observed larger performance increases with GAP-MINE, consistently supported by higher precisions accompanied by smaller recall decreases.

The RWR has the best performance under all the different tested conditions and module types, with the obtained candidates shown to be biologically relevant. This agrees with previous benchmark studies

[5,83]. Our results with GAP-MINE using RWR suggest that further optimization of GAP-MINE implementation with other RWR-based methods, feature selection, and machine learning algorithms could result in a state-of-art gene annotation prediction system. Such implementation could also be improved with the integration of network-independent data sources. This could be easily achieved by adding extra variables to the GAP-MINE embedding matrix before the machine learning step.

Lastly, one main advantage of the GAP-MINE framework is the biological interpretability of its results. Previous works have combined network embeddings with machine learning to predict gene annotations [84,85]. But the embeddings used, such as node2vec [86,87], do not allow easy extraction of biological insights. As the GAP-MINE embedding is composed of variables associated with network modules and the corresponding annotations, it is relatively easy to get such biological interpretations. Firstly, the modules chosen in the feature selection step can support the generation of hypotheses about common mechanisms between diseases. Alternatively, if one predicts disease associations using embeddings based on biological processes or pathways, the relevance of such pathways for disease development can be inferred. Secondly, the RWR scores for the different modules in the embedding can help to understand why some candidates are being predicted. Even the known proteins of the target annotation can have different profiles in the embedding matrix, leading to the decomposition of the network module into submodules. These submodules can give hints for different roles in the associated process or disease.

Besides its good performance results, especially robust to the presence of false annotations, GAP-MINE's strong points are its adaptability and its biological interpretability. Overall, GAP-MINE's algorithm provides a suitable approach for network-based gene annotation prediction.

Bibliography

1. Zahn-Zabal, M.; Michel, P.-A.; Gateau, A.; Nikitin, F.; Schaeffer, M.; Audot, E.; Gaudet, P.; Duek, P.D.; Teixeira, D.; de Laval, V.R.; et al. The NeXtProt Knowledgebase in 2020: Data, Tools and Usability Improvements. *Nucleic Acids Res* **2019**, *48*, D328–D334, doi:10.1093/NAR/GKZ995.

2. Ramos, E.M.; Hoffman, D.; Junkins, H.A.; Maglott, D.; Phan, L.; Sherry, S.T.; Feolo, M.; Hindorff, L.A. Phenotype-Genotype Integrator (PheGenI): Synthesizing Genome-Wide Association Study (GWAS) Data with Existing Genomic Resources. *European Journal of Human Genetics* **2014**, *22*, 144–147, doi:10.1038/ejhg.2013.96.

3. Venkatesan, K.; Rual, J.F.; Vazquez, A.; Stelzl, U.; Lemmens, I.; Hirozane-Kishikawa, T.; Hao, T.; Zenkner, M.; Xin, X.; Goh, K. il; et al. An Empirical Framework for Binary Interactome Mapping. *Nature Methods 2008 6:1* **2008**, *6*, 83–90, doi:10.1038/nmeth.1280.

4. Colhoun, H.M.; McKeigue, P.M.; Smith, G.D. Problems of Reporting Genetic Associations with Complex Outcomes. *Lancet* **2003**, *361*, 865–872, doi:10.1016/S0140-6736(03)12715-8.

5. Zhang, H.; Ferguson, A.; Robertson, G.; Jiang, M.; Zhang, T.; Sudlow, C.; Smith, K.; Rannikmae, K.; Wu, H. Benchmarking Network-Based Gene Prioritization Methods for Cerebral Small Vessel Disease. *Brief Bioinform* **2021**, *00*, 1–12, doi:10.1093/bib/bbab006.

6. del Sol, A.; Balling, R.; Hood, L.; Galas, D. Diseases as Network Perturbations. *Curr Opin Biotechnol* **2010**, *21*, 566–571, doi:10.1016/J.COPBIO.2010.07.010.

7. George, R.A.; Liu, J.Y.; Feng, L.L.; Bryson-Richardson, R.J.; Fatkin, D.; Wouters, M.A. Analysis of Protein Sequence and Interaction Data for Candidate Disease Gene Prediction. *Nucleic Acids Res* **2006**, *34*, doi:10.1093/nar/gkl707.

8. Yin, T.; Chen, S.; Wu, X.; Tian, W. GenePANDA - a Novel Network-Based Gene Prioritizing Tool for Complex Diseases. *Nature Publishing Group* **2017**, doi:10.1038/srep43258.

9. Guala, D.; Sjölund, E.; Sonnhammer, E.L.L. MaxLink: Network-Based Prioritization of Genes Tightly Linked to a Disease Seed Set. *Bioinformatics* **2014**, *30*, 2689–2690, doi:10.1093/bioinformatics/btu344.

10. Ghiassian, S.D.; Menche, J.; Barabási, A.L. A DIseAse MOdule Detection (DIAMOnD) Algorithm Derived from a Systematic Analysis of Connectivity Patterns of Disease Proteins in the Human Interactome. *PLoS Comput Biol* **2015**, *11*, 1–21, doi:10.1371/journal.pcbi.1004120.

11. Petti, M.; Bizzarri, D.; Verrienti, A.; Falcone, R.; Farina, L. Connectivity Significance for Disease Gene Prioritization in an Expanding Universe. *IEEE/ACM Transactions of Computational Biology and Bioinformatics* **2020**, *17*, 2155–2161, doi:10.1109/TCBB.2019.2938512.

12. Guney, E.; Oliva, B. Exploiting Protein-Protein Interaction Networks for Genome-Wide Disease-Gene Prioritization. *PLoS One* **2012**, *7*, 43557, doi:10.1371/journal.pone.0043557.

13. Garcia-Vaquero, M.L.; Gama-Carvalho, M.; De, J.; Rivas, L.; Pinto, F.R. Searching the Overlap between Network Modules with Specific Betweeness (S2B) and Its Application to Cross-Disease Analysis. *Sci Rep* **2018**, *8*, doi:10.1038/s41598-018-29990-7.

14. Maiorino, E.; Baek, S.H.; Guo, F.; Zhou, X.; Kothari, P.H.; Silverman, E.K.; Barabási, A.-L.; Weiss, S.T.; Raby, B.A.; Sharma, A. Discovering the Genes Mediating the Interactions between Chronic Respiratory Diseases in the Human Interactome. *Nat Commun* **2020**, *11*, doi:doi.org/10.1038/s41467-021-22939-x.

15. Vandin, F.; Upfal, E.; Raphael, B.J. Algorithms for Detecting Significantly Mutated Pathways in Cancer. *Journal of Computational Biology* **2011**, *18*, 507–522, doi:10.1089/cmb.2010.0265.

16. Köhler, S.; Bauer, S.; Horn, D.; Robinson, P.N. Walking the Interactome for Prioritization of Candidate Disease Genes. *Am J Hum Genet* **2008**, *82*, 949–958, doi:10.1016/J.AJHG.2008.02.013.

17. Vanunu, O.; Magger, O.; Ruppin, E.; Shlomi, T.; Sharan, R. Associating Genes and Protein Complexes with Disease via Network Propagation. *PLoS Comput Biol* **2010**, *6*, doi:10.1371/journal.pcbi.1000641.

18. Erten, S.; Bebek, G.; Koyutürk, M. Vavien: An Algorithm for Prioritizing Candidate Disease Genes Based on Topological Similarity of Proteins in Interaction Networks. *Journal of Computational Biology* **2011**, *18*, 1561–1574, doi:10.1089/cmb.2011.0154.

19. Le, D.H.; Kwon, Y.K. Neighbor-Favoring Weight Reinforcement to Improve Random Walk-Based Disease Gene Prioritization. *Comput Biol Chem* **2013**, *44*, 1–8, doi:10.1016/J.COMPBIOLCHEM.2013.01.001.

20. Cáceres, J.J.; Paccanaro, A. Disease Gene Prediction for Molecularly Uncharacterized Diseases. *PLoS Comput Biol* **2019**, *15*, 1–14, doi:10.1371/journal.pcbi.1007078.

21. Mordelet, F.; Vert, J.P. ProDiGe: Prioritization Of Disease Genes with Multitask Machine Learning from Positive and Unlabeled Examples. *BMC Bioinformatics* **2011**, *12*, 1–15, doi:10.1186/1471-2105-12-389.

22. Liu, Z.; Miller, D.; Li, F.; Liu, X.; Levy, S.F. A Large Accessory Protein Interactome Is Rewired across Environments. *Elife* **2020**, *9*, doi:10.7554/eLife.62365.

23. The International Human Genome Mapping Consortium A Physical Map of the Human Genome. *Nature* **2001**, *409*, 934–941, doi:10.1038/35057157.

24. Little, J.; Bradley, L.; Bray, M.S.; Clyne, M.; Dorman, J.; Ellsworth, D.L.; Hanson, J.; Khoury, M.; Lau, J.; O'Brien, T.R.; et al. Reporting, Appraising, and Integrating Data on Genotype Prevalence and Gene-Disease Associations. *Am J Epidemiol* **2002**, *156*, 300–310, doi:10.1093/aje/kwf054.

25. Barabási, A.L.; Gulbahce, N.; Loscalzo, J. Network Medicine: A Network-Based Approach to Human Disease. *Nat Rev Genet* **2011**, *12*, 56–68, doi:10.1038/nrg2918.

26. Risch, N.J. Searching for Genetic Determinants in the New Millennium. *Nature* **2000**, *405*, 847–856, doi:10.1038/35015718.

27. Hamosh, A.; Scott, A.F.; Amberger, J.S.; Bocchini, C.A.; Mckusick, V.A. Online Mendelian Inheritance in Man (OMIM), a Knowledgebase of Human Genes and Genetic Disorders. *Nucleic Acids Res* **2005**, *33*, 514–517, doi:10.1093/nar/gki033.

28. Piñero, J.; Ramírez-Anguita, J.M.; Saüch-Pitarch, J.; Ronzano, F.; Centeno, E.; Sanz, F.; Furlong, L.I. The DisGeNET Knowledge Platform for Disease Genomics: 2019 Update. *Nucleic Acids Res* **2020**, *48*, D845–D855, doi:10.1093/nar/gkz1021.

29. Piñero, J.; Bravo, Á.; Queralt-Rosinach, N.; Gutiérrez-Sacristán, A.; Deu-Pons, J.; Centeno, E.; García-García, J.; Sanz, F.; Furlong, L.I. DisGeNET: A Comprehensive Platform Integrating Information on Human Disease-Associated Genes and Variants. *Nucleic Acids Res* **2017**, *45*, D833–D839, doi:10.1093/nar/gkw943.

30. Piñero, J.; Queralt-Rosinach, N.; Bravo, À.; Deu-Pons, J.; Bauer-Mehren, A.; Baron, M.; Sanz, F.; Furlong, L.I. DisGeNET: A Discovery Platform for the Dynamical Exploration of Human Diseases and Their Genes. *Database* **2015**, *2015*, 1–17, doi:10.1093/database/bav028.

31. Bateman, A.; Martin, M.J.; Orchard, S.; Magrane, M.; Agivetova, R.; Ahmad, S.; Alpi, E.; Bowler-Barnett, E.H.; Britto, R.; Bursteinas, B.; et al. UniProt: The Universal Protein Knowledgebase in 2021. *Nucleic Acids Res* **2021**, *49*, D480–D489, doi:10.1093/nar/gkaa1100.

32. Davis, A.P.; Grondin, C.J.; Johnson, R.J.; Sciaky, D.; Wiegers, J.; Wiegers, T.C.; Mattingly, C.J. Comparative Toxicogenomics Database (CTD): Update 2021. *Nucleic Acids Res* **2021**, *49*, doi:10.1093/nar/gkaa891.

33. Rehm, H.L.; Berg, J.S.; Brooks, L.D.; Bustamante, C.D.; Evans, J.P.; Landrum, M.J.; Ledbetter, D.H.; Maglott, D.R.; Martin, C.L.; Nussbaum, R.L.; et al. ClinGen - The Clinical Genome Resource. *N Engl J Med* **2015**, *23*, 2235–2242, doi:10.1056/NEJMsr1406261.

34. Tamborero, D.; Rubio-Perez, C.; Deu-Pons, J.; Schroeder, M.P.; Vivancos, A.; Rovira, A.; Tusquets, I.; Albanell, J.; Rodon, J.; Tabernero, J.; et al. Cancer Genome Interpreter Annotates the Biological and Clinical Relevance of Tumor Alterations., doi:10.1186/s13073-018-0531-8.

35. Gutiérrez-Sacristá, A.; Grosdidier, S.; Valverde, O.; Torrens, M.; Bravo, À.; Piñero, J.; Sanz, F.; Furlong, L.I. PsyGeNET: A Knowledge Platform on Psychiatric Disorders and Their Genes. *Bioinformatics* **2015**, *30*, 3075–3077, doi:10.1093/bioinformatics/btv301.

36. Smith, C.L.; Blake, J.A.; Kadin, J.A.; Richardson, J.E.; Bult, C.J. Mouse Genome Database (MGD)-2018: Knowledgebase for the Laboratory Mouse. *Nucleic Acids Res* **2018**, *46*, doi:10.1093/nar/gkx1006.

37. Laulederkind, S.J.F.; Hayman, G.T.; Wang, S.J.; Smith, J.R.; Petri, V.; Hoffman, M.J.; de Pons, J.; Tutaj, M.A.; Ghiasvand, O.; Tutaj, M.; et al. A Primer for the Rat Genome Database (RGD). *Methods Mol Biol* **2018**, *1757*, 163–209, doi:10.1007/978-1-4939-7737-6_8.

38. Gargano, M.; Matentzoglu, N.; Carmody, L.C.; Lewis-Smith, D.; Vasilevsky, N.A.; Danis, D.; Balagura, G.; Baynam, G.; Brower, A.M.; Krause, R.; et al. The Human Phenotype Ontology in 2021. *Nucleic Acids Res* **2020**, *49*, 1207–1217, doi:10.1093/nar/gkaa1043.

39. Landrum, M.J.; Kattman, B.L. ClinVar at Five Years: Delivering on the Promise. *Hum Mutat* **2018**, *39*, 1623–1630, doi:10.1002/HUMU.23641.

40. Jassal, B.; Matthews, L.; Viteri, G.; Gong, C.; Lorente, P.; Fabregat, A.; Sidiropoulos, K.; Cook, J.; Gillespie, M.; Haw, R.; et al. The Reactome Pathway Knowledgebase. *Nucleic Acids Res* **2020**, *48*, D498–D503, doi:10.1093/nar/gkz1031.

41. Kuzmanov, U.; Emili, A. Protein-Protein Interaction Networks: Probing Disease Mechanisms Using Model Systems. *Genome Med* **2013**, *5*, doi:10.1186/gm441.

42. Milo, R. What Is the Total Number of Protein Molecules per Cell Volume? A Call to Rethink Some Published Values. *BioEssays* **2013**, *35*, 1050–1055, doi:10.1002/bies.201300066.

43. Stumpf, M.P.H.; Thorne, T.; de Silva, E.; Stewart, R.; Jun An, H.; Lappe, M.; Wiuf, C. Estimating the Size of the Human Interactome. *PNAS* **2008**, *105*, 6959–6964, doi:10.1073/pnas.0708078105.

44. Fields, S.; Song, O.K. A Novel Genetic System to Detect Protein–Protein Interactions. *Nature* **1989**, *340*, 245–246, doi:10.1038/340245a0.

45. Stynen, B.; Tournu, H.; Tavernier, J.; van Dijck, P. Diversity in Genetic in Vivo Methods for Protein-Protein Interaction Studies: From the Yeast Two-Hybrid System to the Mammalian Split-Luciferase System. *Microbiol Mol Biol Rev* **2012**, *76*, 331–382, doi:10.1128/MMBR.05021-11.

46. Titeca, K.; Lemmens, I.; Tavernier, J.; Eyckerman, S. Discovering Cellular Protein-Protein Interactions: Technological Strategies and Opportunities. *Mass Spectrom Rev* **2019**, *38*, 79–111, doi:10.1002/mas.21574.

47. Alonso-López, D.; Campos-Laborie, F.J.; Gutiérrez, M.A.; Lambourne, L.; Calderwood, M.A.; Vidal, M.; de Las Rivas, J. APID Database: Redefining Protein-Protein Interaction Experimental Evidences and Binary Interactomes. *Database* **2019**, *2019*, 1–8, doi:10.1093/database/baz005.

48. Alonso-López, D.; Gutiérrez, M.A.; Lopes, K.P.; Prieto, C.; Santamaría, R.; de Las Rivas, J. APID Interactomes: Providing Proteome-Based Interactomes with Controlled Quality for Multiple Species and Derived Networks. *Nucleic Acids Res* **2016**, *44*, W529–W535, doi:10.1093/nar/gkw363.

49. Stark, C.; Breitkreutz, B.J.; Reguly, T.; Boucher, L.; Breitkreutz, A.; Tyers, M. BioGRID: A General Repository for Interaction Datasets. *Nucleic Acids Res* **2006**, *34*, 535–539, doi:10.1093/nar/gkj109.

50. Salwinski, L.; Miller, C.S.; Smith, A.J.; Pettit, F.K.; Bowie, J.U.; Eisenberg, D. The Database of Interacting Proteins: 2004 Update. *Nucleic Acids Res* **2004**, *32*, 449–451, doi:10.1093/nar/gkh086.

51. Peri, S.; Navarro, J.D.; Amanchy, R.; Kristiansen, T.Z.; Jonnalagadda, C.K.; Surendranath, V.; Niranjan, V.; Muthusamy, B.; Gandhi, T.K.B.; Gronborg, M.; et al. Development of Human Protein Reference Database as an Initial Platform for Approaching Systems Biology in Humans. *Genome Res* **2003**, *13*, 2363–2371, doi:10.1101/gr.1680803.

52. Orchard, S.; Ammari, M.; Aranda, B.; Breuza, L.; Briganti, L.; Broackes-Carter, F.; Campbell, N.H.; Chavali, G.; Chen, C.; Del-Toro, N.; et al. The MIntAct Project - IntAct as a Common Curation Platform for 11 Molecular Interaction Databases. *Nucleic Acids Res* **2014**, *42*, 358–363, doi:10.1093/nar/gkt1115.

53. Licata, L.; Briganti, L.; Peluso, D.; Perfetto, L.; Iannuccelli, M.; Galeota, E.; Sacco, F.; Palma, A.; Nardozza, A.P.; Santonico, E.; et al. MINT, the Molecular Interaction Database: 2012 Update. *Nucleic Acids Res* **2012**, *40*, 857–861, doi:10.1093/nar/gkr930.

54. Luck, K.; Kim, D.K.; Lambourne, L.; Spirohn, K.; Begg, B.E.; Bian, W.; Brignall, R.; Cafarelli, T.; Campos-Laborie, F.J.; Charloteaux, B.; et al. A Reference Map of the Human Binary Protein Interactome. *Nature* **2020**, *580*, 402–408, doi:10.1038/s41586-020-2188-x.

55. Balakrishnan, V.K. *Graph Theory*; 1997; ISBN 0070054894.

56. Adali, T.; Ortega, A. Applications of Graph Theory. *Proceedings of the IEEE* **2018**, *106*, 784–786, doi:10.1109/JPROC.2018.2820300.

57. Zhu, X.; Gerstein, M.; Snyder, M. Getting Connected: Analysis and Principles of Biological Networks. *Genes Dev* **2007**, *21*, doi:10.1101/gad.1528707.

58. Cai, H.; Zheng, V.W.; Chen-Chuan Chang, K. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Trans Knowl Data Eng* **2018**, *30*, 1616–1637, doi:10.1109/TKDE.2018.2807452.

59. Koutra, D.; Ke, T.-Y.; Kang, U.; Horng Chau, D.; Kenneth Pao, H.-K.; Faloutsos, C. Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms. In Proceedings of the ECML PKDD; 2011; pp. 245–260.

60. Shannon, P.; Markiel, A.; Ozier, O.; Baliga, N.S.; Wang, J.T.; Ramage, D.; Amin, N.; Schwikowski, B.; Ideker, T. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks., doi:10.1101/gr.1239303.

61. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781* **2013**, doi:10.48550/arXiv.1301.3781.

62. Wang, X.; Cui, P.; Wang, J.; Pei, J.; Zhu, W.; Yang, S. Community Preserving Network Embedding. *Proceedings of the AAAI Conference on Artificial Intelligence* **2017**, *31*, 203–209, doi:10.1609/AAAI.V31I1.10488.

63. Burkov, A. *The Hundred-Page Machine Learing Book*; 2019; ISBN 1999579518.

64. Raschka, Sebastian.; Mirjalili, Vahid. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow*; Packt Publishing, 2017; ISBN 9781787125933.

65. Shalev-Shwartz, S.; Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*; Cambridge University Press, 2014; ISBN 9781107057135.

66. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830, doi:10.48550/arXiv.1201.0490.

67. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*; 2012; ISBN 9780123814791.

68. Trygg, J.; Wold, S. Orthogonal Projections to Latent Structures (O-PLS). *J Chemom* **2002**, *16*, 119–128, doi:10.1002/cem.695.

69. Thévenot, E.A.; Roux, A.; Xu, Y.; Ezan, E.; Junot, C. Analysis of the Human Adult Urinary Metabolome Variations with Age, Body Mass Index, and Gender by Implementing a Comprehensive Workflow for Univariate and OPLS Statistical Analyses. *J Proteome Res* **2015**, *14*, 3322–3335, doi:10.1021/acs.jproteome.5b00354.

70. Galindo-Prieto, B.; Eriksson, L.; Trygg, J. Variable Influence on Projection (VIP) for Orthogonal Projections to Latent Structures (OPLS). *J Chemom* **2014**, *28*, 623–632, doi:10.1002/CEM.2627.

71. Mehmood, T.; Liland, K.H.; Snipen, L.; Sæbø, S. A Review of Variable Selection Methods in Partial Least Squares Regression. *Chemometrics and Intelligent Laboratory Systems* **2012**, *118*, 62–69, doi:10.1016/j.chemolab.2012.07.010.

72. Bard, J.B.L.; Rhee, S.Y. Ontologies in Biology: Design, Applications and Future Challenges. *Nat Rev Genet* **2004**, *5*, 213–222, doi:10.1038/nrg1295.

73. Ashburner, M.; Ball, C.A.; Blake, J.A.; Botstein, D.; Butler, H.; Cherry, J.M.; Davis, A.P.; Dolinski, K.; Dwight, S.S.; Eppig, J.T.; et al. Gene Ontology: Tool for the Unification of Biology. *Nat Genet* **2000**, *25*, 25–29, doi:10.1038/75556.

74. Klopfenstein, D. v.; Zhang, L.; Pedersen, B.S.; Ramírez, F.; Vesztrocy, A.W.; Naldi, A.; Mungall, C.J.; Yunes, J.M.; Botvinnik, O.; Weigel, M.; et al. GOATOOLS: A Python Library for Gene Ontology Analyses. *Sci Rep* **2018**, *8*, doi:10.1038/s41598-018-28948-z.

75. Tweedie, S.; Braschi, B.; Gray, K.; Jones, T.E.M.; Seal, R.L.; Yates, B.; Bruford, E.A. Genenames.Org: The HGNC and VGNC Resources in 2021. *Nucleic Acids Res* **2021**, *49*, 939–946, doi:10.1093/nar/gkaa980.

76. Csárdi, G.; Nepusz, T. The Igraph Software Package for Complex Network Research. *InterJournal, Complex Systems* **2006**, *1695*.

77. Wang, R.S.; Loscalzo, J. Network-Based Disease Module Discovery by a Novel Seed Connector Algorithm with Pathobiological Implications. *J Mol Biol* **2018**, *430*, 2939–2950, doi:10.1016/J.JMB.2018.05.016.

78. Wheeler, D.L.; Church, D.M.; Federhen, S.; Lash, A.E.; Madden, T.L.; Pontius, J.U.; Schuler, G.D.; Schriml, L.M.; Sequeira, E.; Tatusova, T.A.; et al. Database Resources of the National Center for Biotechnology Information. *Nucleic Acids Res* **2021**, *49*, 10–17, doi:10.1093/nar/gkaa892.

79. Jensen, L.J.; Kuhn, M.; Stark, M.; Chaffron, S.; Creevey, C.; Muller, J.; Doerks, T.; Julien, P.; Roth, A.; Simonovic, M.; et al. STRING 8-a Global View on Proteins and Their Functional Interactions in 630 Organisms. *Nucleic Acids Res* **2009**, *37*, doi:10.1093/nar/gkn760.

80. Plotly Technologies Inc. Collaborative Data Science.

81. Mcinnes, I.B.; Schett, G. The Pathogenesis of Rheumatoid Arthritis. *N Engl J Med* **2011**, *365*, 2205–2219, doi:10.1056/NEJMra1004965.

82. Skeoch, S.; Bruce, I.N. Atherosclerosis in Rheumatoid Arthritis: Is It All about Inflammation? *Nat Rev Rheumatol* **2015**, *11*, 390–400, doi:10.1038/nrrheum.2015.40.

83. Hillid, A.; Gleimid, S.; Kiefer, F.; Sigoillotid, F.; Loureiro, J.; Jenkinsid, J.; Morrisid, M.K. Benchmarking Network Algorithms for Contextualizing Genes of Interest. *PLoS Comput Biol* **2019**, *15*, doi:10.1371/journal.pcbi.1007403.

84. Pellegrini, M.; Malod-Dognin, N.; Goldenberg, A.; Sharan, R.; Nelson, W.; Zitnik, M.; Wang, B.; Leskovec, J. To Embed or Not: Network Embedding as a Paradigm in Computational Biology. *Front Genet* **2019**, *10*, doi:10.3389/fgene.2019.00381.

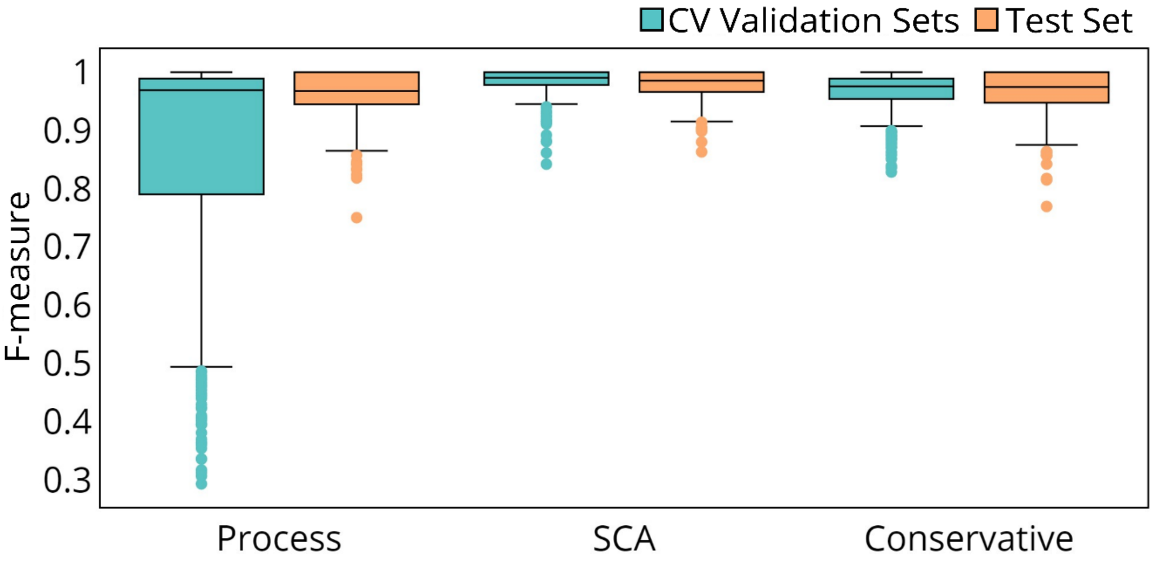
85. Liu, R.; Mancuso, C.A.; Yannakopoulos, A.; Johnson, K.A.; Krishnan, A. Supervised Learning Is an Accurate Method for Network-Based Gene Classification. *Bioinformatics* **2020**, *36*, doi:10.1093/bioinformatics/btaa150.

86. Grover, A.; Leskovec, J. Node2vec - Scalable Feature Learning for Networks. *KDD* **2016**, doi:10.1145/2939672.2939754.

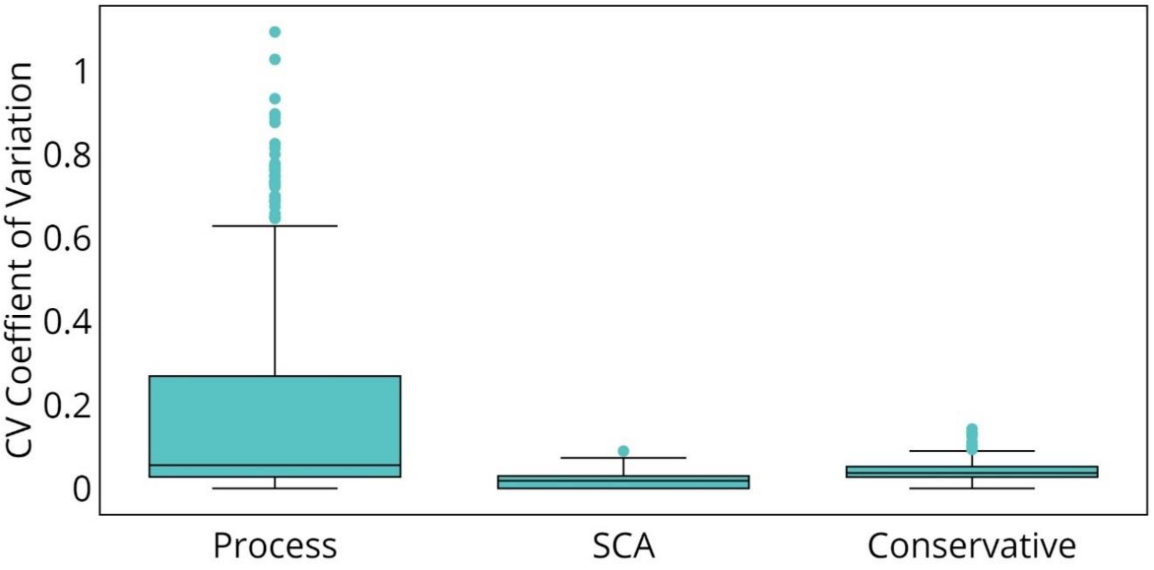
87. Peng, J.; Guan, J.; Shang, X. Predicting Parkinson’s Disease Genes Based on Node2vec and Autoencoder. *Front Genet* **2019**, *10*, doi:10.3389/fgene.2019.00226.

Supplementary Results

Data and code availability: https://github.com/GamaPintoLab/GAP-MINE



A



B

Figure S1 – GAP-MINE RWR performance during cross-validation. (A) Average F-Measure scores were observed in the 10 different validation sets of the cross-validation procedure. (B) Coefficient of variation of the obtained F-measure scores during the model's cross-validation procedure. The coefficient of variation is obtained by dividing the standard deviation of the 10 F-Measure scores by their average. Low values mean low variation of performance between the 10 different sets. Figure created with Plotly [80]

Table S1. GAP-MINE’s and Baseline model’s performance under the complete biological network. Each classifier is scored across three different metrics (F-Measure, Precision, and Recall). The presented scores correspond to the median, 1st, and 3rd quantiles (median (1st quantile – 3rd quantile)). For each performance metric, scoring metric, and module type, a paired Wilcoxon bilateral test was performed to compare GAP-MINE’s and the Baseline model’s performance. Bold values correspond to an observed statistical significance ($p<0.05$) between the two classifiers.

Metric	Module	Classifier	F-Measure	Precision	Recall
Hypergeometric Test	Process	Baseline	0.368 (0.2-0.578)	0.417 (0.231-0.65)	**0.357 (0.182-0.6)**
		GAP-MINE	0.343 (0.188-0.609)	**0.429 (0.2-0.684)**	0.333 (0.182-0.579)
	Disease SCA	Baseline	0.137 (0.104-0.182)	0.119 (0.086-0.167)	**0.167 (0.111-0.252)**
		GAP-MINE	0.139 (0.101-0.179)	**0.125 (0.088-0.168)**	0.167 (0.111-0.235)
	Disease Conservative	Baseline	0.054 (0.0-0.105)	0.045 (0.0-0.095)	**0.067 (0.0-0.125)**
		GAP-MINE	0.056 (0.0-0.108)	**0.052 (0.0-0.118)**	0.061 (0.0-0.111)
Closeness	Process	Baseline	0.229 (0.111-0.483)	0.25 (0.125-0.5)	0.25 (0.105-0.5)
		GAP-MINE	**0.323 (0.182-0.571)**	**0.333 (0.188-0.625)**	**0.35 (0.182-0.545)**
	Disease SCA	Baseline	0.1 (0.066-0.137)	0.107 (0.065-0.149)	0.1 (0.056-0.148)
		GAP-MINE	**0.109 (0.074-0.15)**	**0.111 (0.067-0.176)**	**0.116 (0.071-0.168)**
	Disease Conservative	Baseline	0.038 (0.0-0.08)	0.027 (0.0-0.071)	0.074 (0.0-0.136)
		GAP-MINE	0.042 (0.0-0.087)	0.031 (0.0-0.071)	**0.077 (0.0-0.15)**
Betweenness	Process	Baseline	0.0 (0.0-0.009)	0.0 (0.0-0.005)	**0.0 (0.0-0.083)**
		GAP-MINE	0.0 (0.0-0.008)	0.0 (0.0-0.004)	0.0 (0.0-0.062)
	Disease SCA	Baseline	**0.013 (0.0-0.024)**	**0.007 (0.0-0.013)**	**0.067 (0.0-0.156)**
		GAP-MINE	0.005 (0.0-0.022)	0.002 (0.0-0.013)	0.017 (0.0-0.091)
	Disease Conservative	Baseline	**0.007 (0.0-0.015)**	**0.003 (0.0-0.008)**	**0.062 (0.0-0.154)**
		GAP-MINE	0.0 (0.0-0.013)	0.0 (0.0-0.008)	0.0 (0.0-0.05)
Fraction Betweenness	Process	Baseline	**0.237 (0.129-0.407)**	**0.229 (0.143-0.375)**	**0.273 (0.125-0.5)**
		GAP-MINE	0.143 (0.006-0.308)	0.167 (0.003-0.438)	0.158 (0.056-0.333)
	Disease SCA	Baseline	**0.182 (0.132-0.226)**	0.194 (0.15-0.25)	**0.174 (0.117-0.231)**
		GAP-MINE	0.176 (0.124-0.222)	0.185 (0.137-0.255)	0.167 (0.105-0.226)
	Disease Conservative	Baseline	**0.044 (0.009-0.075)**	**0.029 (0.005-0.06)**	0.091 (0.023-0.154)
		GAP-MINE	0.009 (0.0-0.049)	0.005 (0.0-0.053)	**0.083 (0.0-0.286)**
Random Walks with Restart	Process	Baseline	0.963 (0.936-1.0)	1.0 (0.905-1.0)	**1.0 (1.0-1.0)**
		GAP-MINE	**0.968 (0.944-1.0)**	**1.0 (0.92-1.0)**	1.0 (0.938-1.0)
	Disease SCA	Baseline	0.977 (0.951-1.0)	0.967 (0.913-1.0)	**1.0 (1.0-1.0)**
		GAP-MINE	**0.986 (0.966-1.0)**	**1.0 (0.965-1.0)**	1.0 (0.965-1.0)
	Disease Conservative	Baseline	0.976 (0.952-1.0)	1.0 (0.92-1.0)	**1.0 (1.0-1.0)**
		GAP-MINE	0.974 (0.947-1.0)	**1.0 (0.929-1.0)**	1.0 (0.95-1.0)

Table S2. GAP-MINE’s and Baseline model’s performance using the Protein 80% network. Each classifier is scored across three different metrics (F-Measure, Precision, and Recall). The presented scores correspond to the median, 1st, and 3rd quantiles (median (1st quantile – 3rd quantile)). For each performance metric, scoring metric, and module type, a paired Wilcoxon bilateral test was performed to compare GAP-MINE’s and the Baseline model’s performance. Bold values correspond to an observed statistical significance ($p<0.05$) between the two classifiers.

Metric	Module	Classifier	F-Measure	Precision	Recall
Hypergeometric Test	Process	Baseline	0.014 (0.006-0.026)	0.007 (0.003-0.013)	**0.25 (0.1-0.571)**
		GAP-MINE	**0.016 (0.0-0.035)**	**0.008 (0.0-0.019)**	0.125 (0.0-0.385)
	Disease SCA	Baseline	0.105 (0.067-0.15)	0.079 (0.048-0.122)	**0.167 (0.1-0.25)**
		GAP-MINE	**0.107 (0.065-0.157)**	**0.094 (0.054-0.154)**	0.133 (0.071-0.208)
	Disease Conservative	Baseline	0.032 (0.0-0.08)	0.02 (0.0-0.071)	0.062 (0.0-0.125)
		GAP-MINE	**0.043 (0.0-0.098)**	**0.029 (0.0-0.091)**	0.062 (0.0-0.13)
Closeness	Process	Baseline	0.235 (0.118-0.471)	0.25 (0.111-0.5)	0.25 (0.111-0.5)
		GAP-MINE	**0.4 (0.233-0.588)**	**0.429 (0.231-0.667)**	**0.4 (0.231-0.571)**
	Disease SCA	Baseline	0.097 (0.058-0.14)	0.087 (0.052-0.13)	0.111 (0.062-0.179)
		GAP-MINE	**0.105 (0.066-0.156)**	**0.103 (0.062-0.167)**	0.111 (0.062-0.182)
	Disease Conservative	Baseline	0.033 (0.0-0.087)	0.021 (0.0-0.074)	0.067 (0.0-0.125)
		GAP-MINE	**0.057 (0.0-0.105)**	**0.043 (0.0-0.1)**	**0.083 (0.0-0.143)**
Betweenness	Process	Baseline	**0.571 (0.429-0.727)**	**0.5 (0.357-0.636)**	**0.733 (0.5-0.889)**
		GAP-MINE	0.016 (0.0-0.035)	0.008 (0.0-0.019)	0.125 (0.0-0.385)
	Disease SCA	Baseline	**0.5 (0.444-0.568)**	**0.419 (0.366-0.484)**	**0.641 (0.536-0.749)**
		GAP-MINE	0.107 (0.065-0.157)	0.094 (0.054-0.154)	0.133 (0.071-0.208)
	Disease Conservative	Baseline	0.321 (0.222-0.4)	0.3 (0.222-0.389)	0.333 (0.222-0.462)
		GAP-MINE	0.0 (0.0-0.013)	0.0 (0.0-0.007)	0.0 (0.0-0.5)
Fraction Betweenness	Process	Baseline	**0.235 (0.133-0.375)**	0.231 (0.125-0.357)	**0.263 (0.125-0.495)**
		GAP-MINE	0.19 (0.052-0.381)	**0.25 (0.039-0.5)**	0.2 (0.083-0.385)
	Disease SCA	Baseline	**0.148 (0.1-0.194)**	0.135 (0.091-0.188)	**0.167 (0.1-0.233)**
		GAP-MINE	0.131 (0.08-0.182)	0.137 (0.083-0.2)	0.125 (0.071-0.192)
	Disease Conservative	Baseline	**0.038 (0.0-0.08)**	**0.024 (0.0-0.062)**	**0.071 (0.0-0.133)**
		GAP-MINE	0.007 (0.0-0.051)	0.003 (0.0-0.04)	0.071 (0.0-0.208)
Random Walks with Restart	Process	Baseline	**0.963 (0.933-1.0)**	1.0 (0.9-1.0)	**1.0 (1.0-1.0)**
		GAP-MINE	0.96 (0.933-1.0)	**1.0 (0.938-1.0)**	1.0 (0.9-1.0)
	Disease SCA	Baseline	0.833 (0.767-0.889)	1.0 (0.938-1.0)	0.727 (0.643-0.833)
		GAP-MINE	0.83 (0.764-0.893)	**1.0 (0.952-1.0)**	0.727 (0.64-0.824)
	Disease Conservative	Baseline	**0.812 (0.741-0.872)**	**1.0 (1.0-1.0)**	**0.692 (0.6-0.778)**
		GAP-MINE	0.8 (0.727-0.857)	1.0 (1.0-1.0)	0.667 (0.581-0.769)

Table S3. GAP-MINE's and Baseline model's performance using the PPI 80% network. Each classifier is scored across three different metrics (F-Measure, Precision, and Recall). The presented scores correspond to the median, 1st, and 3rd quantiles (median (1st quantile – 3rd quantile)). For each performance metric, scoring metric, and module type, a paired Wilcoxon bilateral test was performed to compare GAP-MINE's and the Baseline model's performance. Bold values correspond to an observed statistical significance ($p<0.05$) between the two classifiers.

Metric	Module	Classifier	F-Measure	Precision	Recall
Hypergeometric Test	Process	Baseline	0.02 (0.0-0.035)	0.01 (0.0-0.019)	**0.154 (0.0-0.455)**
		GAP-MINE	0.012 (0.0-0.039)	**0.006 (0.0-0.023)**	0.091 (0.0-0.3)
	Disease SCA	Baseline	**0.125 (0.083-0.169)**	**0.108 (0.071-0.15)**	**0.158 (0.103-0.224)**
		GAP-MINE	0.011 (0.0-0.022)	0.006 (0.0-0.013)	0.048 (0.0-0.124)
	Disease Conservative	Baseline	**0.047 (0.0-0.091)**	**0.033 (0.0-0.083)**	**0.067 (0.0-0.125)**
		GAP-MINE	0.0 (0.0-0.013)	0.0 (0.0-0.007)	0.0 (0.0-0.067)
Closeness	Process	Baseline	0.222 (0.111-0.444)	0.25 (0.111-0.5)	0.231 (0.1-0.462)
		GAP-MINE	**0.387 (0.243-0.588)**	**0.417 (0.25-0.667)**	**0.4 (0.235-0.588)**
	Disease SCA	Baseline	**0.095 (0.059-0.133)**	**0.095 (0.059-0.136)**	**0.095 (0.056-0.15)**
		GAP-MINE	0.011 (0.0-0.023)	0.006 (0.0-0.013)	0.045 (0.0-0.134)
	Disease Conservative	Baseline	**0.037 (0.0-0.077)**	**0.025 (0.0-0.065)**	**0.067 (0.0-0.13)**
		GAP-MINE	0.0 (0.0-0.016)	0.0 (0.0-0.009)	0.0 (0.0-0.091)
Betweenness	Process	Baseline	**0.566 (0.435-0.709)**	**0.461 (0.35-0.611)**	**0.739 (0.562-0.9)**
		GAP-MINE	0.005 (0.0-0.01)	0.002 (0.0-0.005)	0.059 (0.0-0.833)
	Disease SCA	Baseline	**0.514 (0.479-0.556)**	**0.371 (0.338-0.41)**	**0.851 (0.789-0.9)**
		GAP-MINE	0.01 (0.0-0.022)	0.005 (0.0-0.013)	0.045 (0.0-0.228)
	Disease Conservative	Baseline	**0.382 (0.316-0.442)**	**0.289 (0.242-0.333)**	**0.55 (0.434-0.667)**
		GAP-MINE	0.004 (0.0-0.014)	0.002 (0.0-0.008)	0.033 (0.0-0.417)
Fraction Betweenness	Process	Baseline	0.0 (0.0-0.013)	0.0 (0.0-0.007)	0.0 (0.0-0.071)
		GAP-MINE	**0.222 (0.096-0.424)**	**0.286 (0.111-0.5)**	**0.2 (0.091-0.417)**
	Disease SCA	Baseline	**0.163 (0.118-0.213)**	0.171 (0.125-0.235)	**0.162 (0.109-0.224)**
		GAP-MINE	0.154 (0.103-0.2)	0.167 (0.113-0.235)	0.143 (0.089-0.208)
	Disease Conservative	Baseline	**0.04 (0.0-0.077)**	**0.027 (0.0-0.059)**	0.071 (0.0-0.133)
		GAP-MINE	0.009 (0.003-0.043)	0.004 (0.002-0.033)	**0.15 (0.048-0.429)**
Random Walks with Restart	Process	Baseline	0.963 (0.936-1.0)	0.96 (0.9-1.0)	**1.0 (1.0-1.0)**
		GAP-MINE	0.968 (0.938-1.0)	**1.0 (0.929-1.0)**	1.0 (0.927-1.0)
	Disease SCA	Baseline	0.967 (0.935-1.0)	0.958 (0.911-1.0)	**1.0 (0.963-1.0)**
		GAP-MINE	**0.974 (0.955-1.0)**	**1.0 (0.97-1.0)**	0.967 (0.933-1.0)
	Disease Conservative	Baseline	**0.966 (0.941-1.0)**	1.0 (0.929-1.0)	**1.0 (0.938-1.0)**
		GAP-MINE	0.957 (0.923-0.984)	**1.0 (0.941-1.0)**	0.95 (0.9-1.0)

Table S4. GAP-MINE's and Baseline model's performance using a network of 10% added noise. Each classifier is scored across three different metrics (F-Measure, Precision, and Recall). The presented scores correspond to the median, 1st, and 3rd quantiles (median (1st quantile – 3rd quantile)). For each performance metric, scoring metric, and module type, a paired Wilcoxon bilateral test was performed to compare GAP-MINE's and the Baseline model's performance. Bold values correspond to an observed statistical significance (p<0.05) between the two classifiers.

Metric	Module	Classifier	F-Measure	Precision	Recall
Hypergeometric Test	Process	Baseline	**0.37 (0.25-0.6)**	0.4 (0.25-0.611)	**0.375 (0.25-0.619)**
		GAP-MINE	0.375 (0.19-0.6)	**0.444 (0.25-0.722)**	0.333 (0.167-0.562)
	Disease SCA	Baseline	0.14 (0.097-0.185)	0.118 (0.077-0.159)	**0.182 (0.125-0.25)**
		GAP-MINE	0.136 (0.099-0.178)	0.118 (0.083-0.16)	0.167 (0.113-0.227)
	Disease Conservative	Baseline	0.059 (0.0-0.103)	0.045 (0.0-0.098)	0.083 (0.0-0.143)
		GAP-MINE	**0.065 (0.0-0.125)**	**0.057 (0.0-0.118)**	0.083 (0.0-0.143)
Closeness	Process	Baseline	0.211 (0.105-0.467)	0.231 (0.111-0.471)	0.25 (0.1-0.5)
		GAP-MINE	**0.308 (0.154-0.56)**	**0.333 (0.167-0.667)**	**0.353 (0.167-0.556)**
	Disease SCA	Baseline	0.098 (0.071-0.14)	0.098 (0.064-0.13)	0.116 (0.071-0.158)
		GAP-MINE	0.093 (0.062-0.134)	0.088 (0.054-0.131)	0.109 (0.062-0.167)
	Disease Conservative	Baseline	0.033 (0.0-0.077)	0.024 (0.0-0.066)	0.059 (0.0-0.118)
		GAP-MINE	**0.036 (0.0-0.08)**	0.026 (0.0-0.068)	**0.067 (0.0-0.143)**
Betweenness	Process	Baseline	0.006 (0.0-0.009)	0.003 (0.0-0.005)	**1.0 (0.0-1.0)**
		GAP-MINE	**0.024 (0.0-0.098)**	**0.013 (0.0-0.077)**	0.083 (0.0-0.231)
	Disease SCA	Baseline	0.012 (0.0-0.023)	0.006 (0.0-0.013)	**0.071 (0.0-0.266)**
		GAP-MINE	0.011 (0.0-0.024)	0.005 (0.0-0.014)	0.042 (0.0-0.239)
	Disease Conservative	Baseline	**0.007 (0.0-0.016)**	**0.004 (0.0-0.009)**	**0.05 (0.0-0.125)**
		GAP-MINE	0.0 (0.0-0.01)	0.0 (0.0-0.005)	0.0 (0.0-0.059)
Fraction Betweenness	Process	Baseline	**0.235 (0.133-0.364)**	**0.222 (0.143-0.364)**	**0.273 (0.143-0.455)**
		GAP-MINE	0.154 (0.009-0.303)	0.2 (0.005-0.375)	0.154 (0.062-0.333)
	Disease SCA	Baseline	**0.182 (0.125-0.222)**	0.182 (0.135-0.235)	**0.184 (0.125-0.233)**
		GAP-MINE	0.16 (0.11-0.211)	0.182 (0.125-0.244)	0.154 (0.094-0.225)
	Disease Conservative	Baseline	**0.05 (0.016-0.1)**	**0.034 (0.01-0.071)**	0.1 (0.04-0.182)
		GAP-MINE	0.009 (0.0-0.067)	0.005 (0.0-0.062)	0.079 (0.0-0.2)
Random Walks with Restart	Process	Baseline	0.917 (0.874-0.952)	0.857 (0.81-0.917)	**1.0 (0.944-1.0)**
		GAP-MINE	**0.957 (0.917-1.0)**	**0.952 (0.889-1.0)**	1.0 (0.938-1.0)
	Disease SCA	Baseline	0.914 (0.887-0.949)	0.85 (0.8-0.91)	**1.0 (1.0-1.0)**
		GAP-MINE	**0.97 (0.923-1.0)**	**0.968 (0.906-1.0)**	1.0 (0.947-1.0)
	Disease Conservative	Baseline	0.923 (0.889-0.952)	0.867 (0.815-0.917)	**1.0 (1.0-1.0)**
		GAP-MINE	**0.952 (0.913-0.978)**	**0.938 (0.885-1.0)**	1.0 (0.935-1.0)