# Examining approaches to target validation and drug repurposing in large scale genomic projects

Submitted in partial fulfillment of the requirements of the Degree of Doctor of Philosophy

Daniel Jason Rhodes

Supervisors:

Professor Michael R. Barnes

Professor Sir Mark Caulfield

The William Harvey Research Institute

## Statement of originality

I, Daniel Rhodes, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below. I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material. I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis. I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university. The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: Daniel Rhodes
Date: 02/03/2021

## Conflicts of interest

I am an employee at Genomics England at the date of submission. Research carried out within this thesis was totally independent of my role here to this date. I have no other conflicts of interest.

# Acknowledgements

I would like to thank my funders, the Farr Institute, for making this thesis possible. I would also like to thank my supervisors, Professor Michael Barnes and Professor Sir Mark Caulfield for their insight, guidance and patience throughout this process, and the extended group of the Centre for Translational Bioinformatics for their support. I would also like to extend my gratitude to Dr. Daniel MacArthur and his team at the Broad Institute of MIT and Harvard for an enriching placement and collaboration, and Dr. Sherry Cao and her team at AbbVie for their invaluable input and collaborative efforts. Finally I would like to thank my host research institute, the William Harvey Research Institute, and university, Queen Mary University of London.

# Table of contents

# Figures and tables

# Abstract

Drug repurposing presents an opportunity to quickly produce new medications in a cost effective manner. This is especially important in rare diseases where patients are frequently underserved. Here, we apply various methods to first select good targets for repurposing. We analyse loss-of-function (LoF) data, and assess its role in informing drug discovery. We achieve this by curating, aggregating and labelling LoF data and then building a model to predict genes that may harbour homozygous LoF with no negative associated phenotypes. We produce a model with a relatively high degree of accuracy and recall (F-score 0.7), generating 442 predicted genes in addition to 1,744 from aggregation. Following this, we assess whether such data could inform drug discovery in collaboration with AbbVie, an industrial partner. Through the study of historic drug data, comparing our LoF labels with data from previous studies detailing the effect of genetic knowledge on drug discovery, and against the loss-of-function observed/expected upper bound fraction (LOEUF) score, a metric of constraint, we demonstrate that this data adds significant value to drug discovery. Finally, we build a database focussing on rare diseases, and use LoF data, in addition to drug data and expertly curated gene panels to nominate candidates for repurposing. This database will be made available for researchers within the GEL community, such that avenues for repurposing can be further explored.

# Chapter 1: Introduction

# 1.1 Introduction

## 1.1.1 An introduction to drug discovery

The process of drug discovery is long, complex, very expensive and requires the input of stakeholders from academia, industry and patient groups. The first step is the establishment of a molecular target of interest, generally stemming from an understanding of disease aetiology. This basic science requires the use of model systems such as animal models and human cell lines, genetic data, epidemiological data and many other fields of biology. Upon the identification of this target, a 12 year process (on average) begins, with molecule screening, lead molecule optimisation, preclinical trials and finally 4 phase human clinical trials (see Fig. 1.1).

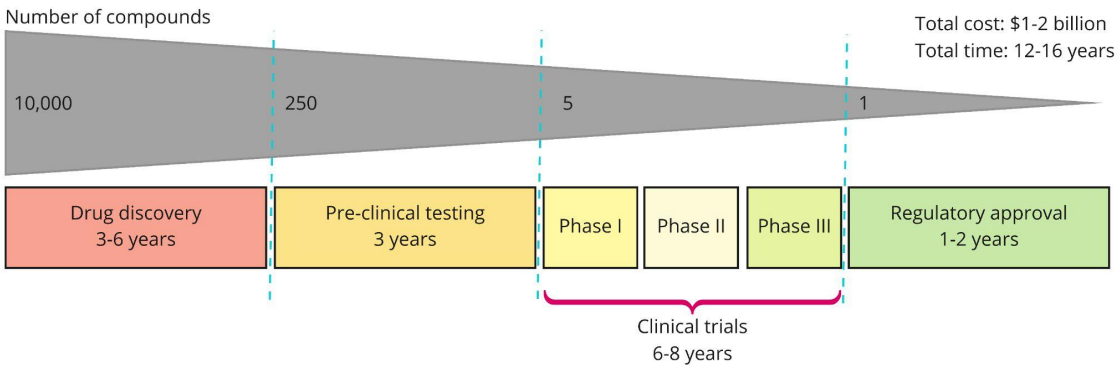



**Figure 1.1 - The drug development pipeline.** Adapted from Nosengo (2016).

The process from pre-clinical trials onwards costs a median of $985 million, although estimates range from $314 million to as high as $2.8 billion [1–3]. This cost accounts for failed drug development attempts, which represents 96% of cases [4–8]. This severe attrition rate has led to an examination of the crisis in pharmaceutical productivity, and attempts to improve methodology at each stage of the development pipeline. Here we will discuss some of the reasons for failure, and attempts to ameliorate the problem, including target validation, what we can learn from side effects, the role of genetics, and finally, drug repurposing.

#### 1.1.1.2 Target discovery and validation

A drug can only be as good as its target. Failure to show efficacy is the most common reason for clinical trial failure with between 57-90% of trials falling at this hurdle [5,9,10]. The reasons for this are multifactorial, with ill-considered trial design, including not recruiting a large enough trial group and improper selection of clinical and statistical end-points, and improper target selection both playing important roles. This is why target discovery and validation is so critical to the probability of success of any drug development pipeline. The classical approach to this is a basic science, classical (forward) genetics one, in which genes of interest are identified by studying the disease models and phenotypes of interest in model organisms, and identifying genes that are responsible for this phenotype [11]. This approach has been invaluable for the understanding of normal and aberrant biology, however it has proven to be a poor predictor of efficacy in drug development [12–14]. The second approach is reverse genetics, in which genes are targeted for inhibition or deletion, and then the phenotypic effects are studied. This approach has been particularly driven by the rise of genomics and bioinformatics in the early 2000s, and huge breakthroughs in technologies such as RNA interference and CRISPR-CAS9. Upon identification of a gene, e.g. through a genome-wide association study, the ‘discovery’ phase of drug development proceeds to ‘validation’. Targets of interest are systematically perturbed in *in vivo* and *in vitro* model systems allowing for the delineation of a gene's functional effect [15,16]. If the knockout of a target generates a phenotype of interest such as a model of diabetes, then modulation of this target may ameliorate the disease of interest. Then a ligand must be identified for this target to recapitulate this effect. A classical example of this approach can be found with PCSK9 [17]. Identification of a PCSK9 gain-of-function variant in a patient with familial hypercholesterolemia began to uncover the underlying biology of low-density lipoprotein (LDL) metabolism[18]. Subsequently, individuals with rare homozygous loss-of-function variants in this gene were found [19]. These individuals exhibited very low LDL levels but no tangible evidence of related health problems. These observations of naturally occurring biology led to a rapid translation of basic science to clinically approved drugs in the form of PCSK9 inhibitors. The ability to rapidly move from observations in humans, to defined models, with compounds following shortly after, demonstrates the power of reverse genetics for target validation [17].

Over the last few decades, high-throughput screening (HTS) has predominated the search for a ligand [20]. HTS aims to screen large numbers of compounds for bioactivity against a small panel of target proteins. Small to medium size screens may evaluate hundreds of compounds, but advancements in robotics and computing now allow for ultra-high throughput experiments to conduct in the order of $10^5$ compound assays a day [21]. The libraries of compounds are generated from combinatorial chemistry methods that generate many different permutations of chemical compounds. This process allows for a staggering number of possible combinations, with an estimated $10^{60}$ [22] compounds meeting Lipinski's "rule of five", a set of rules designed to identify potential drugs [23]. However, this search space proves to be intractable, with drug development driven in this manner having limited success. Factors associated with this will be discussed in more detail later in this chapter.

### 1.1.1.3 Side effects in drug design

#### 1.1.1.3.1 A background to side effects

Side effects (SEs) are the effect of a drug that is not intended as the primary effect. They can be either therapeutic (potentially highlighting repositioning opportunities - which will be covered later in the introduction) or adverse, and in common parlance the latter meaning is often implied. SEs can be wide ranging in form and severity and number, with an average of 69.8 side effects per drug reported on FDA drug labels [24]. However, such numbers may be the result of over-reporting that has occured due to increasingly high regulatory demands, with newer and more commonly prescribed drugs having very high SE burdens [24].

SEs can greatly affect a patient's quality of life, and represent a significant burden on healthcare. The willingness to tolerate SEs varies depending on the condition. Where acute and/or life-threatening diseases can be aggressively treated even in the knowledge of probable serious SEs (such as in cancer treatment); other conditions such as chronic and/or minor conditions can not accept SE profiles in which a patient's quality of life is too adversely affected. Patient groups including psychiatric and hypertension patients must often take drugs for a significant period, and in both cases, studies have shown that adherence to medication regimens are reduced in patients

suffering SEs. This leads to worse outcomes for patients as their disorders are less well managed, and increases health care costs due to increased use of healthcare resources [25,26].
Rare diseases frequently fall within this latter group, as severe but chronic disorders which may require life-time courses of treatment. As such, minimising SE profiles must be a key consideration when seeking to develop or repurpose new drugs for such a disease group.

SEs result in a significant proportion of hospital admissions, however heterogeneous evaluation of this subject makes it almost impossible to provide accurate figures on the resulting cost [27]. Estimates range from a cost of £466m per year for patients admitted to hospital in the NHS [28], to €943.40 to €7,192.36 per patient hospital admission across various other countries [29]. The real cost will be significantly higher due to most studies being limited to hospital based statistics, thereby not factoring economic cost due to increased sick-leave and poor adherence to treatment or increased engagement with primary care services. SE related mortality is similarly difficult to measure [30], but has been shown to be significant, especially as many such fatalities are clustered amongst a small group of drugs and may also demonstrate inappropriate drug prescribing or dosing by physicians [31].

In addition to the impact on patient health, SEs are a significant cause of drug development failure, with safety and efficacy concerns being the main culprits. Furthermore, 90% of drugs withdrawn in phase IV monitoring (after a drug has been approved) are withdrawn due to toxicity concerns [32,33]. Therefore understanding the cause of side effects, how to mitigate them, and what we might learn from them, is an important avenue of research.

#### 1.1.1.3.2 The causes of side effects

SEs arise from both intended and unintended consequences of drug administration. Broadly SEs are either pharmacodynamic or pharmacokinetic in nature, the former describes what the drug does to the body, in a target mediated manner, the later describes what the body does to the drug, usually mediated by processes of adsorption, dissemination, metabolism and excretion (ADME). The processes mediating ADME-related side effects are not the focus of this research so they are not explored further here, but should always be considered in the investigation of SEs.

Type A SEs make up a large proportion of side effects and are the result of the intended consequence of the drug having a knock-on, negative impact. For example, the blocking of coagulation by Warfarin to reduce risk of atrial fibrillation may also lead to severe bleeding and increased bruising. Such SEs are generally managed through dose control, whereby physicians will aim to achieve a dose that allows for maximal clinical impact with minimal SEs.

Type B, or idiosyncratic SEs are responses that are unexpected based on knowledge of the drug, and are less predictable as they may be rare or population dependent.

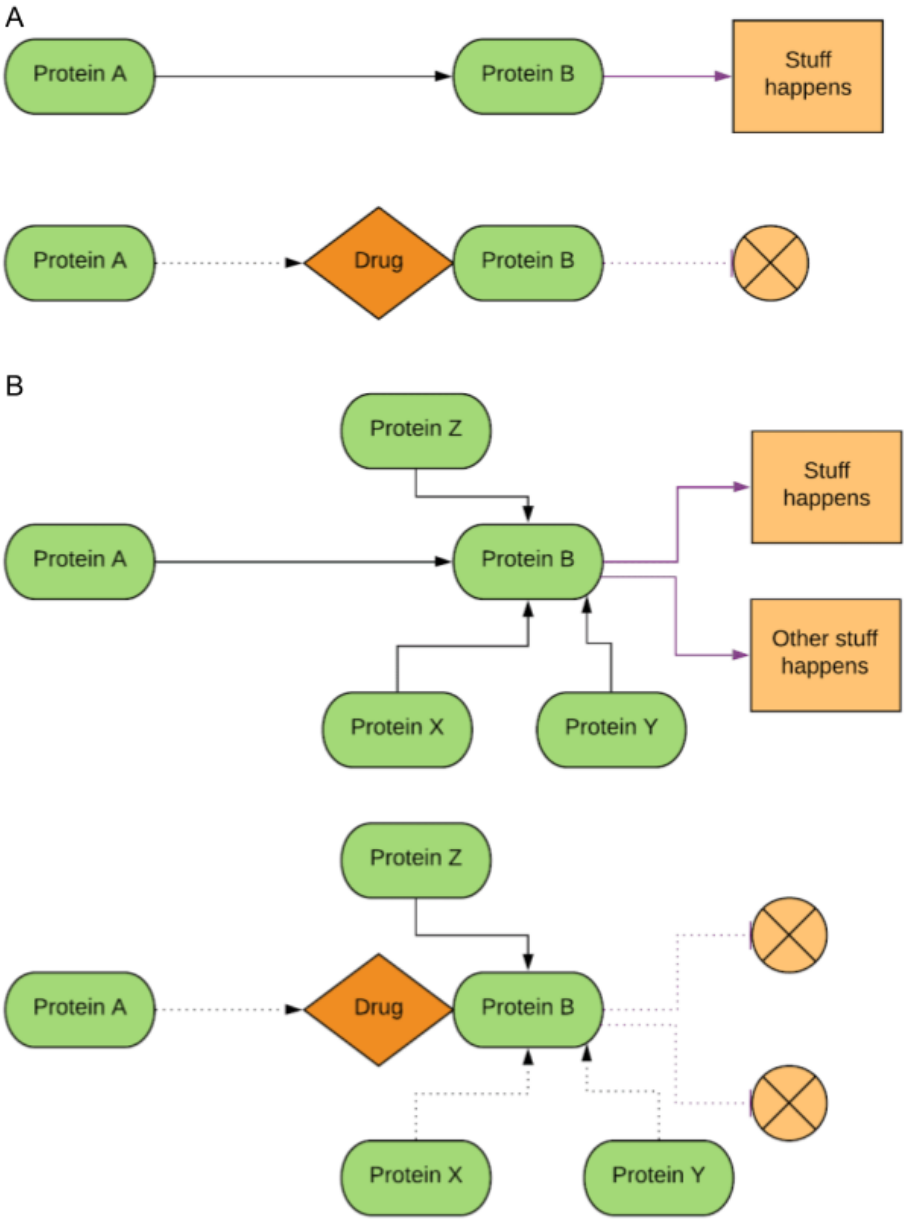



**Figure 1.2 - A diagrammatic representation of on target side effect causation.** A) A simplified model of protein-protein interaction leading to an intended biological

outcome, followed by the perturbation of this process using a drug. B) A more complex version of this model, illustrating why perturbation of a target may lead to unforeseen side effects through the disruption of other processes that may also be mediated via the target protein B.

The classical approach to drug design has been very target-centric and operated on an oversimplified model of protein interaction. As shown in Fig. 1.2A, targeting a process resulting from the interaction of protein A with protein B is hypothetically simple, requiring simply the blocking of that interaction, for example by chemically blocking interaction with the latter. The reality is that the interaction of proteins A and B is context dependent, and protein B may be involved in numerous processes with multiple possible binding proteins depending on factors such as substrate concentration or tissue localisation (Fig. 1.2B). For example, systematically targeting the interaction of A and B in the lung may also result in the disruption of other processes in other tissues such as the skin in ways that are unintended. Additionally, drug interactions at different domains on the same protein target could lead to differing outcomes. Such SEs are termed on-target SEs, in which the correct protein is targeted, but an unintended process is perturbed.

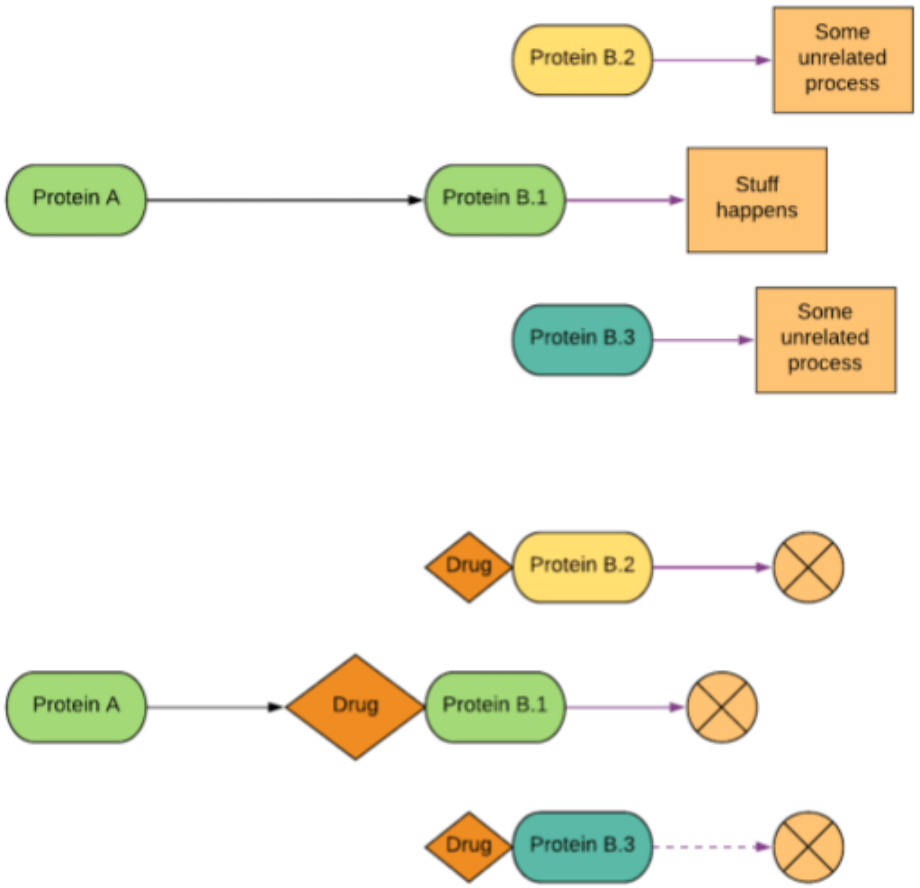



**Figure 1.3 - A diagrammatic representation of the effect of homologs in drug promiscuity.** Structurally similar proteins (Proteins B.2 + B.3) may also bind to the drug, resulting in the unintended blocking or modulation of related processes. This blocking may occur at differing potencies, such as for Protein B.3, where the dotted line denotes reduced efficacy.

To further complicate matters, proteins frequently share structural or sequence homology, and therefore, when designing a drug to target a particular domain, it may also target other proteins sharing this domain usually with reduced potency (see Fig. 1.3). This may also result in unintended processes being perturbed. This mechanism represents a common form of off-target SEs, in which unintended proteins are targeted, an outcome made probable due to the promiscuity of drugs. This latter class of side effects however provides us with an opportunity, as these similar proteins may be important in modulating disease pathways aside from the intended indication. Study of side effects has led to the successful repurposing of drugs, in which existing drugs are used in novel indications [34,35].

#### 1.1.1.3.3 The problem with side effect data

Despite the fact that SEs are detrimental to human health, expensive to healthcare services, and a great source of failure within the pharmaceutical industry, SE data are relatively scarce, and generally lacking in quality, possibly reflecting industrial over incentivisation of success at the expense of greater understanding derived from failure, or potentially the competitive nature of data on drug failure. This presents some major hurdles when researching this area.

In order for a drug to be licensed, regulatory agencies such as the FDA and the EMA require data on SEs collected within clinical trials. However, reports filed by drug companies to the FDA are often incomplete, with variables when reporting serious or fatal SEs often left blank, with completion of reports ranging from 24.4% to 67% (data reported refers to FDA data from 2014 [36]. Transparency is challenged with a lack of reporting of commercial sponsors of trials by academic investigators. Additional areas of concern include missing data such as the name of the investigational product, the number, type and severity of SEs and other data necessary to inform any proper assessments of the SE reporting data [37].

SEs are also reported at a national level through initiatives such as the Yellow Card Scheme in the UK and the FDA's Adverse Event Reporting System. However, over 100 of these systems exist globally. Due to the lack of standardisation, comparing SE profiles for drugs beyond that ascertained at trial stages, across territories is challenging. This is problematic due to the fact that drugs are known to exhibit different effects in different ethnic groups, as the genetic underpinning of a disease may be ethnicity dependent. Such an example can be found in the Framingham heart study cohort, in which it was observed that South Asians do not respond to heart medication in the same manner as white Caucasians [38].

With SEs being responsible for the failure of such a high percentage of clinical trials, in addition to reducing patient medication adherence, prioritising compounds and targets with a low risk of having severe SE profiles is an important consideration. Furthermore, SEs are direct evidence that the one drug to one target paradigm is reductive, and that drugs affect systems in broader terms than just the indication of interest. A more thorough accumulation of data surrounding SEs may help fill in the gaps in knowledge around how current and future drugs act. This could help to reduce the percentage of candidate compounds that are shelved as their full target complement is not understood, or stop trials from starting if deleterious cross-targeting can be accurately predicted.

### 1.1.1.4 Moving away from target centric drug design

Historically, drug design has been conducted by identifying a target, such as a disease modifying gene, and then screening this target against compounds. However, due to the large number of combinations of targets and compounds previously described, this search space is intractable for either chemical synthesis or *in silico* predictions. Thus even with the great increases in HTS scale, only a small fraction of possible compounds have been synthesised and tested.

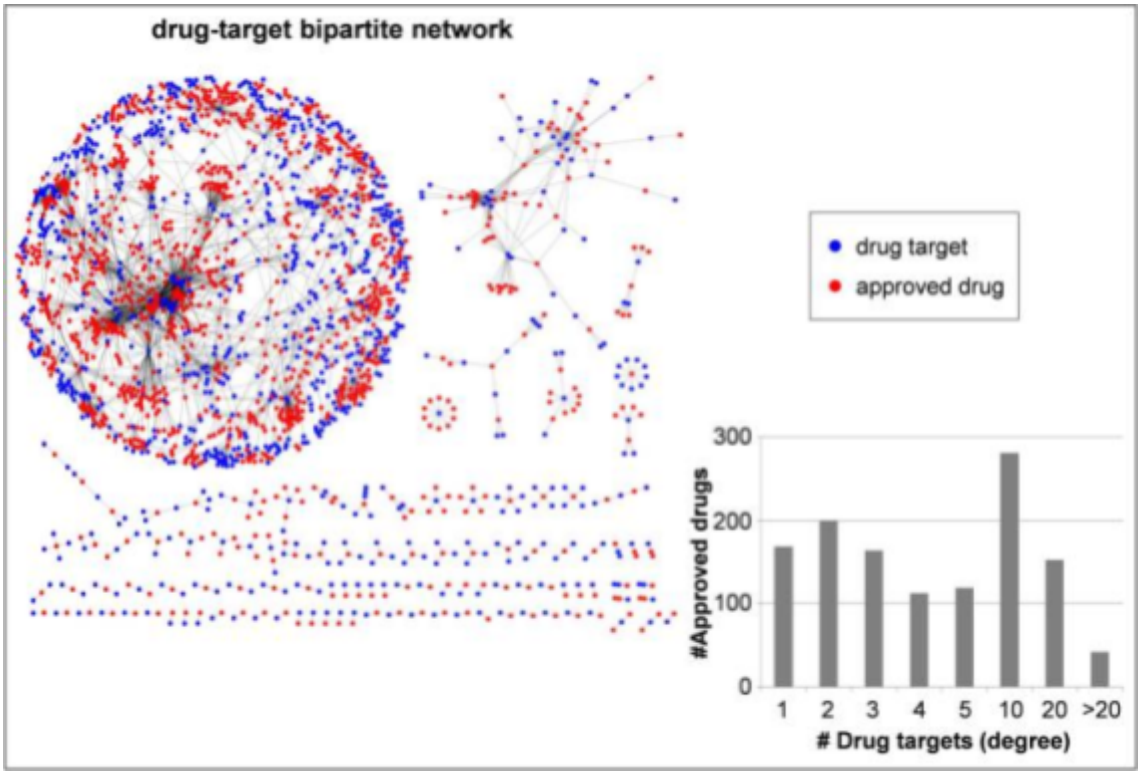



**Figure 1.4 - The drug-target landscape**. Red nodes in the bipartite network are approved drugs from DrugBank 3.0, blue nodes are drug targets with edges representing known interactions. The distribution of the degree of approved drugs is shown in the inset bar plot. Adapted from Hu et al. (2014).

The HTS model primarily concerns itself with identifying a compound with bioactivity against a single target. However, compounds are themselves promiscuous and increasingly have been shown to interact with a multitude of proteins. Drug-Target networks show that over 50% of approved drugs have more than 5 targets, whereas only 15% have one *known* target (see Fig. 1.4, it remains to be seen whether this number will decrease as more data becomes available) [39,40]. Drug-interaction data has

been collected in a target-centric manner, leading to incomplete data about the full complement of targets of any drug. One possible explanation for this is that there is little incentive to investigate data about these additional targets if they are not implicated in your disease area of interest. A second, more substantial explanation originates in the design of these HTS experiments, where only small panels of targets are actually screened against. Publicly available resources such as Drugbank, PubChem and PharmGKB seek to catalog drug interaction data, thereby allowing a more complete picture of this landscape to be formed [41–43]. However much data is proprietary and therefore remains undisclosed implying there still exists a data "missingness" problem. Historically, drug companies have little incentive to release data that may aid their competitors in a febrile market, however in recent years, companies have started being more open with their data realising that this will benefit their business as well as patients [44]. Fully exploring the range of drug-target interactions may lead to the repurposing of many drugs that are currently underutilised.

### 1.1.2 The case for drug repurposing

Drug repurposing is the use of existing drugs for new indications. There are numerous different definitions representing slightly different objectives in drug repurposing. For example, drug repurposing is the identification and development of new indications for abandoned compounds, whereas repositioning focuses on approved drugs only [45]. However generally (and within this document unless explicitly stated) these terms are used interchangeably and reflect the broader definition given and could equally apply to new indications for old drugs or additional indications for compounds in development. Drug repurposing presents an opportunity to de-risk drug discovery. The cost of research and development for bringing a drug to market increased exponentially for much of the 20th century and into the 2010s (Fig. 1.1a), in a trend known as Eroom's law (a reference to Moore's law in computing) [46,47].

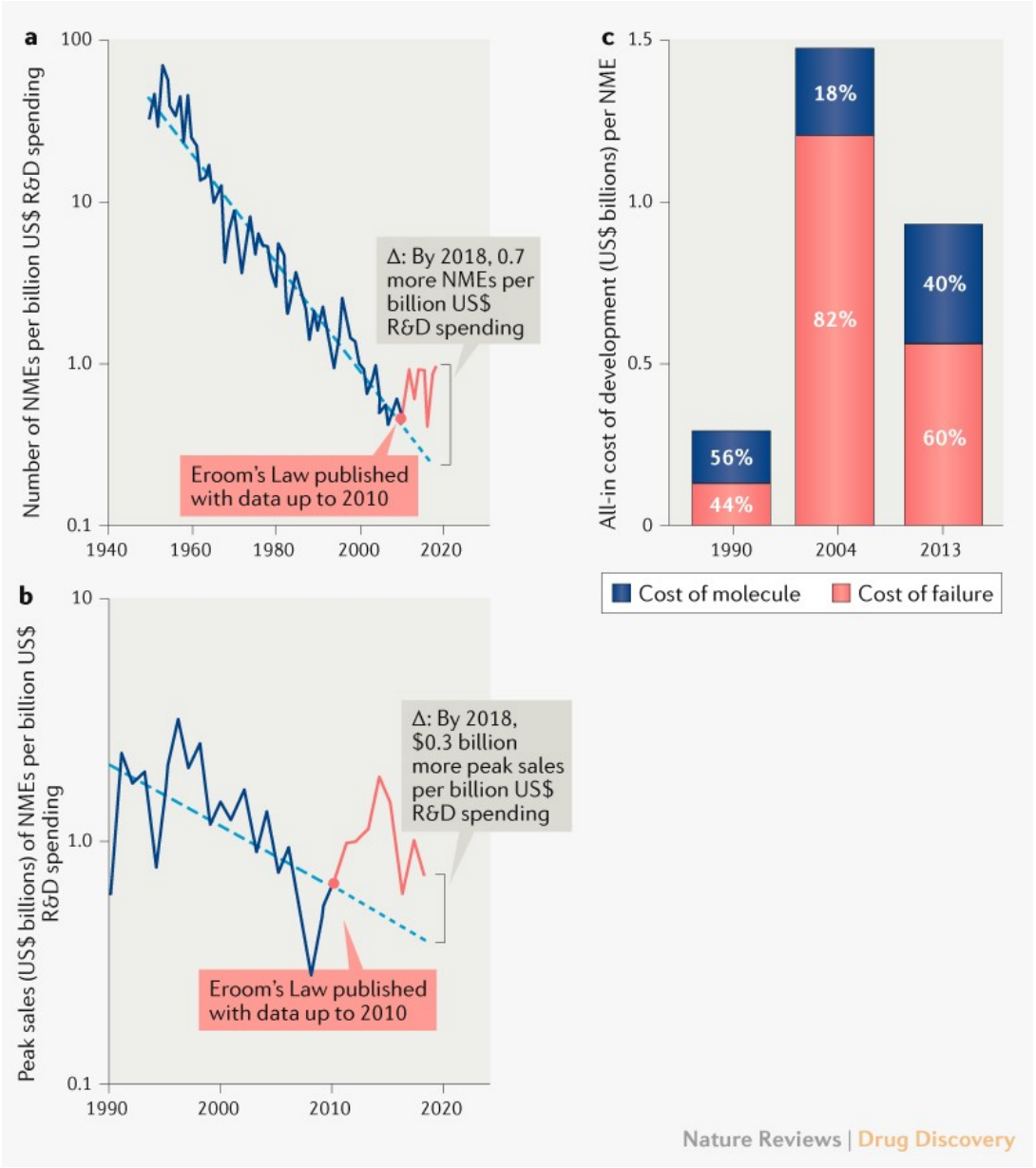



**Figure 1.5 - Breaking Eroom's law.** a) Count of new molecular entities (NMEs) per billion $US R&D spending. b) Aggregate peak sales ($US billions) of NMEs per billion $US R&D spending. c) All-in cost of development ($US billions) per NME. Copied, with permission, from Ringel et al. 2020 [47].

Whilst the number of drugs achieving approval each year remained relatively flat, the increase in expenditure to develop them increased by several orders of magnitude. This process was thought to be driven by four main factors [46]. The 'better than the beatles' problem stems from drugs competing within the same disease area only achieving marginal benefits over each other. In order to prove its worth to regulators, a

drug would have to demonstrate higher efficacy (or better side effect profiles) than a competitor. However, this causes an issue of power, as effect sizes are reduced when comparing drug vs drug instead of vs placebo, resulting in a need for larger sample sizes. The ‘cautious regulator’ problem reflects the fact that drugs have had to pass ever more stringent pre-clinical safety evaluations in various animal models, and increased scrutiny of phase 1 safety trials. This increase in regulatory burden followed several instances of unforeseen side effects in approved drugs. The most famous of these cases occurred in the 1960s with Thalidomide, in which pregnant women using the drug bore children with severe teratogenic side effects. The ‘basic research-brute force’ bias mirrors the rise of high-throughput approaches to drug discovery, in which the screening of compound libraries against targets became a primary focus. This approach, whilst offering the opportunity to test orders of magnitude more compounds, resulted in lower success rates [46–48]. Other possible factors include the idea that most of the easy to develop drugs for ‘good’ targets have already been developed, and therefore new indications must be found in diseases with complex aetiologies [48]. Intellectual property laws also present a challenge to drug companies. With the average drug taking 12-16 years to arrive to market, this leaves little time for a company to recoup its costs and profit from any drug before generics and biosimilars can be produced by competitors. This is compounded by the fact that for every 10,000 compounds evaluated for an indication, only 1 will make it to market. This extraordinary rate of attrition and limited market exclusivity presents a grave financial challenge for drug companies who must price drugs at a level that makes such expensive research profitable. This cost is in turn passed on to national health providers or private consumers, thereby contributing to the ever increasing cost of healthcare delivery.

The turn-around that has been witnessed (Fig. 1.5B+C) in the past decade and some of the factors that have contributed to it will be discussed later in the introduction.

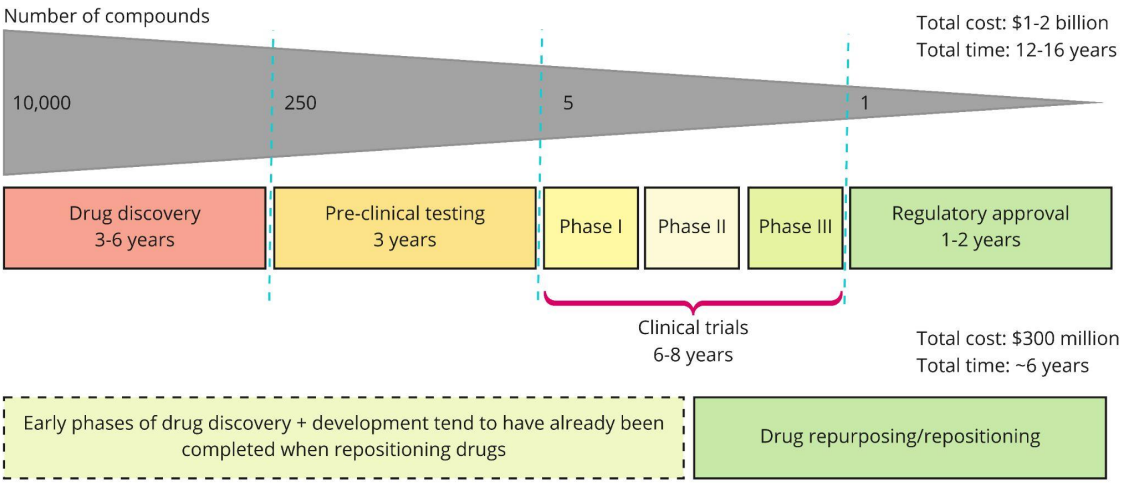



**Figure 1.6 - Comparing traditional drug discovery pipelines to drug repurposing.** This infographic timeline highlights the shorter timescales and reduced costs involved in bringing a repositioned drug to market when compared to a new molecular entity. Adapted from Nosengo (2016).

With the preceding factors in mind, the central value of drug-repurposing can be summarised as the de-risking of drug discovery. Firstly, the cost of repurposing is substantially lower than that of novel drug development (Fig. 1.6). This is because repositioned drugs have already been through a round of successful drug development. Thus, much more information is known about the drug, such as pharmacokinetic and pharmacodynamic profiles and the safety profile, both in a clinical trial setting, and in a real-world, polypharmacy setting. This information allows for the design of more specific trials with better patient and end-point selection. Additionally, the repurposed drug may only have to repeat efficacy trials to display efficacy in the new indication. This means that the drug development pipeline is reduced to approximately 6 years, at a cost of only ~$300 million [49]. It must be stated that this is true in the instance of repositioning a drug, but, to draw on the more formal definitions previously mentioned, repositioned drugs are drugs that have previously been approved in other indications. The same is not true of repurposed drugs, that are generally understood to have failed in clinical development. In this case, evidence suggests that success rates are much lower, with one study showing that of 667 failed drugs, only 10-16% were tested in another indication with a success rate of around 9% [50]. However, the reasons for why only this small percentage of drugs were tested in other areas are difficult to divine. Reasons for failure may be key, failure driven by the drug safety profile would be more difficult to overcome than limited therapeutic efficacy. In an ideal case a “failed drug” might be safe and highly potent on target, but the target

could be of marginal importance to the drug indication. Thus such failures of target validation could represent excellent repurposing opportunities. But as discussed earlier, access to data on early development failure is generally limited. A major consideration is that companies are generally unwilling to share data on proprietary compounds, and this limits the analysis of such compounds to the therapeutic areas of interest of the company. It is probable that other successes would arise if more systematic approaches to repurposing were regularly employed.

### 1.1.2.1 Examples of drug repurposing

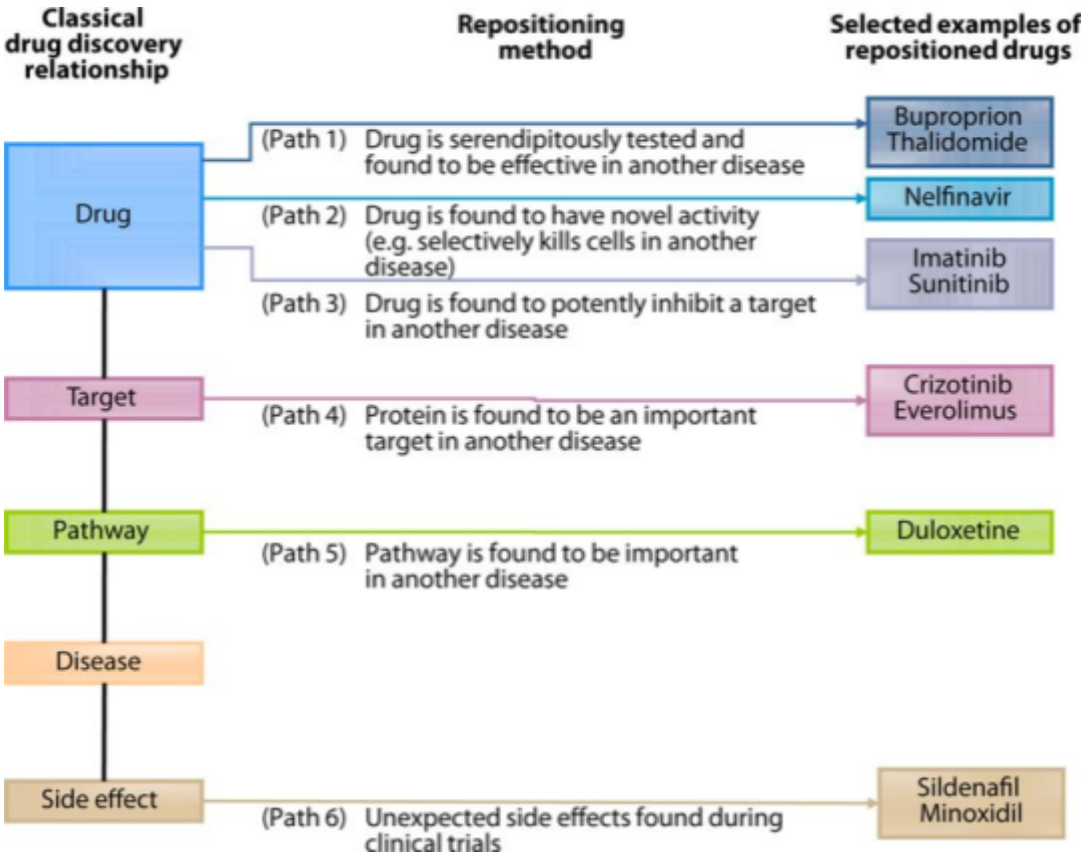



**Figure 1.7 - Methods and examples of drug repurposing.** Most repositioned drugs so far have been discovered through serendipitous treatment or unexpected side effects observed during clinical trials (path 1, path 6). More rational approaches to the identification of drug repositioning candidates involve finding existing drugs that can modulate specific disease phenotypes (path 2), finding new drug-target interactions (path 3), finding new roles for existing targets (path 4), or finding new pathways in disease (path 5). Adapted from Li & Jones (2012)

There exist numerous successful examples of drug repurposing, the most famous of which are Sildenafil (Viagra) and Minoxidil (Regaine) (see Fig. 1.7). Both of these drugs

were initially aimed to target hypertension, but presented very marketable side effects, treating erectile dysfunction and hair loss respectively. These are both examples of serendipitous findings rather than any systematic approach. Other such instances include of the aforementioned Thalidomide, which is now used to treat leprosy and multiple myeloma, and Imatinib, a blockbuster Chronic Myeloid Leukemia drug targeting the BCR-ABL fusion protein which has also been shown to inhibit key driver proteins in gastrointestinal stromal tumours, and finally metformin, a diabetes medication now being investigated in over 100 cancer based clinical trials. As such successes have been uncovered, a need to systematically uncover more opportunities has arisen. This can occur at various stages of a drug's lifespan. Early stage repurposing, in which multiple indications are discovered prior to market approval allow for the maximisation of a drug's target market and protect against the risk of large, late-stage trial failures. Repurposing of approved drugs is also a method pharmaceutical companies use to extend patent life on their blockbuster drugs (although this is not possible in all markets). For example, the biologic Humira (AbbVie) has been repurposed several times (from rheumatoid arthritis to ulcerative colitis and Crohn's disease amongst others) to extend patent life in the US. Other repurposing approaches include the screening of a drug against targets implicated in other diseases, for example Nelfinavir, an anti-HIV drug was screened against cancer cell lines and displayed potent inhibition of proliferation [51]. To date, 31 clinical trials have been registered investigating the efficacy of Nelfinavir in numerous cancer settings. Other simple approaches involve identifying whether the target protein is implicated in other diseases, as was the case with the immunosuppressant Everolimus, an mTOR inhibitor repurposed to treat Pancreatic neuroendocrine tumors driven by aberrant mTOR signalling [52].

#### 1.1.2.2 Drug repurposing using guilt-by-association (GBA)

As discussed, there are a myriad of approaches to *in silico* drug repurposing of varying complexity. These range from machine learning and network based inference, to compound based sub-structure searches [53–59]. One of the most simple, yet productive approaches is GBA [53,60–62].

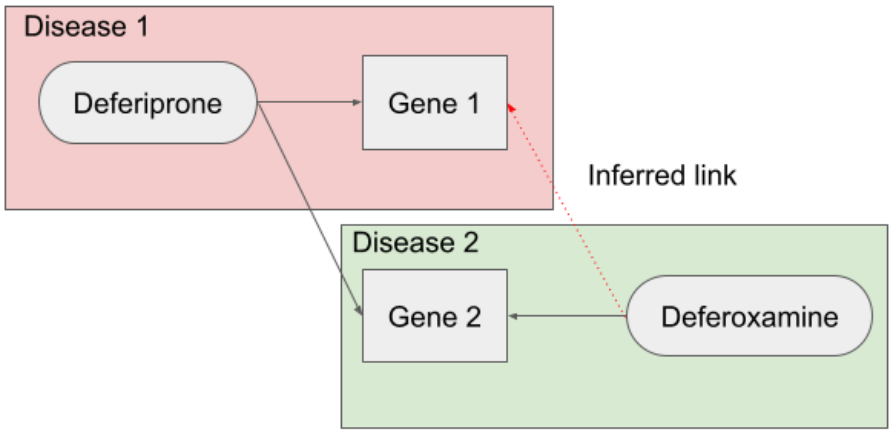



**Figure 1.8 - A diagrammatic representation of the GBA principle.** A novel gene-drug link is inferred by making the assumption that drugs that share gene targets are likely to share targets that may not have been associated with one of the drug pair.

This approach is predicated on the incompleteness of biological data, and simply aims to fill in the gaps. Simply put, if two diseases share a therapeutic, then it is possible that a therapeutic that is known to modulate disease 1 may also modulate disease 2 (see Fig. 1.8) [60]. In a fully observed model, we would already know that these potential links exist, however in reality, research groups and companies are focussed on their domain of interest, and (reasonably) would not seek to find a drug's full complement of interaction partners beyond their targets of interest. Furthermore, it is important to remember that many FDA-approved drugs were developed without knowledge of their mechanism of action, further compounding this lack of knowledge that would be required to identify alternate targets [35]. The GBA concept will be revisited in chapter 5, where examples relating to our work can be found.

### 1.1.2.3 Drug repurposing in the context of rare disease

Within the EU, rare diseases are defined as diseases affecting fewer than 1 in 2,000 people (the U.S. defines it as diseases affecting fewer than 200,000 people), and it is estimated that between 7-10% of the population will suffer from a rare disease in their lifetime. This translates to roughly 30 million people across Europe [63]. Currently there are roughly 10,000 rare diseases, with around 260 new ones being characterised each year [64,65]. As many as 80% of rare diseases may be genetic in cause, and therefore sequencing of affected individuals and their close family members offers the opportunity to identify new causal variants, and possibly an avenue to treatment [66].

However, this oft cited figure of 80% is contentious, as there is no clear analysis to point to as the source of this number, with other data suggesting the number may actually be as low as 40% [67]. Of the > 10,000 known rare diseases, only ~400 have licensed treatments, representing a significant area of unmet need [68].

Despite the number of people who will be affected by a rare disease being high, the small number of patients within any one disease group presents a challenge in finding treatments. This, compounded with the unclear pathology of many diseases makes drug development in this area uneconomical. The drugs targeting rare diseases (orphan drugs) that have been produced in recent decades have come at a prohibitive cost for health services and patients [69]. For example, the recently FDA approved spinal muscular atrophy medication Nusinersen (Spinraza, Biogen) is being marketed at a cost of $750,000 for the first year and $375,000 per year subsequently, for the lifetime of the patient. Further examples of high expense costs targeting rare diseases can be seen in table 1.1. Orphan drug sales account for roughly 16% of all non-generic prescription drug sales, with mean costs per patient per year for orphan drugs totalling 4.8x more than those for non-orphan drugs [67]. This places an undue burden on healthcare systems and is clearly unsustainable. Drug repurposing offers a pragmatic solution to this issue due to the aforementioned reduction in development costs.

| Drug | Disorder | Affected population | Estimated price ($US) | Manufacturer |
|---|---|---|---|---|
| Eculizumab (Soliris) | Paroxysmal nocturnal haemoglobinuria; atypical haemolytic-uremic syndrome | 2000 | 409,500 | Alexion Pharmaceuticals |
| Idursulfase (Elaprase) | Mucopolysaccharidosis II | 2000 | 375,000 | Shire |
| Galsulfase (Naglazyme) | Mucopolysaccharidosis VI | 1100 | 365,000 | BioMarin Pharmaceuticals |
| Alglucosidase alpha (Myozyme) | Pompe disease | 900 | 300,000 | Genzyme, BioMarin Pharmaceuticals |
| Rilonacept (Arkalyst) | Muckle-Wells disease | 2000 | 250,000 | Regeneron |

**Table 1.1 - The most expensive drugs.** Affected population sizes are estimates and drug names are followed by brand names in parentheses. Adapted from Luzzatto et al. (2018).

The problem of finding treatments for rare diseases is further compounded by the need for large clinical trials to prove efficacy and safety within patient populations that may be fewer than 100 people. Recognising this, regulatory bodies will accept trials involving fewer people for compounds that have passed safety trials in other indications. For example, repurposing of the approved contraceptive drug mifepristone for Cushing's Syndrome was possible with a trial cohort of only 30 patients. For comparison, a new chemical entity being tested for the same indication required more than 90 patients due to the requirement for both safety and efficacy to be demonstrated [70].

## 1.1.3 Using human genetics to inform drug discovery

In addition to sequencing allowing for the diagnosis of genetic disorders, human genetics also contributes to various elements of the drug development process. As

previously discussed, there is a high attrition rate within drug discovery, with compounds failing at numerous phases of development. However, despite extensive preclinical research, fewer than 5% make it to phase 1 clinical trials reach the market, of which at least 50% fail due to lack of efficacy [5,7,8,46,71]. Human genetics allows for a reduction in the reliance on *in vitro* and *in vivo* animal model data, which evidently have limited predictive value [4,12,72–77]. This is in part due to the fact that preclinical studies are effective in demonstrating that a given target is perturbed by a compound, but are less able to assess whether the target is a causal disease gene in the human context (which is especially true in complex diseases). Despite this, the power of these preclinical models has been in their ability to interrogate gene function through the disruption of their function at a systems or organism level, an approach which clearly cannot be attempted in humans. Thankfully, large scale exome and genome sequencing projects have revealed variants across the genome that may aid in illuminating hitherto unknown functions of genes.

Whilst we have spent much of the chapter discussing reasons for the increase in drug development cost highlighted in Eroom's law. We have not addressed the increase in success visible from the 2010 through to 2020 (Fig. 1.5 A-C). Such increases in the number of drugs produced per billion $US, whilst multifactoral in origin, have been partly attributed to the inclusion of genetic evidence in the process of drug development [78–80]. The growth of genetic association studies such as the genome-wide association study (GWAS) has provided a better route to interrogating disease biology. This has led to a rapid increase in what is understood about diseases and the genes that modulate them [81]. This evidence in turn provides a solid scientific foundation on which to select targets for drug discovery [71,78], helping to reduce the steep attrition rate in clinical development associated with lack of efficacy [48,82,83]. Analysis of success rates in drug development show that drugs with genetic evidence are twice as likely to obtain market approval than those without such evidence [78,79]. Databases such as the GWAS catalog (MacArthur et al. 2017) and the Online Mendelian Inheritance in Man (OMIM) amongst others, provide valuable repositories of information; information which help to build evidence for the targeting of specific genes and their products. This increased use of genetic evidence will hopefully perpetuate the improvement in drug discovery efficiency that we are currently witnessing.

The utility of human genetics in aiding drug discovery is not limited to the target identification stage. Genomics also aids in the diagnosis of disease, clinical trial recruitment, and in the proper assignment of treatments, otherwise referred to as

personalised medicine. A key challenge in providing drugs for rare disease patients, is not having an understanding of the genes driving their diseases, or of having a clear diagnosis of their disease at all.

#### 1.1.3.1 Loss of function variation

Loss of function (LoF) variants are variants that result in the aberration of the protein function. The causes for this LoF can be wide-ranging, with the possibility for variants to affect non-coding regulatory regions, or disrupt secondary, tertiary or quaternary protein structure through missense mutation [84]. However within this thesis we will narrow the definition to variants resulting in the substantial truncation of protein-coding transcript, as is standard in the literature [84–91]. These protein truncating variants can be categorised as either **stop-gained** variants, in which a non-synonymous variant leads to a premature stop-codon; **essential splice site** variants in which splicing of the transcript fails due to disruption of splice donor or acceptor sites; and **frameshift variants** caused by insertions or deletions (indels) resulting in a change to the reading frame (Fig. 1.9). LoF variants typically result in negative phenotypes, which places them under heavy negative selection. This is relatively intuitive, as a lack of phenotype resulting from the partial or complete knockout of a gene implies that the gene itself is not necessary for survival. This means that LoF variants should only be seen at very low allele frequencies [84]. This has borne to be true, however a surprising number of LoF variants have been identified when sequencing human populations, even in healthy individuals [89,92].

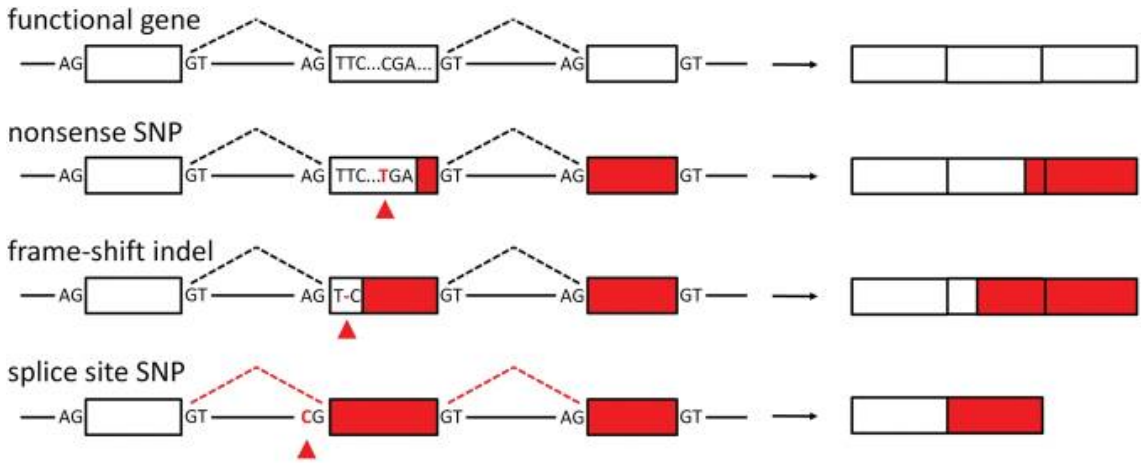



**Figure 1.9 - Diagram of LoF causing variants.** The first row shows an intact three-exon gene, with each subsequent row highlighting a protein truncating variant (in red, indicated with red triangles). The resulting effect is shown on the right, with red boxes indicating lost protein-coding functionality. Adapted from Macarthur & Tyler-Smith. 2010.

These appear in heterozygous, and more rarely, homozygous forms. Homozygous knockouts are especially interesting for their ability to highlight the phenotypic consequences of knocking out human genes within humans, and therefore serve as a model of life-long inhibition of a gene [93]. The possible implications of such information can be hugely valuable. Such a case exists with PCSK9, a protein responsible for increasing serum low-density-lipoprotein (LDL) serum levels, sometimes leading to an increased cardiovascular disease risk. Study of this gene uncovered both gain-of-function mutants with a commensurate increase in both LDL serum levels and cardiovascular risk, and LoF variants resulting in the opposite effect [18,94]. This indicates a causative relationship that can be modulated using a targeted therapy [71]. However, this case also illustrates a second useful indicator in the context of drug discovery. An individual was discovered who was a compound heterozygote for two LoF variants in PCSK9, with no discernable negative associated phenotype [95]. This indicates that inhibition of the target is not only likely to be effective, but also safe, thereby alleviating toxicity concerns [96]. Many other studies have since begun to uncover the surprisingly high number of total knockouts that exist within the human population. Study of outbred populations revealed 1775 genes with homozygous predicted LoF (pLoF) genotypes [85], and study of bottleneck or founder populations in Iceland and Finland revealed enrichment for homozygous pLoF individuals [86,91]. Despite accumulating relatively large sample sizes, only pLoF variants of moderate allele frequency could be discovered and

a true map of the human LoF landscape, incorporating variants of significant rarity, would be difficult to draw without rarer variants [92]. Indeed, for 38% of genes, not finding a homozygous LoF genotype within all the outbred individuals on the planet would still not provide statistical evidence that a homozygous LoF genotype will not be tolerated [93]. However, study of consanguineous populations increases the probability of identifying these rare variants in a homozygous state. This is due to the increased proportion of runs of autozygosity within the genome, i.e. regions of chromosomal identity stemming from a recent common ancestor [92]. Autozygous regions containing a pLoF will necessarily be homozygous for this variant. The Born in Bradford cohort (BiB) represents such a population, comprising 3,222 British-Pakistanis. Upon sequencing of this population, 781 protein-coding genes were identified as complete knock-outs. Significantly, this work took place in healthy individuals, and no correlation was observed between the presence of null LoF genotypes and engagement with health services [92]. Further efforts in consanguineous populations have continued to reveal more homozygous LoF genes in adult populations including the East London Genes and Health (ELGH), an expansion of the BiB cohort, and the Pakistan Risk of Myocardial Infarction Study (PROMIS) cohort [87]. ELGH aims to sequence 100,000 people of Bangladeshi and Pakistani origin living in East London, and link this data with health records (as was done with the BiB cohort). Similarly, the PROMIS study is sequencing 30,000 Pakistani's (50% who have suffered myocardial infarction, and 50% controls) living in Pakistan, with further demographic and environmental questionnaires [87]. Such studies are addressing an imbalance in genetic data available to researchers, whereby the vast majority of genetic studies have occurred in European populations [97–99]. Furthermore, as previously mentioned these populations are enriched for the offspring of consanguineous couplings, leading to an enrichment for rare homozygous LoF variants.

The use of LoF variation in drug discovery will be covered in chapters 2, 3 and 4.

### 1.1.3.2 Genetic datasets relevant to this thesis

In order for human genetics data to be widely used, they need to be aggregated and made available for research. Two large scale collections of genomic data, the genome aggregation database, and the 100,000 genomes project are of particular relevance in this thesis, and therefore will be briefly introduced here.

#### 1.1.3.2.1 The Genome Aggregation Database (gnomAD)

GnomAD (v2.1) is a collection of genomic and exonic data covering 141,456 unrelated individuals. The project is managed out of the Broad Institute of MIT and Harvard, but reflects the work of a global assortment of principal investigators and their groups. Summary statistics of the data have been made readily available through the gnomAD browser, which is accessed by researchers across the world.

The generation of these data required the collection of raw data from each of the contributing projects, followed by sample quality control (QC), joint-calling of the variants and site QC. Then statistics such as allele frequencies and gene constraint were calculated. In order to make the statistics more generalisable, data from closely related individuals and individuals known to suffer from severe pediatric diseases were removed. This dataset features heavily in chapters 2, 3 and 4, with data on constraint playing a key role in the analyses therein.

1.1.3.2.2 The 100,000 Genomes Project

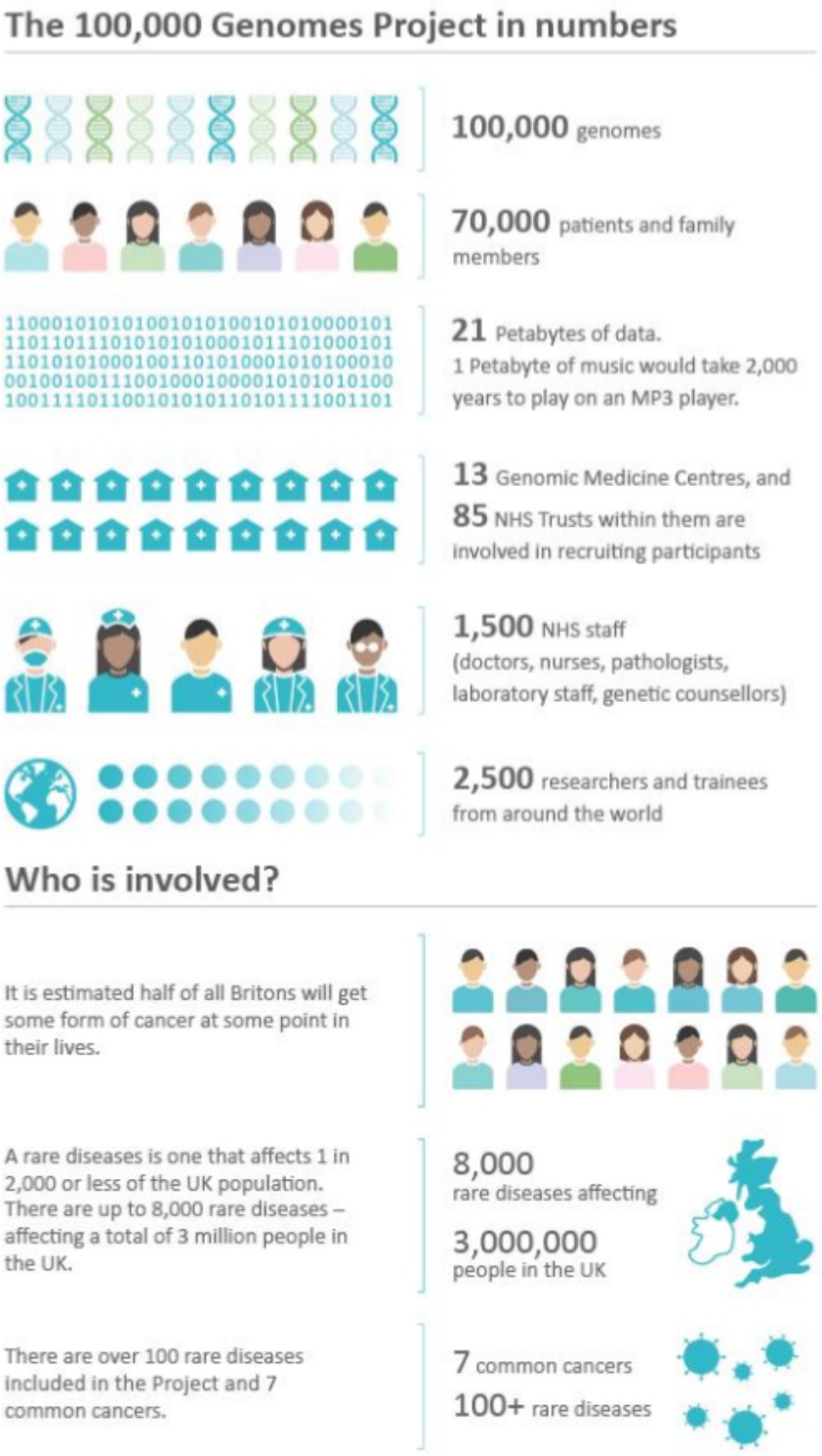



**Figure 1.10 - An infographic overview of the 100KGP.**

The 100KGP is a flagship genomics project launched by David Cameron in 2012. Genomics England (GEL), a subsidiary of the Department of Health and Social Care, was set up to deliver this project which focussed on rare diseases, cancers and infectious diseases [100]. The central aim of the 100KGP was to combine genomic data, along with medical records and other data sources to uncover the drivers of these diseases, provide diagnoses to individuals who as of yet did not have one, and to aid in the personalisation of healthcare.
The 100KGP finished sequencing the 100,000 genomes across ~75,000 individuals (cancer patients supplied two samples, one cancer genome and one of healthy tissue, see Fig. 1.10) in October of 2018. With genomics coming to the fore in scientific research, the 100KGP aimed to kickstart the development and use of genomic medicine in the NHS, an organisation that already contains a wealth of various forms of data about patients. This ambition was realised with the creation of the NHS genomic medicine service in December 2018.

The ‘diagnosis odyssey’ is a term used to describe the journey rare disease patients must navigate through the healthcare sector before receiving a diagnosis and the hope of treatment. This journey takes on average 4-8 years, with the involvement of multiple specialists, and a battery of invasive tests [101]. During this time, the patient will likely be receiving suboptimal care, despite involvement of healthcare services, simply because not enough is known about their diseases. The 100KGP recruited such patients, using WGS to remove the reliance on gene-panel tests focussing on only a few genes at a time that are commonly used in such difficult to solve cases [102,103]. Of this group, GEL has returned actionable findings (those that result in an impact in clinical care) for 20-25% of participants. These patient genomes, along with unaffected relatives and healthcare records are now available for researchers to use freely within a trusted research environment, and therefore the hope remains that with time, more clinically actionable findings will be uncovered.

Within the course of this thesis, any reference made to GEL data will be synonymous with 100KGP data.

## 1.1.4 An introduction to networks

Network science is a recently emerging domain which lends itself to study of biological systems and their complex interactions. Networks describe the relationship between entities, for example a protein-protein interaction network describes the interaction between proteins. Many other types of biological networks exist such as metabolic, transcriptional regulatory and cell signalling networks. In networks, the entities are termed nodes, and their relationships (which can be either direct or indirect) are edges. More than one type of entity can be represented in a network, for example a drug-target interaction network features interactions between drugs and proteins in a bipartite network.

Networks appear at various points through this thesis, so we introduce basic concepts and examples of their use in biology here.

### 1.1.4.1 A brief history of network science

Graph theory, the mathematical representation of networks, first originated in the 18th century with Euler and the 'Seven bridge of Konigsberg' problem. Konigsberg was the capital of Prussia, and had seven bridges crossing the river Pregel. The particular configuration of bridges led to the question of whether it was possible to traverse each of the seven bridges once, whilst never crossing any one more than once. Euler negatively solved (i.e. proved it was not possible) this problem using the first example of graph theory. The important observation resulting from this work is that graphical representation results in a simplification of the problem, and that there are inherent properties in the graph that dictate their behaviour. In other words, it doesn't matter how hard you try, there is no way of (positively) solving this problem [104]. Little occurred in the development of graph theory in the following years, until the study of real world networks in the mid-20th century. Real world networks often appear random, and therefore early network science attempted to establish the random properties of these networks. Early work in the field of the random network model includes that of Erdos and Renyi. They examined the mathematical consequences of throwing a random number of buttons on the floor and then randomly connecting them with a random number of links. They found that resulting networks will have a Poisson distribution with

small clustering coefficients (measures of how densely interconnected nodes neighbours are). Clustering coefficients approaching zero indicate that none of the nodes connected to a 'subject' node are connected to each other. This class of random networks were termed 'small world' networks, as the average distance between any two nodes is small (an extension of this is the six-degrees of separation theory, that all people are connected by a relatively small number of acquaintances). Further work on small world networks by Watts and Strogatz in 1998 sought to further investigate random networks. Their contribution (greatly summarised) solved a limitation of Erdos-Renyi graphs that meant that triadic closures did not occur (e.g. in a network, if A and B are connected, then C can only be connected to one or the other). Examination of simple real world networks such as the power grid show that such triadic closures are in fact common occurrences. With this came the understanding that networks are less random than initially thought, and therefore do not follow such distributions. This led to the work of Barabasi and Albert (1999) which describes scale-free networks, networks in which the degree distribution (the number of neighbours a node has) follows a power-law [105]. Most observed real-world networks are scale-free (i.e. have power-law distributions) including the world wide web, citation networks, and protein regulatory networks [106–108]. The hypothesised reason for the formation of networks of this class are, that as networks grow, new entities preferentially attach to existing entities with a high degree in a process often dubbed 'the rich get richer'.

Network science aims to characterise complex systems. Systems are described as such because of the challenging aspect of trying to divine collective behaviour of entities even *with* knowledge of each constituent entity [104]. Many of the methods currently used in analysing biological networks stem from work on other complex networks. Here I will describe some approaches to analysing networks that whilst originating in other domains, have found application in biological networks.

Network theory allows us to explore which nodes are key to resilience of the local or global network, a property referred to as its robustness. Small networks, such as a network composed of the components of a truck, may no longer function at all if a component such as the engine or spark plugs fail. However in general, real world networks are surprisingly resilient to component failures; the breakdown of a single truck would not stop the whole delivery company from functioning. This notion of robustness was first explored in the context of the internet [109], where the interconnectedness of nodes (the average shortest path between any two nodes, also

called the network diameter) was studied before and after the removal of fractions of nodes. Scale-free networks, by virtue of few nodes having very many connections, are resilient as very few nodes are able to drastically alter the interconnectedness of the network to any substantial degree. As the probability of removing a node with many connections is low, even the failure of 5% of nodes was shown to not result in communication between the nodes being affected. Conversely, the removal of the top 5% of the most connected nodes resulted in a doubling of the network diameter. It is important to note that this study also showed that the type of network is critical to the observed robustness of the network [106]. That is to say, networks following an Erdos-Renyi model (networks in which edges are randomly assigned to nodes without preferential attachment to highly connected nodes) display much less resilience to random removal of nodes, as each removal of a node represents a higher probability of removing more edges, whereas they exhibit more resilience to targeted attacks in which the highest degree nodes are removed. Such properties have important consequences when considering biological networks as it will greatly impact your targeting strategy. It asks the question of whether it is better to target a single point of weakness in a pathway, or instead better to target as much of the pathway as you can.

Another feature of importance within networks is the formation of communities, groups in which nodes are more likely to connect to each other than they are to other nodes. Community detection aims to partition the graph based on some notion of an inherent community structure denoted by the graph topology. Whilst this alludes to an objective truth of community organisation, the reality is that communities can be defined in a range of ways depending on what restrictions you place upon the nature of the community. For example, limiting to ‘strong’ communities, in which all nodes interact with all other nodes, may be overly restrictive, especially if you cannot be sure that you have complete information for a network. In this instance, you may choose to identify ‘weak’ communities - subgraphs in which the total degree within the subgraph is greater than the total degree of connections from nodes within the subgraph to nodes outside the subgraph. However this can increase the noise in your network. Further to this, the computational power required to brute force a search of possible communities is astronomical due to this being an NP-complete problem. Therefore, various algorithmic approaches have been developed to approximate underlying community structure. The main approaches include both agglomerative and divisive hierarchical clustering, and modularity maximising approaches. All approaches have drawbacks that will be discussed at the point at which they become relevant within the context of

this thesis. Further information relating to the topics covered here can be found in Network Science by Albert Barabasi [104].

### 1.1.4.2 Biological networks

Since the completion of the human genome project and the birth of the era of genomics, the classical view of protein function as being the action of a protein on another single protein to perform a discrete action has evolved. The understanding of a protein's function cannot be achieved without exploring its wider network as proteins function cooperatively with other entities, and seldom in isolation (Fig 1.11) [110–112]. This holds true for other biological systems and components such as the genome, transcriptome, and metabolome.

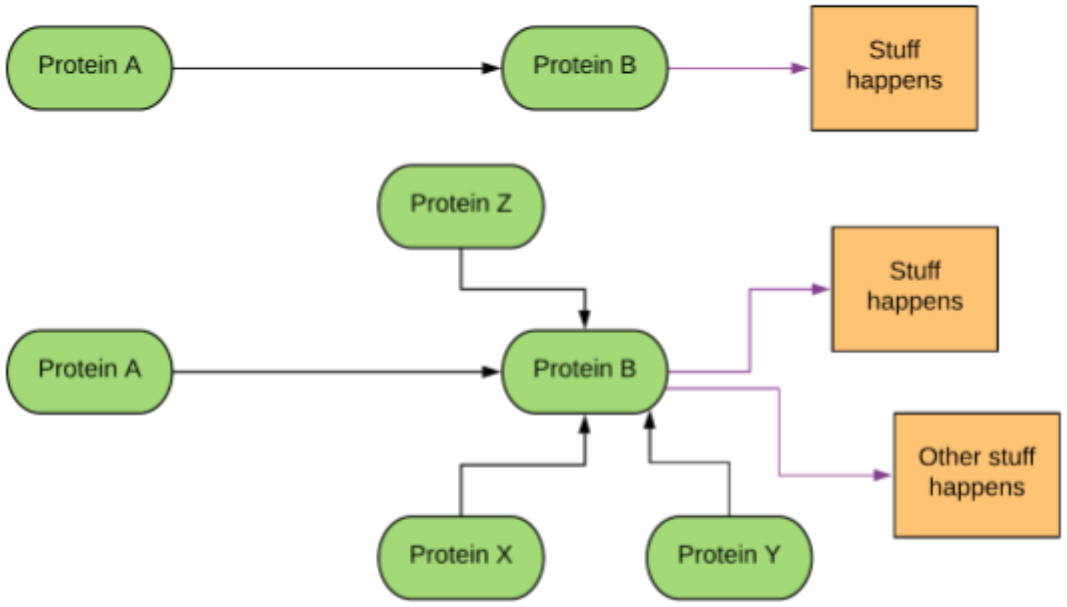



**Figure 1.11 - A schematic of protein interactions,** comparing a historic, linear view of protein interaction (top) to an updated, network view of protein interaction below. Protein action is often contextual and can be influenced by numerous other proteins.

Biological networks have grown in popularity as the scale of biological data has grown across all domains. This is because graph theory provides a common language with which to describe the complex interactions observed. This allows for the integration of data across different biological levels, for example, gene-disease networks, in which information on multiple genes can be incorporated to identify the underlying causes of

the disease [113,114]. This is critical to understanding diseases that are frequently the result of the action of several genetic variants [115] and can highlight previously unknown shared genetic aetiologies of diseases [114]. Applications of this have uncovered aetiologies of rare diseases such as systemic sclerosis and congenital hyperinsulinism [116,117], predicted a further 128 disease-causing genes in a variety of rare diseases [118], and aided in areas of drug development ranging from target identification [119,120] to target validation [121] and even side effect prediction [122,123], amongst others.

Network theory allows us to explore which nodes are key to resilience of the local or global network. For example, the degree of a node has been shown to be positively correlated with the essentiality of the gene [124], or the role of the gene in disease such as cancer [125], with disease related proteins as a whole having on average 32% more connections than other proteins [114]. This also has important implications in drug-target identification, with networks highlighting unexplored opportunities made apparent by the joining of DrugBank data on drug-target interactions, and approved indications for medications [35,54].

## 1.2 Thesis aims

The aim of this PhD is to use large, publicly available genomic data from sources including with the 100KGP and gnomAD, combined with other public resources to identify targets for therapeutic intervention. We aim to use the genomic data to identify homozygous loss-of-function variants of neutral effect, and use features derived from wider data sources to predict more of this class of loss-of-function variant. Following this, we will assess the value of these variants in the prediction of drug development probability of success. Finally, we will aim to identify therapeutic targets specifically for the rare diseases studied by the 100KGP. We will highlight targets for which drug repurposing opportunities may exist, and facilitate research into these areas by creating a database with the data necessary to make such inferences.

# 1.3 References


1. Wouters, O. J., McKee, M. & Luyten, J. Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. *JAMA* **323**, 844–853 (2020).
2. DiMasi, J. A., Hansen, R. W. & Grabowski, H. G. The price of innovation: new estimates of drug development costs. *J. Health Econ.* **22**, 151–185 (2003).
3. DiMasi, J. A., Grabowski, H. G. & Hansen, R. W. Innovation in the pharmaceutical industry: New estimates of R&D costs. *J. Health Econ.* **47**, 20–33 (2016).
4. Hingorani, A. D. *et al.* Improving the odds of drug development success through human genomics: modelling study. *Sci. Rep.* **9**, 18911 (2019).
5. Hay, M., Thomas, D. W., Craighead, J. L., Economides, C. & Rosenthal, J. Clinical development success rates for investigational drugs. *Nat. Biotechnol.* **32**, 40–51 (2014).
6. Munos, B. Lessons from 60 years of pharmaceutical innovation. *Nat. Rev. Drug Discov.* **8**, 959–968 (2009).
7. Pammolli, F., Magazzini, L. & Riccaboni, M. The productivity crisis in pharmaceutical R&D. *Nat. Rev. Drug Discov.* **10**, 428–438 (2011).
8. Kola, I. & Landis, J. Can the pharmaceutical industry reduce attrition rates? *Nat. Rev. Drug Discov.* **3**, 711–715 (2004).
9. Hwang, T. J. *et al.* Failure of Investigational Drugs in Late-Stage Clinical Development and Publication of Trial Results. *JAMA Intern. Med.* **176**, 1826–1833 (2016).
10. Paul, S. M. *et al.* How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nat. Rev. Drug Discov.* **9**, 203–214 (2010).
11. Schenone, M., Dančík, V., Wagner, B. K. & Clemons, P. A. Target identification and mechanism of action in chemical biology and drug discovery. *Nat. Chem. Biol.* **9**, 232–240 (2013).

12. Van Norman, G. A. Limitations of Animal Studies for Predicting Toxicity in Clinical Trials: Is it Time to Rethink Our Current Approach? *JACC Basic Transl Sci* **4**, 845–854 (2019).
13. Denayer, T., Stöhr, T. & Van Roy, M. Animal models in translational medicine: Validation and prediction. *New Horizons in Translational Medicine* **2**, 5–11 (2014).
14. Pound, P. & Ritskes-Hoitinga, M. Is it possible to overcome issues of external validity in preclinical animal research? Why most animal models are bound to fail. *J. Transl. Med.* **16**, 304 (2018).
15. Metcalf, B. W. & Dillon, S. *Target Validation in Drug Discovery*. (Elsevier, 2011).
16. Moll, J. & Carotta, S. *Target Identification and Validation in Drug Discovery: Methods and Protocols*. (Springer New York, 2019).
17. Shapiro, M. D., Tavori, H. & Fazio, S. PCSK9: From Basic Science Discoveries to Clinical Trials. *Circ. Res.* **122**, 1420–1438 (2018).
18. Abifadel, M. *et al.* Mutations and polymorphisms in the proprotein convertase subtilisin kexin 9 (PCSK9) gene in cholesterol metabolism and disease. *Hum. Mutat.* **30**, 520–529 (2009).
19. Cohen, J. *et al.* Low LDL cholesterol in individuals of African descent resulting from frequent nonsense mutations in PCSK9. *Nat. Genet.* **37**, 161–165 (2005).
20. Eder, J., Sedrani, R. & Wiesmann, C. The discovery of first-in-class drugs: origins and evolution. *Nat. Rev. Drug Discov.* **13**, 577–587 (2014).
21. Szymański, P., Markowicz, M. & Mikiciuk-Olasik, E. Adaptation of high-throughput screening in drug discovery-toxicological screening tests. *Int. J. Mol. Sci.* **13**, 427–452 (2012).
22. Lahana, R. How many leads from HTS? *Drug Discov. Today* **4**, 447–448 (1999).
23. Lipinski, C. A. Drug-like properties and the causes of poor solubility and poor permeability. *J. Pharmacol. Toxicol. Methods* **44**, 235–249 (2000).
24. Duke, J., Friedlin, J. & Ryan, P. A quantitative analysis of adverse events and 'overwarning' in drug labeling. *Arch. Intern. Med.* **171**, 944–946 (2011).

25. DiBonaventura, M., Gabriel, S., Dupclay, L., Gupta, S. & Kim, E. A patient perspective of the impact of medication side effects on adherence: results of a cross-sectional nationwide survey of patients with schizophrenia. *BMC Psychiatry* **12**, 20 (2012).
26. Tedla, Y. G. & Bautista, L. E. Drug Side Effect Symptoms and Adherence to Antihypertensive Medication. *Am. J. Hypertens.* **29**, 772–779 (2016).
27. Formica, D. *et al.* The economic burden of preventable adverse drug reactions: a systematic review of observational studies. *Expert Opin. Drug Saf.* **17**, 681–695 (2018).
28. Pirmohamed, M. *et al.* Adverse drug reactions as cause of admission to hospital: prospective analysis of 18 820 patients. *BMJ* **329**, 15–19 (2004).
29. Batel-Marques, F., Penedones, A., Mendes, D. & Alves, C. A systematic review of observational studies evaluating costs of adverse drug reactions. *Clinicoecon. Outcomes Res.* **8**, 413–426 (2016).
30. Chyka, P. A. How many deaths occur annually from adverse drug reactions in the United States? *Am. J. Med.* **109**, 122–130 (2000).
31. Patel, T. K. & Patel, P. B. Mortality among patients due to adverse drug reactions that lead to hospitalization: a meta-analysis. *Eur. J. Clin. Pharmacol.* **74**, 819–832 (2018).
32. Schuster, D., Laggner, C. & Langer, T. Why drugs fail – A study on side effects in new chemical entities. *Antitargets* 1–22 (2008) doi:10.1002/9783527621460.ch1.
33. Veeren, J. C. & Weiss, M. Trends in emergency hospital admissions in England due to adverse drug reactions: 2008–2015. *J. Pharm. Health Serv. Res.* **8**, 5–11 (2016).
34. Mizushima, T. Drug discovery and development focusing on existing medicines: drug re-profiling strategy. *J. Biochem.* **149**, 499–505 (2011).
35. Cheng, F. *et al.* Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput. Biol.* **8**, e1002503 (2012).

36. Moore, T. J., Furberg, C. D., Mattison, D. R. & Cohen, M. R. Completeness of serious adverse drug event reports received by the US Food and Drug Administration in 2014. *Pharmacoepidemiol. Drug Saf.* **25**, 713–718 (2016).
37. Crépin, S., Villeneuve, C. & Merle, L. Quality of serious adverse events reporting to academic sponsors of clinical trials: far from optimal. *Pharmacoepidemiol. Drug Saf.* **25**, 719–724 (2016).
38. Gijsberts, C. M. *et al.* Race/Ethnic Differences in the Associations of the Framingham Risk Factors with Carotid IMT and Cardiovascular Events. *PLoS One* **10**, e0132321 (2015).
39. Hu, Y. & Bajorath, J. Monitoring drug promiscuity over time. *F1000Res.* **3**, 218 (2014).
40. Hu, Y., Jasial, S. & Bajorath, J. Promiscuity progression of bioactive compounds over time. *F1000Research* vol. 4 118 (2015).
41. Whirl-Carrillo, M. *et al.* Pharmacogenomics knowledge for personalized medicine. *Clin. Pharmacol. Ther.* **92**, 414–417 (2012).
42. Kim, S. *et al.* PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **47**, D1102–D1109 (2019).
43. Wishart, D. S. *et al.* DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* **46**, D1074–D1082 (2018).
44. Glicksberg, B. S., Li, L., Chen, R., Dudley, J. & Chen, B. Leveraging Big Data to Transform Drug Discovery. *Methods Mol. Biol.* **1939**, 91–118 (2019).
45. Doan, T. L., Pollastri, M., Walters, M. A. & Georg, G. I. Chapter 23 - The Future of Drug Repositioning: Old Drugs, New Opportunities. in *Annual Reports in Medicinal Chemistry* (ed. Macor, J. E.) vol. 46 385–401 (Academic Press, 2011).
46. Scannell, J. W., Blanckley, A., Boldon, H. & Warrington, B. Diagnosing the decline in pharmaceutical R&D efficiency. *Nat. Rev. Drug Discov.* **11**, 191–200 (2012).
47. Ringel, M. S., Scannell, J. W., Baedeker, M. & Schulze, U. Breaking Eroom's Law. *Nat. Rev. Drug Discov.* **19**, 833–834 (2020).

48. Scannell, J. W. & Bosley, J. When Quality Beats Quantity: Decision Theory, Drug Discovery, and the Reproducibility Crisis. *PLoS One* **11**, e0147215 (2016).
49. Nosengo, N. Can you teach old drugs new tricks? *Nature* **534**, 314–316 (2016).
50. Neuberger, A., Oraiopoulos, N. & Drakeman, D. L. Renovation as innovation: is repurposing the future of drug discovery research? *Drug Discov. Today* **24**, 1–3 (2019).
51. Gills, J. J. *et al.* Nelfinavir, A lead HIV protease inhibitor, is a broad-spectrum, anticancer agent that induces endoplasmic reticulum stress, autophagy, and apoptosis in vitro and in vivo. *Clin. Cancer Res.* **13**, 5183–5194 (2007).
52. Li, Y. Y. & Jones, S. J. Drug repositioning for personalized medicine. *Genome Med.* **4**, 27 (2012).
53. Li, Z.-C. *et al.* Identification of drug–target interaction from interactome network with 'guilt-by-association' principle and topology features. *Bioinformatics* **32**, 1057–1064 (2015).
54. Zheng, Y. *et al.* Old drug repositioning and new drug discovery through similarity learning from drug-target joint feature spaces. *BMC Bioinformatics* **20**, 605 (2019).
55. Lee, I. & Nam, H. Identification of drug-target interaction by a random walk with restart method on an interactome network. *BMC Bioinformatics* **19**, 208 (2018).
56. Janes, J. *et al.* The ReFRAME library as a comprehensive drug repurposing library and its application to the treatment of cryptosporidiosis. *Proc. Natl. Acad. Sci. U. S. A.* **115**, 10750–10755 (2018).
57. Kumar, S. & Kumar, S. Chapter 6 - Molecular Docking: A Structure-Based Approach for Drug Repurposing. in *In Silico Drug Design* (ed. Roy, K.) 161–189 (Academic Press, 2019).
58. Gl, B. *et al.* Structure-based drug repurposing to inhibit the DNA gyrase of Mycobacterium tuberculosis. *Biochem. J* **477**, 4167–4190 (2020).
59. Schuler, J. & Samudrala, R. Fingerprinting CANDO: Increased Accuracy with Structure- and Ligand-Based Shotgun Drug Repurposing. *ACS Omega* **4**,

17393–17403 (2019).

60. Chiang, A. P. & Butte, A. J. Systematic evaluation of drug-disease relationships to identify leads for novel drug uses. *Clin. Pharmacol. Ther.* **86**, 507–510 (2009).
61. Walker, M. G., Volkmuth, W. & Klingler, T. M. Pharmaceutical target discovery using Guilt-by-Association: schizophrenia and Parkinson's disease genes. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 282–286 (1999).
62. Wang, W., Yang, S., Zhang, X. & Li, J. Drug repositioning by integrating target information through a heterogeneous network model. *Bioinformatics* **30**, 2923–2930 (2014).
63. Eurordis.org. *EURORDIS* https://www.eurordis.org/.
64. Richter, T. *et al.* Rare Disease Terminology and Definitions-A Systematic Global Review: Report of the ISPOR Rare Disease Special Interest Group. *Value Health* **18**, 906–914 (2015).
65. Haendel, M. *et al.* How many rare diseases are there? *Nat. Rev. Drug Discov.* (2019) doi:10.1038/d41573-019-00180-y.
66. Boycott, K. M. *et al.* International Cooperation to Enable the Diagnosis of All Rare Genetic Diseases. *Am. J. Hum. Genet.* **100**, 695–705 (2017).
67. Ferreira, C. R. The burden of rare diseases. *Am. J. Med. Genet. A* **179**, 885–892 (2019).
68. Lancet, T. & The Lancet. Rare diseases need sustainable options. *The Lancet* vol. 395 660 (2020).
69. Luzzatto, L. *et al.* Outrageous prices of orphan drugs: a call for collaboration. *Lancet* **392**, 791–794 (2018).
70. Hernandez, J. J. *et al.* Giving Drugs a Second Chance: Overcoming Regulatory and Financial Hurdles in Repurposing Approved Drugs As Cancer Therapeutics. *Front. Oncol.* **7**, 273 (2017).
71. Plenge, R. M., Scolnick, E. M. & Altshuler, D. Validating therapeutic targets through human genetics. *Nat. Rev. Drug Discov.* **12**, 581–594 (2013).

72. Lindner, M. D. Clinical attrition due to biased preclinical assessments of potential efficacy. *Pharmacol. Ther.* **115**, 148–175 (2007).
73. Perel, P. *et al.* Comparison of treatment effects between animal experiments and clinical trials: systematic review. *BMJ* **334**, 197 (2007).
74. Henderson, V. C., Kimmelman, J., Fergusson, D., Grimshaw, J. M. & Hackam, D. G. Threats to validity in the design and conduct of preclinical efficacy studies: a systematic review of guidelines for in vivo animal experiments. *PLoS Med.* **10**, e1001489 (2013).
75. Hackam, D. G. & Redelmeier, D. A. Translation of research evidence from animals to humans. *JAMA* **296**, 1731–1732 (2006).
76. Wang, B. & Gray, G. Concordance of Noncarcinogenic Endpoints in Rodent Chemical Bioassays. *Risk Anal.* **35**, 1154–1166 (2015).
77. Bailey, J., Thew, M. & Balls, M. An analysis of the use of animal models in predicting human toxicology and drug safety. *Altern. Lab. Anim.* **42**, 181–199 (2014).
78. Nelson, M. R. *et al.* The support of human genetic evidence for approved drug indications. *Nat. Genet.* **47**, 856–860 (2015).
79. King, E. A., Davis, J. W. & Degner, J. F. Are drug targets with genetic support twice as likely to be approved? Revised estimates of the impact of genetic support for drug mechanisms on the probability of drug approval. *PLoS Genet.* **15**, e1008489 (2019).
80. Morgan, P. *et al.* Impact of a five-dimensional framework on R&D productivity at AstraZeneca. *Nat. Rev. Drug Discov.* **17**, 167–181 (2018).
81. Welter, D. *et al.* The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic Acids Res.* **42**, D1001–6 (2014).
82. DiMasi, J. A., Feldman, L., Seckler, A. & Wilson, A. Trends in risks associated with new drug development: success rates for investigational drugs. *Clin. Pharmacol. Ther.* **87**, 272–277 (2010).

83. Morgan, P. *et al.* Can the flow of medicines be improved? Fundamental pharmacokinetic and pharmacological principles toward improving Phase II survival. *Drug Discov. Today* **17**, 419–424 (2012).
84. MacArthur, D. G. & Tyler-Smith, C. Loss-of-function variants in the genomes of healthy humans. *Hum. Mol. Genet.* **19**, R125–30 (2010).
85. MacArthur, D. G. *et al.* A systematic survey of loss-of-function variants in human protein-coding genes. *Science* **335**, 823–828 (2012).
86. Lim, E. T. *et al.* Distribution and medical impact of loss-of-function variants in the Finnish founder population. *PLoS Genet.* **10**, e1004494 (2014).
87. Saleheen, D. *et al.* Human knockouts and phenotypic analysis in a cohort with a high rate of consanguinity. *Nature* **544**, 235–239 (2017).
88. Samocha, K. E. *et al.* A framework for the interpretation of de novo mutation in human disease. *Nat. Genet.* **46**, 944–950 (2014).
89. Lek, M. *et al.* Analysis of protein-coding genetic variation in 60,706 humans. *Nature* **536**, 285–291 (2016).
90. Karczewski, K. J., Francioli, L. C., Tiao, G. & Cummings, B. B. The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature* 434–443 (2020).
91. Sulem, P. *et al.* Identification of a large set of rare complete human knockouts. *Nat. Genet.* **47**, 448–452 (2015).
92. Narasimhan, V. M. *et al.* Health and population effects of rare gene knockouts in adult humans with related parents. *Science* **352**, 474–477 (2016).
93. Minikel, E. V. *et al.* Evaluating potential drug targets through human loss-of-function genetic variation. *Nature* 459–464 (2020).
94. Abifadel, M. *et al.* Mutations in PCSK9 cause autosomal dominant hypercholesterolemia. *Nat. Genet.* **34**, 154–156 (2003).
95. Zhao, Z. *et al.* Molecular characterization of loss-of-function mutations in PCSK9 and identification of a compound heterozygote. *Am. J. Hum. Genet.* **79**, 514–523

(2006).

96. McGregor, T. L. *et al.* Deep phenotyping of a healthy human HAO1 knockout informs therapeutic development for primary hyperoxaluria type 1. *bioRxiv* 524256 (2019) doi:10.1101/524256.

97. Petrovski, S. & Goldstein, D. B. Unequal representation of genetic variation across ancestry groups creates healthcare inequality in the application of precision medicine. *Genome Biol.* **17**, 157 (2016).

98. Need, A. C. & Goldstein, D. B. Next generation disparities in human genomics: concerns and remedies. *Trends Genet.* **25**, 489–494 (2009).

99. Popejoy, A. B. & Fullerton, S. M. Genomics is failing on diversity. *Nature* vol. 538 161–164 (2016).

100. Turnbull, C. *et al.* The 100 000 Genomes Project: bringing whole genome sequencing to the NHS. *BMJ* **361**, k1687 (2018).

101. Vissers, L. E. L. M. *et al.* A clinical utility study of exome sequencing versus conventional genetic testing in pediatric neurology. *Genet. Med.* **19**, 1055–1063 (2017).

102. Lionel, A. C. *et al.* Improved diagnostic yield compared with targeted gene sequencing panels suggests a role for whole-genome sequencing as a first-tier genetic test. *Genet. Med.* **20**, 435–443 (2018).

103. Wright, C. F. *et al.* Genetic diagnosis of developmental disorders in the DDD study: a scalable analysis of genome-wide research data. *Lancet* **385**, 1305–1314 (2015).

104. Barabási, A.-L. & PÃ3sfai, M. *Network Science*. (Cambridge University Press, 2016).

105. Barabasi, A. L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).

106. Albert, R., Jeong, H. & Barabási, A.-L. Diameter of the World-Wide Web. *Nature* **401**, 130 (1999).

107. Redner, S. How popular is your paper? An empirical study of the citation

distribution. *The European Physical Journal B - Condensed Matter and Complex Systems* **4**, 131–134 (1998).

108. Yook, S.-H., Oltvai, Z. N. & Barabási, A.-L. Functional and topological characterization of protein interaction networks. *Proteomics* **4**, 928–942 (2004).
109. Albert, R., Jeong, H. & Barabási, A.-L. Error and attack tolerance of complex networks. *Nature* vol. 406 378–382 (2000).
110. Eisenberg, D., Marcotte, E. M., Xenarios, I. & Yeates, T. O. Protein function in the post-genomic era. *Nature* **405**, 823–826 (2000).
111. Barabási, A.-L. & Oltvai, Z. N. Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* **5**, 101–113 (2004).
112. Barabási, A.-L., Gulbahce, N. & Loscalzo, J. Network medicine: a network-based approach to human disease. *Nat. Rev. Genet.* **12**, 56–68 (2011).
113. Bauer-Mehren, A. *et al.* Gene-disease network analysis reveals functional modules in mendelian, complex and environmental diseases. *PLoS One* **6**, e20284 (2011).
114. Goh, K.-I. *et al.* The human disease network. *Proc. Natl. Acad. Sci. U. S. A.* **104**, 8685–8690 (2007).
115. Hirschhorn, J. N. & Daly, M. J. Genome-wide association studies for common diseases and complex traits. *Nat. Rev. Genet.* **6**, 95–108 (2005).
116. Stevens, A. *et al.* Can network biology unravel the aetiology of congenital hyperinsulinism? *Orphanet Journal of Rare Diseases* vol. 8 21 (2013).
117. Taroni, J. N. *et al.* A novel multi-network approach reveals tissue-specific cellular modulators of fibrosis in systemic sclerosis. *Genome Med.* **9**, 27 (2017).
118. Liu, X., Yang, Z., Lin, H., Simmons, M. & Lu, Z. DIGNiFI: Discovering causative genes for orphan diseases using protein-protein interaction networks. *BMC Syst. Biol.* **11**, 23 (2017).
119. Fotis, C., Antoranz, A., Hatziavramidis, D., Sakellaropoulos, T. & Alexopoulos, L. G. Network-based technologies for early drug discovery. *Drug Discov. Today* **23**, 626–635 (2018).

120. Xia, J., Sinelnikov, I. V., Han, B. & Wishart, D. S. MetaboAnalyst 3.0--making metabolomics more meaningful. *Nucleic Acids Res.* **43**, W251–7 (2015).

121. Guney, E., Menche, J., Vidal, M. & Barábasi, A.-L. Network-based in silico drug efficacy screening. *Nat. Commun.* **7**, 10331 (2016).

122. Huang, L.-C., Wu, X. & Chen, J. Y. Predicting adverse side effects of drugs. *BMC Genomics* **12 Suppl 5**, S11 (2011).

123. Dimitri, G. M. & Lió, P. DrugClust: A machine learning approach for drugs side effects prediction. *Comput. Biol. Chem.* **68**, 204–210 (2017).

124. Said, M. R., Begley, T. J., Oppenheim, A. V., Lauffenburger, D. A. & Samson, L. D. Global network analysis of phenotypic effects: protein networks and toxicity modulation in Saccharomyces cerevisiae. *Proc. Natl. Acad. Sci. U. S. A.* **101**, 18006–18011 (2004).

125. Wachi, S., Yoneda, K. & Wu, R. Interactome-transcriptome analysis reveals the high centrality of genes differentially expressed in lung cancer tissues. *Bioinformatics* **21**, 4205–4208 (2005).

# Chapter 2: LoF variant curation of the Genome Aggregation Database

# Chapter Commentary

As discussed in chapter 1, the use of human genetics in drug discovery is increasing rates of drug development success [1,2]. One particular area of interest is the analysis of loss of function (LoF) mutations. Protein truncating variants (PTVs) resulting in the LoF of a gene-product can serve as a model of acute or life-long inhibition of the gene in question. Such information can shed light on the function of a gene [3], as well as indicate whether pharmacological inhibition of this gene could be tolerated [4]. PTVs have long been studied in model organisms, with groundbreaking work in *C.elegans* [5–7], *Drosophila* [8–10] and Zebrafish [11–13] providing ample examples of the utility of this approach. However with the rise of large-scale genome sequencing projects including the UK Biobank, the 100,000 Genomes Project, and coordinating efforts, like the Genome Aggregation Database (gnomAD), it has become possible to study the consequences of LoF mutations in humans, and at scale.

The work described in this chapter is work arising from a secondment within the MacArthur group at the Broad Institute. Much of the work referred to stems from my analytical contribution to two papers:

1. Karczewski et al. 2020, Nature
   Contribution:
   a. Association of network measures of centrality and measures of constraint
   b. Curation of homozygous LoF variants as described within this chapter
2. Minikel et al. 2020 Nature
   Contribution:
   a. Generalised analytical insight and interrogation of datasets

The following reflects either work that I directly worked on, or information that is necessary in understanding the context of this work. Exact author contributions are located within each of the relevant papers.

# 2.1 Introduction

The accumulation of sequencing data has uncovered a much larger number of LoF variants across the genome than previously thought feasible. Individuals can sustain partial or total LoF in many genes and still remain relatively healthy. However there still exists a strong selective pressure against most LoF variants that one could hypothetically find. Due to this there is a high error rate for LoF variants sequenced, and it can often be difficult to tell those that are real from those that are spurious. In this chapter we will discuss curation efforts in a specific class of LoF variants, homozygous variants.

## 2.1.1 Measuring tolerance to gene inactivation

The Genome Aggregation Database (gnomAD, v2) is a resource providing summary data for 141,456 human exomes and genomes [14]. Analysis of a dataset of this magnitude has allowed for the development of a continuous metric of LoF intolerance across all genes [15]. This metric - the Loss-of-function Observed/Expected Upper bound Fraction (LOEUF) [14] builds on previous constraint metrics to create a continuous measure of tolerance to gene inactivation across the human transcriptome [16,17]. Previously, data from the exome aggregation consortium (ExAC) were used to generate a set of 3,230 genes likely to be intolerant to heterozygous predicted loss of function (pLoF) variation (pLI) [16]. ExAC was a progenitor of gnomAD, and the majority of the 60,706 individuals represented in ExAC are also found within gnomAD. A dichotomous metric was developed using these data [17]. The pLI was calculated by first estimating the mutability of a gene based on factors such as gene length and sequence context, with scores developed per gene and per mutation type, including synonymous, missense, nonsense, essential splice site and frameshift mutations. These values were then compared to observed mutational frequency within a sequenced population. Genes with far fewer observed missense variants than expected are considered constrained [16,17].

Despite the scale achieved in ExAC, it was not possible to calculate a continuous metric as the low frequency of many LoF variants requires very large sample sizes to be sufficiently powered. The more than doubling of samples in the collation of gnomAD

v2.1 provides sufficient power, with 72.1% of genes having more than 10 pLoF variants. From the mutational model described, a ratio of the observed/expected pLoF is calculated. In order to mitigate the effect of gene size on the score, the upper bound of the confidence interval was then calculated to give the LOEUF score. LOEUF scores closer to 0 indicate constraint, i.e. pLoF variants being selected against due to their detrimental impact on fitness. Therefore, we expect genes with high LOEUF scores to be enriched for homozygous pLoF variants [15].

The identification of pLoF variation is focussed on the study of several major classes of variant. Stop-gained, frame-shift and splice-site variants are amongst the types of variation known as protein-truncating variants (PTVs) that result in the disruption of transcription and the partial or total loss of protein function. Structural variation is also an important contributor to PTVs [18], but due to the challenges in resolving such variants with short-read sequencing [18–20], they are not considered in this, or subsequent chapters. Therefore, our PTVs are defined as stop-gained variants, in which a non-synonymous variant leads to a premature stop-codon; essential splice site variants in which splicing of the transcript fails due to disruption of splice donor or acceptor sites; and frameshift variants caused by insertions or deletions (indels) resulting in a change to the reading frame [21].

## 2.1.2 Filtering of spurious LoF calls

A challenge when identifying LoF mutations arises in the form of systematic enrichment for false positives. These errors are inherent in sequencing technology and the following analysis steps such as variant calling. However due to the depletion of true positive LoF variants due to negative selection, pLoF variants are enriched for false positives [21]. Previous studies have identified that as many as 50% of SNP based pLoF variants may be spurious [4,21,22]. It is for this reason that the Loss-of-Function Transcript Effect Estimator (LOFTEE) was developed. This pipeline was inspired by earlier work on LoF variation [16,21], and focuses on the automated curation of stop-gained, splice site disrupting and frameshift variants. As previously described in the introduction, these variants are known as protein truncating variants, and result in the abrogation of functioning gene-product. LOFTEE aims to filter out several forms of identifiable error modes and therefore can serve as an important step in variant annotation.
The LOFTEE pipeline is described in detail in [14], but the filters applied at each stage to pLoF variants are as follows. **Stop-gained and frame-shift variants** are filtered to

remove variants found in the last exon or within 50bp of the 3' end of the penultimate exon (a rule derived from [23]). Variants failing this filter are assessed for the proportion of the transcript affected, and a score of the base-pairs deleted weighted by their evolutionary constraint as determined using the Genomic Evolutionary Rate Profiling (GERP) score [24]. Variants resulting in the loss of constrained genic regions can still be deleterious, and therefore these variants are kept as pLoF. Variants found in exons flanked by non-canonical splice sites are also filtered out. **Splice-site variants** are filtered out if they were found in the splice sites of UTRs, or not predicted to affect a donor site, or if an in-frame rescue splice site could be identified. Variants are further assessed using a logistic regression model centred around scores from MaxEntScan [25,26] and other scores, such that variants not predicted to affect splicing are filtered out. Finally, variants in introns of fewer than 15bp are discarded.

LOFTEE is a relatively conservative approach to variant filtering, favouring precision over recall [14]. However due to the large number of spurious pLoF variants, such an approach is warranted when working at genome-wide scales. This does not necessarily hold for single gene analysis, where it is possible for a curator to analyse all pLoFs of interest. Although labour intensive, such analyses can provide important information to inform machine learning approaches to the same problem. Understanding the reasons for incorrect annotation will be crucial for understanding how to build models to better capture the true state of LoF variation across the genome.

In this chapter we describe the manual curation process performed to filter out such false positives to derive a high confidence set of homozygous pLoF variants. Further work done to functionally categorise such homozygous pLoF containing genes, such as studying their protein-protein interaction network properties will then be described.

# 2.2 Methods

## 2.2.1 Homozygous variant curation

All data were derived from the gnomAD V2.1 dataset, and were sample and site QC'd as described in [14]. Following this, LoF variants were further assessed with LOFTEE. From the 345,458 variants with the most stringent set of LOFTEE criteria (no filters or warning flags), we further filtered to 4,379 variants where at least one homozygous individual was observed. Of these, we further removed variants with low evidence of expression (as described in [27]), resulting in 3,385 variants spanning 2,166 genes, which we subjected to extensive manual curation, in order to filter technical errors commonly found in homozygous LoF variant prediction. These technical errors comprise three main groups: technical errors, rescue events, and transcript errors. Combinations of errors detected within these categories were used to determine if a variant was likely to ablate gene function. After reviewing each variant for technical artifacts, the variant was scored using a five point scale: not LoF, likely not LoF, uncertain, likely LoF, and LoF (see Table 2.1).

| (1) LoF | (2) Likely LoF | (3) Uncertain | (4) Likely not LoF | (5) Not LoF |
|---|---|---|---|---|
| Absence of any evidence to refute a LOF consequence | Weak exon conservation with somewhat high expression | Conflicting evidence/ Ambiguous evidence | Predicted methionine rescue | Multiple errors |
| Weak exon conservation with high expression | Partial loss of exon conservation | | Homopolymer | Frame restoring indel |
| Minority of transcripts but has high expression | Minority of transcripts with somewhat high expression | | Predicted splice rescue | MNV |
| | Predicted weak splice rescue | | Complex mapping | Predicted strong splice rescue |
| | | | Weak exon conservation and poor expression | Reference error |

**Table 2.1 - The scoring scheme for homozygous variant curation.** Variants are individually assessed according to the criteria in this table. The variant is assessed according to its worst flag.

Multiple factors were considered for each level of categorisation, reflecting the level of certainty in the flag assigned.
Technical errors included mapping errors and genotyping errors from sequencing issues, as well as misalignment of reads that could be detected in the Integrated Genome Viewer (IGV) and the UCSC genome browser. Mapping errors are evident when reads around the variant harbored many other variants, especially those with abnormal allele balances. Furthermore, UCSC tracks for large segmental duplications, self chain alignments, and simple tandem repeats were used to determine mapping error status. Genotyping errors were partially eliminated by upstream filtering for read depth, genotype quality, and allele balance (see above). Additional hallmarks for genotyping errors included homopolymer repeats (defined as an insertion or deletion within or directly neighboring a sequence of five or more of the same nucleotide), GC

rich regions, and repetitive regions in which sequencing errors would be more common.

Rescue events include multi-nucleotide variants (MNVs), frame-restoring indels, and essential splice site rescues. MNVs visually identified in IGV and resulting in incorrectly called stop-gained mutations were classified as not LoF. Frame-restoring indels were verified by counting the length of the insertions and deletions to determine if the resulting variation disrupted the frame of the gene. The window used to detect surrounding indels was 80 bp in length. Lastly, splice site rescues were verified by visually inspecting the +/- 20 bp region for an inframe splice site that could rescue the essential splice site. Any inframe splice site within 6 bp of the essential splice site was automatically considered a rescue with a loss/gain of at most two amino acids. Other possible in-frame splice site rescues between the six to 20 bp region of the essential splice site were filtered using Alamut (v.2.11), an alternative splice site prediction tool. Splice sites were classified as rescues if the majority of the splice predictors agreed with an alternative splice site and if the new splice site was not a predicted alternative splice site in the reference transcript.

Finally, transcript errors were described as variants that occur in an exon found in a minority of transcripts for that gene or that occur in a poorly conserved exon. The UCSC genome browser was used to detect both of these situational errors. For an exon to be considered in a minority of transcripts, it had to be present in fewer than 50% of that gene's coding Ensembl transcripts. Exon conservation was determined by looking at the nucleotide bp conservation based on PhyloP. When the variant was found in an exon that occurred in fewer than 50% of transcripts and was not well conserved, it was hypothesized that this exon may be spurious and was therefore curated as likely not LoF or not LoF. If these error modes were found in singular cases, such as when the exon was well conserved but only found in one of four transcripts, then the variant was curated as likely LoF or LoF (these cases were not weighted as heavily due to the transcript filtering that was used before manual curation).
In order for a variant to be considered as LoF, it had to have no major error modes selected (such as LoF rescue). If a single minor error mode was noted for a variant, which include some genotyping or mapping errors, exon conservation, or minority of transcripts, it would be classified as likely LoF. In contrast, rescue errors were automatically classified as likely not LoF or not LoF. Multiple error modes (>= 3) resulted in a "not LoF" curation of the variant. Variants in which there was inconclusive evidence supporting the variant as LoF or not LoF were curated as unknown.

## 2.2.2 Functional categorization

We assessed the correlation between the LOEUF metric and a proxy measure for biological knowledge, the target development level (TDL) from the Pharos database (v5.4.0) [28]. A full definition of the TDL and the associated categories can be found at https://pharos.nih.gov. In brief, gene-products can be categorised into one of four categories based on the drugs and small molecules that target them:

- **Tclin** - targets with approved drugs
- **Tchem** - targets with drug activities in ChEMBL that are not approved for market
- **Tbio** - targets with weaker drug activities that do not meet the required activity thresholds to be classified as Tchem
- **Tdark** - targets about which little is known

For each class, we counted the number of genes in the list in each LOEUF decile and normalized according to the number of genes in the list.

### 2.2.3 Network analysis of gnomAD

Protein-protein interaction networks were used to compare the LOEUF metric to gene-product functional importance. The STRING database [29] was queried using the R API (STRINGdb, v1.22.0) for the protein-protein interactions of all genes with at least 10 expected pLoFs. We then filtered interactions based on their combined scores [30] such that only high confidence interactions (score > 0.7) remained. From this, we generated a directed acyclic graph and kept the largest component resulting in a protein-protein interaction network of 14,955 nodes (proteins) and 315,217 edges (interactions). We then calculated the degree (the number of nodes a node is connected to) of each node. Lastly, we binned proteins based on their gene LOEUF deciles and computed the within decile mean degree with 95% confidence intervals (Fig. 2.2a). These results are discussed in detail in the next chapter.

# 2.3 Results

## 2.3.1 Homozygous variant curation

Using aggregated human sequencing data derived from gnomAD(v2.1), we examined homozygous pLoF variants found in at least 1 individual. To reduce the effect of sequencing and annotation artifacts, these variants were filtered using the Loss-Of-Function Transcript Effect Estimator (LOFTEE) as described in [14]. In addition to this, variants in exons that are unlikely to be expressed in adult tissue were removed (pext score < 0.1 [31]). This left a set of 3,514 homozygous pLoF variants in 2,166 individual genes for deep manual curation. Of these, 73% of variants passed curation filters, leaving 1,752 genes that are likely tolerant to biallelic inactivation (see Table 2.2).

| Verdict | Number of variants | Mean # of flags |
|---|---|---|
| LoF | 1413 | 0.59 |
| Likely LoF | 1163 | 1.65 |
| Uncertain | 292 | 1.69 |
| Likely Not LoF | 252 | 2.46 |
| Not LoF | 394 | 2.37 |

**Table 2.2 - The results of manual curation of homozygous pLoF variants** and the mean number of curation flags for each variant within each verdict class.

Of those confirmed as LoF, 1163 (45%) had some evidence suggesting they may be erroneous, but not enough to overturn a LoF verdict. 292 were marked as ‘Uncertain’ indicating that no clear decision could be made. The mean number of flags per variant increased as the certainty over the LoF designation decreased, with Not LoF variants having on average 4 fold more flags than confirmed LoF variants.
The most common error flag was ‘Genotyping error’, and the least common was ‘Reference error’ (see Fig. 2.1A). LoF verdict variants were significantly depleted for all

error modes (odds ratio < 1, Bonferonni adjusted p.value < 0.05, Fisher's exact test) with the exception of strand bias, for which no enrichment or depletion was present, and the last exon flag, for which they were significantly enriched.

A

| Flag | N (LoF/Uncertain or not LoF)* | Flag | N (LoF/Uncertain or not LoF)* |
|---|---|---|---|
| Genotyping error | 907 (475/432) * | Last exon | 327 (265/62) * |
| Weak exon conservation | 897 (585/312) * | Essential splice site rescue | 222 (90/132) * |
| Mapping error | 716 (478/238) * | Multinucleotide variant | 193 (3/190) * |
| Strand bias | 550 (426/124) | Homopolymer | 104 (2/102) * |
| Minority of transcripts | 529 (257/272) * | Other transcript error | 74 (21/53) * |
| No read data | 394 (149/90) * | Reference error | 42 (2/40) * |

B

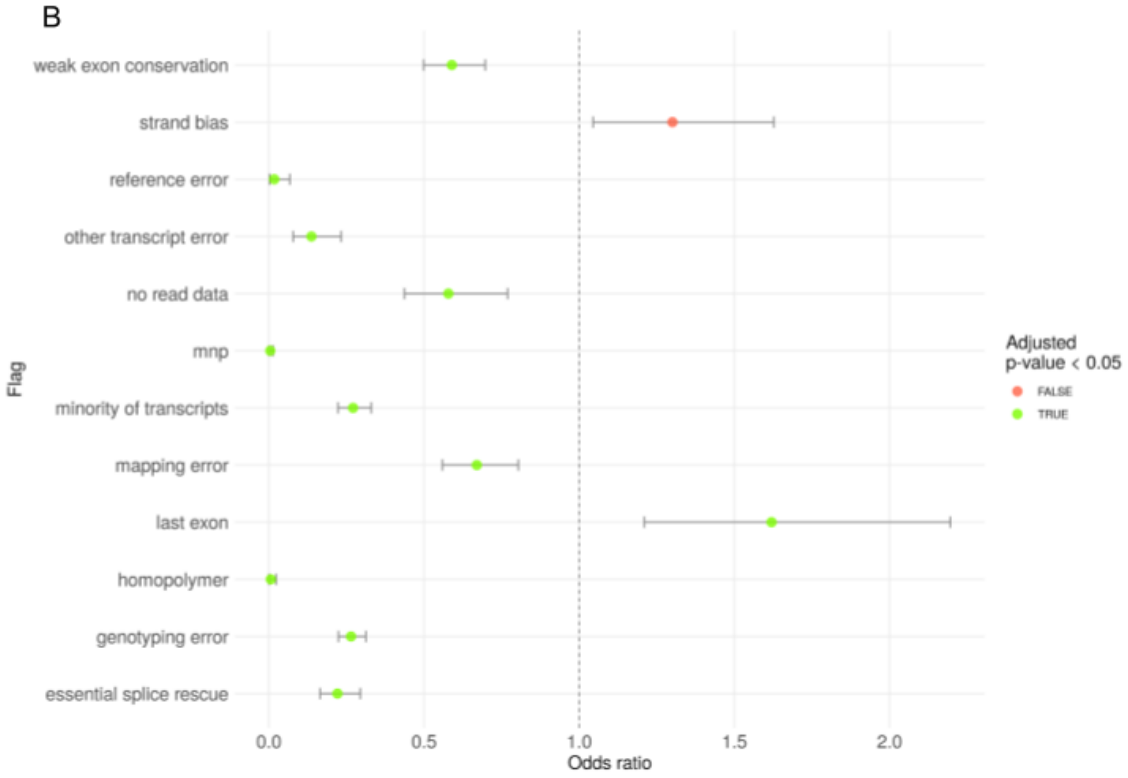



**Figure 2.1 Examining the error modes in LoF and Not LoF variants.** A) Table of the total number of each class of flag for 3,514 homozygous pLoF variants, broken down by LoF and Uncertain or not LoF variants respectively (see within brackets). *Adjusted p.val < 0.05, Fisher's Exact test (Bonferroni correction). B) The odds ratio of error modes, with values above one indicating variants with this flag are more likely to be

LoF. Points are coloured according to their adjusted p-value (as in A), and error bars show 95% confidence interval boundaries.

### 2.3.2 Functional characterisation of LOEUF across the transcriptome

We compared LOEUF to various orthologous measures of functional characteristics. LOEUF was inversely correlated to the degree centrality of the PPI indicating that more constrained genes had more protein interactions (Fig. 2.2A). LOEUF was also inversely correlated to the ‘Target development level’, an indicator of accumulated knowledge about a target, suggesting that unconstrained genes are less studied than their constrained counterparts (Fig. 2.2B).

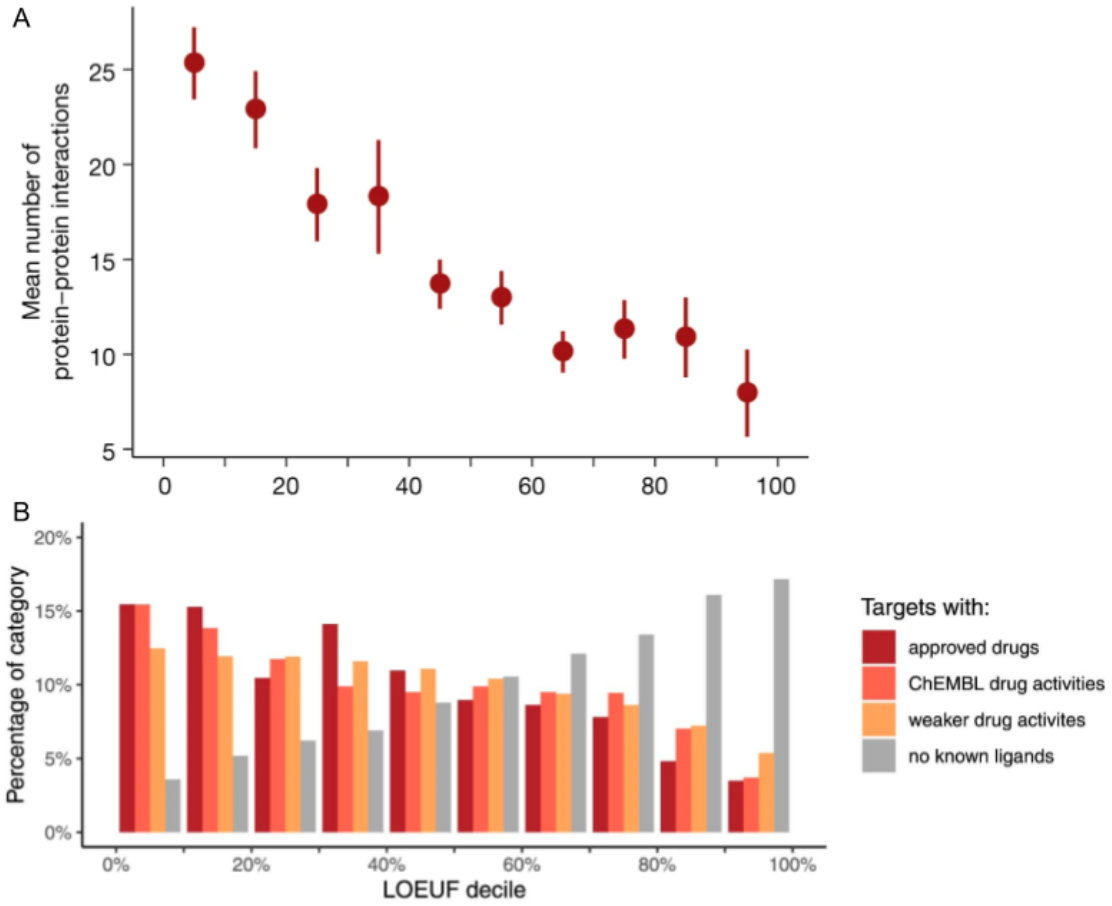



**Figure 2.2 - Biological properties of constrained genes and transcripts**. A) The mean number of protein-protein interactions is plotted as a function of LOEUF decile: more constrained genes have more interaction partners (LOEUF r = -0.14, p = 1.7 x $10^{-51}$). Error bars correspond to 95% confidence intervals. B) The percentage of genes in each functional category from Pharos is broken down by LOEUF decile.

# 2.4 Discussion

## 2.4.1 Manual pLoF variant curation

Studying gene tolerance to LoF within the human population allows to study the real effects of protein knockdown without having to resort to model organisms. Identifying LoF variants allows for the study of the phenotypic consequences of gene disruption, and therefore provides information on their function and essentiality. This exploration of 'the human lab' is contingent on having sufficient sample size to effectively capture variation within the population.

In this chapter we describe work completed whilst embedded at the Genome Aggregation Database (gnomAD) Macarthur group. The broader work in question has involved the aggregation of over 140,000 individual's genetic data followed by the calculation of a continuous metric to describe the spectrum of tolerance to pLoF variants across all protein-coding genes. Due to the high probability of pLoF variants being spurious, careful curation must be completed to produce high-confidence pLoF variants. This was completed both automatically, through the use of the LOFTEE pipeline, and manually, through curation. Due to there being over 400,000 pLoF variants being discovered after filtering using LOFTEE, only homozygous variants were curated in this first instance.

Previous reports have suggested that the pLoF error rate is as high as 50% [21]. The manual curation of the homozygous pLoF variants in gnomAD indicates an effective error rate of 25% (with a further 8% where no decision could be reached after curation) after filtering with LOFTEE. Systematically identifying erroneous pLoF variants is important, as false functional inferences could be drawn from such variants. This is especially important in the context of rare disease analysis, where pLoF variants may be used to diagnose a patient. Due to the complexity inherent in interpreting such variants, tools such as LOFTEE that filter easy to identify error modes are valuable in reducing subsequent curation workload. However, there still remains much room for improvement before pLoF variants can be taken with confidence, immediately post automated analysis.

The rules developed for the manual curation of the homozygous pLoF variants (Table 2.1) were, and still are being designed iteratively, and therefore will be improved upon.

Manual curation exercises such as that described in this chapter will lead to the identification of more error modes, and allow for us to target those that are most pervasive.

Of the variants we curated, we see that the most common error mode is variants that occur in poorly expressed transcripts (variants with a pext score < 0.2 [27]). Following this, the most common error mode was that the variant occurred on a minority of transcripts. Both of these are error modes that it should be possible to programmatically filter in future iterations of LOFTEE (or any alternative pipelines). Another easy to correct mode should be multi-nucleotide variants (MNVs), where SNVs causing a stop codon are proximate to a secondary SNV within the same codon that prevents the formation of a stop-codon [32].
The remaining error modes are less amenable to automated filtering. Splice rescue variants are more challenging to assess, as elements such as cryptic splice sites, reading frame and upstream polypyrimidine tracts must all be assessed. We assessed such variants in two steps, by first considering any in-frame splice site within 6bp as rescues, and then checking any possible in-frame rescues within 20bp with the splice site prediction tool Alamut. The first step can be relatively trivially automated. The second presents a possible route to automation, however Alamut can not be integrated in its present state as the criteria for splice site rescue in the context of pLoF are different from more general splice site rescue. In other words, it is not enough that a splice site rescue occurs, it must also be true that this rescue preserves the function of the original gene.

In order to achieve better automated filtering, it is likely that artificial intelligence (AI) approaches will need to be used. AI is especially suited to problems such as this, where multiple forms of data must be incorporated to classify an outcome variable. Here the integration of sequence, expression and conservation data from various sources would be used to classify a pLoF variant as either valid or spurious. Such an approach may also highlight features that are important in drawing such distinctions that had previously not been considered. Deep learning approaches in particular are likely to present much benefit in this area. Currently, the manual curation required involves analysis of various visual forms of information. This combined with domain knowledge of various forms of artifacts identifiable when examining sequence data lead to decisions being made. Deep learning approaches have shown the ability to outperform humans in similar tasks. As displayed with the variant filtering deep learning model DeepVariant, visual data such as read data from genome browsers can be

incorporated to outperform other best in class classifiers. In reality, the problem of pLoF curation is highly analogous.

Even with such a classifier, as each pLoF variant curated may have multiple error flags, the automated filtering of some of the flags will not necessarily remove the need for manual curation. It is likely that until the maturation and adoption of sequencing technologies better equipped to deal with challenging low-complexity regions or structural variants, manual curation will remain a feature of pLoF analysis.

## 2.4.2 LOEUF functional and network based characterisation

The development of the LOEUF metric represents a significant improvement over the previously used dichotomous pLI score. However such information is only useful if it actually correlates with other biological properties of genes. In keeping with previous work [21,33,34], we find that LOEUF inversely correlates with the gene's degree of connection within STRING based protein interaction networks. This suggests that key pathway genes are more essential and therefore less tolerant to pLoF. Separate to this, we also show that various other measures of centrality correlate to LOEUF, with integration centrality being the most strongly inversely correlated. Numerous studies have used degree centrality as a measure of choice when studying network properties [14,33,35–39], likely due to the intuitive understanding of what this measure means. However, the best measure is dependent on the type of network studied and the complexity of biological networks can likely not be captured using simple measures of local connectivity [40–42]. In this case, we purport that integration centrality, a shortest path based method similar to closeness centrality, better captures what is biologically relevant (see chapter 3.4.2). Integration centrality describes how easily a node is reached from another node [43], and therefore actually provides a more useful measure of information flow through a network than simply how many connections a node has, which is by definition only descriptive of local neighbourhoods. As network measures become more widely incorporated into biological analyses, care must be taken not to simply use some measures purely on the basis of their simplicity [44].

Analysis of target development level, a measure of how studied a gene is, reveals that unconstrained genes are less studied than those that are constrained (Fig. 2.2B). This is expected due to the strong association between constraint and disease gene

membership [14]. As much of basic and translational science is driven by a need to uncover the mechanisms for what ails us, disease causing genes are an obvious starting point for research. This has left many parts of the genome understudied. Such an example is the Olfactory Receptors (OR) protein family, a family of roughly 400 proteins (a further ~460 are pseudogenes with interrupted open reading frames [45]) deriving their name from their role in odour detection. These proteins, initially discovered in the 1990s, were thought to be expressed solely in olfactory epithelium [46]. However mounting evidence shows not only that these can be found ectopically expressed [47–50], but also that they may play important roles in processes such as immune regulation and diseases such as cancer, Alzheimer's, Creutzfeldt-Jakob and schizophrenia [51–53]. Such an example is OR51E2, a ubiquitously expressed gene that is enriched in prostate tissue, both healthy and cancerous with expression levels exceeding those found in olfactory epithelium in the latter case [45]. Inhibition of this OR has been shown to inhibit the proliferation of prostate cancer cells [54], and it has been further implicated in the development of metastatic disease [55]. However this gene is unconstrained, with an O/E of 0.9. This indicates that whilst clearly LOEUF (and the concept of constraint more generally) is strongly correlated with disease causation, it should not preclude a gene from being considered in disease causation or as a drug target. Factors that may confound the relationship between drug target status and constraint are discussed further in Minikel et al. [4].

## 2.5 Conclusion

The sequencing of humans at large scale has exposed the preponderance of pLoF variation within the genome. Despite the ever increasing accuracy of sequencing technologies and variant calling algorithms, pLoF variants are called with a high degree of false positives. This makes the analysis of such variants problematic, an issue further compounded by the importance of pLoF variants for the diagnosis and understanding of disease. With the number of pLoF found in the population increasing, finding ways to systematically filter out such false positives will pay dividends in related research. LOFTEE follows the basic principles of variant curation outlined by earlier research, and is effective at removing a large number of likely spurious pLoF variants. However even after filtering with LOFTEE, manual curation reveals a large number of variants that are likely spurious. Whilst manual curation efforts are valuable, especially in the study of specific diseases [4,22], this approach is not viable at population scale.

We suggest that pLoF variant curation must follow the same path as other variant filtering, and embrace machine learning based solutions. Previous efforts in manual curation will provide important truth sets and clues to which features will accurately weed out specific error modes. This will lead to faster, more systematic, comprehensive, and hopefully in time, more accurate curation of pLoF variation.

# 2.6 References


1. Nelson, M. R. *et al.* The support of human genetic evidence for approved drug indications. *Nat. Genet.* **47**, 856–860 (2015).
2. King, E. A., Wade Davis, J. & Degner, J. F. Are drug targets with genetic support twice as likely to be approved? Revised estimates of the impact of genetic support for drug mechanisms on the probability of drug approval. *bioRxiv* 513945 (2019) doi:10.1101/513945.
3. Musunuru, K. & Kathiresan, S. Genetics of Common, Complex Coronary Artery Disease. *Cell* **177**, 132–145 (2019).
4. Minikel, E. V. *et al.* Evaluating potential drug targets through human loss-of-function genetic variation. *Nature* 459–464 (2020).
5. Kodoyianni, V., Maine, E. M. & Kimble, J. Molecular basis of loss-of-function mutations in the glp-1 gene of Caenorhabditis elegans. *Mol. Biol. Cell* **3**, 1199–1213 (1992).
6. van der Bent, M. L. *et al.* Loss-of-function of β-catenin bar-1 slows development and activates the Wnt pathway in Caenorhabditis elegans. *Sci. Rep.* **4**, 4926 (2014).
7. Cram, E. J., Clark, S. G. & Schwarzbauer, J. E. Talin loss-of-function uncovers roles in cell contractility and migration in C. elegans. *J. Cell Sci.* **116**, 3871–3878 (2003).
8. Balagopalan, L., Keller, C. A. & Abmayr, S. M. Loss-of-function mutations reveal that the Drosophila nautilus gene is not essential for embryonic myogenesis or viability. *Dev. Biol.* **231**, 374–382 (2001).
9. Julienne, H., Buhl, E., Leslie, D. S. & Hodge, J. J. L. Drosophila PINK1 and parkin loss-of-function mutants display a range of non-motor Parkinson's disease phenotypes. *Neurobiol. Dis.* **104**, 15–23 (2017).

10. Lai, Z. C., Rushton, E., Bate, M. & Rubin, G. M. Loss of function of the Drosophila zfh-1 gene results in abnormal development of mesodermally derived tissues. *Proc. Natl. Acad. Sci. U. S. A.* **90**, 4122–4126 (1993).
11. Lebedeva, S., de Jesus Domingues, A. M., Butter, F. & Ketting, R. F. Characterization of genetic loss-of-function of Fus in zebrafish. *RNA Biol.* **14**, 29–35 (2017).
12. Liu, C. *et al.* Apoc2 loss-of-function zebrafish mutant as a genetic model of hyperlipidemia. *Dis. Model. Mech.* **8**, 989–998 (2015).
13. Kabashi, E., Brustein, E., Champagne, N. & Drapeau, P. Zebrafish models for the functional genomics of neurogenetic disorders. *Biochim. Biophys. Acta* **1812**, 335–345 (2011).
14. Karczewski, K. J., Francioli, L. C., Tiao, G. & Cummings, B. B. The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature* 434–443 (2020).
15. Karczewski, K. J. *et al.* Variation across 141,456 human exomes and genomes reveals the spectrum of loss-of-function intolerance across human protein-coding genes. *bioRxiv* 531210 (2019) doi:10.1101/531210.
16. Lek, M. *et al.* Analysis of protein-coding genetic variation in 60,706 humans. *Nature* **536**, 285–291 (2016).
17. Samocha, K. E. *et al.* A framework for the interpretation of de novo mutation in human disease. *Nat. Genet.* **46**, 944–950 (2014).
18. Collins, R. L. *et al.* A structural variation reference for medical and population genetics. *Nature* **581**, 444–451 (2020).
19. Kosugi, S. *et al.* Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. *Genome Biol.* **20**, 117 (2019).
20. Abel, H. J. *et al.* Mapping and characterization of structural variation in 17,795 human genomes. *Nature* **583**, 83–89 (2020).
21. MacArthur, D. G. *et al.* A systematic survey of loss-of-function variants in human

protein-coding genes. *Science* **335**, 823–828 (2012).

22. Whiffin, N. *et al.* The effect of LRRK2 loss-of-function variants in humans. *Nat. Med.* **26**, 869–877 (2020).
23. Nagy, E. & Maquat, L. E. A rule for termination-codon position within intron-containing genes: when nonsense affects RNA abundance. *Trends Biochem. Sci.* **23**, 198–199 (1998).
24. Davydov, E. V. *et al.* Identifying a high fraction of the human genome to be under selective constraint using GERP++. *PLoS Comput. Biol.* **6**, e1001025 (2010).
25. Eng, L. *et al.* Nonclassical splicing mutations in the coding and noncoding regions of the ATM Gene: maximum entropy estimates of splice junction strengths. *Hum. Mutat.* **23**, 67–76 (2004).
26. Yeo, G. & Burge, C. B. Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. *J. Comput. Biol.* **11**, 377–394 (2004).
27. Cummings, B. B. *et al.* Transcript expression-aware annotation improves rare variant discovery and interpretation. *bioRxiv* 554444 (2019) doi:10.1101/554444.
28. Nguyen, D.-T. *et al.* Pharos: Collating protein information to shed light on the druggable genome. *Nucleic Acids Res.* **45**, D995–D1002 (2017).
29. Szklarczyk, D. *et al.* The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Res.* **45**, D362–D368 (2017).
30. von Mering, C. *et al.* STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res.* **33**, D433–7 (2005).
31. Cummings, B. B. *et al.* Transcript expression-aware annotation improves rare variant interpretation. *Nature* **581**, 452–458 (2020).
32. Wei, L. *et al.* MAC: identifying and correcting annotation for multi-nucleotide variations. *BMC Genomics* **16**, 569 (2015).
33. Narasimhan, V. M. *et al.* Health and population effects of rare gene knockouts in

adult humans with related parents. *Science* **352**, 474–477 (2016).

34. Sulem, P. *et al.* Identification of a large set of rare complete human knockouts. *Nat. Genet.* **47**, 448–452 (2015).
35. Tang, X., Wang, J., Zhong, J. & Pan, Y. Predicting Essential Proteins Based on Weighted Degree Centrality. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **11**, 407–418 (2014).
36. Hahn, M. W. & Kern, A. D. Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks. *Mol. Biol. Evol.* **22**, 803–806 (2005).
37. Joyce, A. R. & Palsson, B. Ø. Predicting gene essentiality using genome-scale in silico models. *Methods Mol. Biol.* **416**, 433–457 (2008).
38. Jeong, H., Mason, S. P., Barabási, A. L. & Oltvai, Z. N. Lethality and centrality in protein networks. *Nature* **411**, 41–42 (2001).
39. Jonsson, P. F. & Bates, P. A. Global topological features of cancer proteins in the human interactome. *Bioinformatics* **22**, 2291–2297 (2006).
40. Milenković, T., Memišević, V., Bonato, A. & Pržulj, N. Dominating biological networks. *PLoS One* **6**, e23016 (2011).
41. Ashtiani, M. *et al.* A systematic survey of centrality measures for protein-protein interaction networks. *BMC Syst. Biol.* **12**, 80 (2018).
42. Tew, K. L., Li, X.-L. & Tan, S.-H. Functional centrality: detecting lethality of proteins in protein interaction networks. *Genome Inform.* **19**, 166–177 (2007).
43. Valente, T. W. & Foreman, R. K. Integration and radiality: Measuring the extent of an individual's connectedness and reachability in a network. *Soc. Networks* **20**, 89–105 (1998).
44. Koschützki, D. & Schreiber, F. Centrality analysis methods for biological networks and their application to gene regulatory networks. *Gene Regul. Syst. Bio.* **2**, 193–201 (2008).
45. Olender, T. *et al.* The human olfactory transcriptome. *BMC Genomics* **17**, 619 (2016).

46. Buck, L. & Axel, R. A novel multigene family may encode odorant receptors: a molecular basis for odor recognition. *Cell* **65**, 175–187 (1991).
47. Feldmesser, E. *et al.* Widespread ectopic expression of olfactory receptor genes. *BMC Genomics* **7**, 121 (2006).
48. Budanova, E. N. & Bystrova, M. F. Immunohistochemical detection of olfactory marker protein in tissues with ectopic expression of olfactory receptor genes. *Biochemistry (Moscow) Supplement Series A: Membrane and Cell Biology* vol. 4 120–123 (2010).
49. Zhang, X. & Firestein, S. Genomics of olfactory receptors. *Results Probl. Cell Differ.* **47**, 25–36 (2009).
50. Flegel, C., Manteniotis, S., Osthold, S., Hatt, H. & Gisselmann, G. Expression profile of ectopic olfactory receptors determined by deep sequencing. *PLoS One* **8**, e55368 (2013).
51. Maßberg, D. & Hatt, H. Human Olfactory Receptors: Novel Cellular Functions Outside of the Nose. *Physiol. Rev.* **98**, 1739–1763 (2018).
52. Ansoleaga, B. *et al.* Dysregulation of brain olfactory and taste receptors in AD, PSP and CJD, and AD-related model. *Neuroscience* **248**, 369–382 (2013).
53. Ansoleaga, B. *et al.* Decrease in olfactory and taste receptor expression in the dorsolateral prefrontal cortex in chronic schizophrenia. *J. Psychiatr. Res.* **60**, 109–116 (2015).
54. Neuhaus, E. M. *et al.* Activation of an olfactory receptor inhibits proliferation of prostate cancer cells. *J. Biol. Chem.* **284**, 16218–16225 (2009).
55. Rodriguez, M., Siwko, S. & Liu, M. Prostate-Specific G-Protein Coupled Receptor, an Emerging Biomarker Regulating Inflammation and Prostate Cancer Invasion. *Curr. Mol. Med.* **16**, 526–532 (2016).

# Chapter 3: Predicting pLoF pathogenicity using network topology and machine learning

# Chapter Commentary

In chapter 2 we discussed aspects of LoF that result in spurious LoF assignment. Data on LoF are valuable for an array of reasons, and the identification of all LoF variation across the genome will greatly increase our understanding of human biology. However this task is one that will require sequencing of the majority of the population on the planet in order to have enough power to detect a relatively rare form of variation. As this task is one that will require the overcoming of significant technical, ethical and financial challenges, predicting variants that may contain LoF may serve as a useful alternative. This chapter describes work we completed to try to predict a specific subset of LoF variants, homozygous LoF. As will be explained, these variants may be of particular interest in drug development, and therefore warrant further investigation.

Papers contributing to this chapter:

Karczewski, K.J., Francioli, L.C., Tiao, G. *et al.* The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature* 581, 434–443 (2020)

Minikel, E.V., Karczewski, K.J., Martin, H.C. *et al.* Evaluating drug targets through human loss-of-function genetic variation. *Nature* 581, 459–464 (2020)

# 3.1 Introduction

The observation that using genetic information to inform drug discovery increases rates of success in drug development has led to a push within the pharmaceutical industry to utilise this information[1,2]. This push is evidenced by the increasing investment of companies such as GSK, Biogen and Takeda in projects such as OpenTargets, sequencing the UK Biobank and partnerships with direct to consumer testing companies like 23andMe.

LoF variation is one form of genetic information with a high potential impact on drug discovery. Genes containing rare homozygous LoF variants are nearly twice as likely to reach approval from phase 1 trials (11.4% versus 6.7%, $\chi 2$ test; p = 0.046)[3]. This may be in part driven by LoF variation serving as a proxy for lifelong partial or complete inhibition of a protein target [4], thus providing us with evidence of potential safety profiles: genes harbouring LoF variants with no associated negative phenotype should be similarly able to be targeted in another individual with limited side effects.

The development of a near transcriptome-wide pLoF constraint metric will likely be useful for target prioritisation. The LOEUF metric introduced in the previous chapter is a measure of heterozygous LoF selection. Upon examination of the constraint of all approved drug targets, we find that on average, drug targets are actually more constrained than 'all genes' [4]. The mean observed/expected pLoF constraint of successful drug targets is 44%, versus a global score of 52%. There are many possible confounders to this score, with the inclusion of spurious pLoF calls, and drug class being amongst them; but there is clearly an apparent discrepancy with the previous finding of drug target success in homozygous pLoF containing genes. We will revisit these issues in the next chapter, however it is important to frame this chapter in the light of this information. Identifying *homozoygous* (or compound heterozygous) LoF harbouring genes may be powerful as a predictor of drug development success. However, this presents another challenge; finding all instances of homozygous LoF tolerant genes will require an extensive effort in sequencing genetically bottle-necked or consanguineous populations [4]. Our projections suggest that a full map of homozygous LoF in outbred populations would require a 1,000-fold increase in sample size, the increased autozygosity present in consanguineous populations makes identification of homozygous LoF individuals more probable [4]. Regardless, wide-spread and exhaustive sequencing of these populations will be challenging and expensive. Whilst such a dataset, when it comes into existence will no doubt be the gold-standard

dataset; attempts to predict such genes using other approaches will be beneficial in the medium-term. Here we describe such an effort, whereby we will use gene-level and protein-level data to identify features that allow for the classification of homozygous pLoF tolerant genes.

Machine learning as a field is well suited to such classification problems. However there exist a myriad of possible approaches to feature selection, hyperparameter optimization and algorithm selection. With this in mind, we chose to use the Tree-based Pipeline Optimization Tool (TPOT), a python based automatic ML platform [5]. TPOT simplifies the selection of an ML model by automating processes such as feature selection, preprocessing and construction, model selection and parameter optimization. It is based on genetic programming, a subclass of evolutionary algorithms. This family of algorithms automate aspects of problem solving using principles borrowed from natural evolution. In brief, a series of starting 'unfit' models are generated, upon which a heuristic search is applied to maximise model performance. The most successful (or 'fittest') models are selected according to their performance, and then aspects of these models are randomly substituted between each other in a process akin to reproduction. These offspring of the first models become a new generation, from which more offspring are derived. Such methods have been shown to be more effective than humans in tasks including software repair and the study of finite algebra [5–7].

In this chapter we will outline the aggregation of available data for homozygous LoF in humans. Then we will cover the process of selecting and assessing machine learning models designed to predict which genes might contain homozygous LoF with no associated negative phenotypes.

# 3.2 Methods

## 3.2.1 Dataset compilation

Figure 3.1 summarises the process used to classify LOF for this study. LOEUF statistics across the protein-coding genome were derived from work described in [8] The number of pLoF homozygotes were binarised to 1 if a homozygous pLoF existed, and 0 if not. Further studies were queried for the presence of homozygous pLoFs and the data was updated to reflect this. Data from the Born in Bradford (BiB) cohort [3], deCODE [9] and the PROMIS cohort [10] were used as filters for benign homozygous LoF genes, meaning that genes harbouring homozygous LoF variants identified in these cohorts (and any additional criteria described below) were labelled as benign. Where appropriate data were available, variants were filtered to rare variants (minor allele frequency < 2%) where Variant Effect Predictor (VEP) variant consequence was reported to be either frameshift, splice acceptor, splice donor or stop-gained.

Gene phenotypes were initially all listed as Not Determined (ND) and then recategorised based on the following. All homozygous LoF genes from the BiB cohort were considered benign, homozygous LoF genes from deCODE in which the first recorded death of an individual carrying the LoF was after the age of 50 were considered benign and homozygous LoF genes within the PROMIS cohort identified in one or more individuals and with no association to any one of 250 disease traits tested were considered benign. Genes not found within these gene lists, and listed within the OMIM gene lists were considered deleterious. Gene membership of various other gene lists were also included as binary variables. These lists included genes essential in mice, essential and non-essential genes in human culture from a CRISPR/CAS9 screen, genes near GWAS catalog peaks (MacArthur et al. 2017) and genes designated as haploinsufficient in ClinGen (see https://github.com/macarthur-lab/gene_lists/ for access to gene lists, accessed 13/02/2019). Genes found within the essential culture and haploinsufficient gene lists without a prior benign designation were classified as deleterious. All such classifications are summarised in Fig. 3.1.
We then added various other features to enrich the dataset including:

- Gene ontology (GO) terms, terms covering molecular function, cellular component and biological process.

- Genotype-Tissue Expression (GTEx) data - featuring data on gene expression across 53 tissues. TPM scores were added as features for all tissues.
- Interactome INSIDER (INtegrated Structural Interactome and genomic Data browsER) - This data integrates genomic and structural data to predict whether genomic variants occur within amino acids involved in protein binding. This has been summarised to the gene level, simply stating whether this gene contains variants in the protein-binding domain or not.
- Paralog data - Data from Ensembl BioMart have been included to show the number of paralogs for each gene in the network. GO molecular function and protein sequence based similarity scores have also been calculated within each paralog group.
- Online Mendelian Inheritance in Man (OMIM) - Genes found within the OMIM database have been highlighted. We subsetted the dataset based on the geneMap file (date downloaded 07/08/2018) such that phenotype IDs indicating non-disease phenotypes, mutations contributing to multifactorial disorders or infections and cases in which the relationship between the gene and phenotype are provisional were all removed.
- Druggability data (PHAROS) - data on target druggability and target development level are included.

Two versions of this dataset were compiled, a loose-deleterious and strict-deleterious set, (l-dS and s-dS respectively). The main difference concerns how deleterious genes are classified based on filtering of OMIM genes. The l-dS seeks to be inclusive of OMIM genes that are linked to disease phenotypes, but are not necessarily disease causing. The s-dS is limited to genes more likely to be directly disease causing. Both datasets are available upon request. All results relate to the s-dS unless specifically stated otherwise, although most database characteristics will be explicitly explored for both sets.

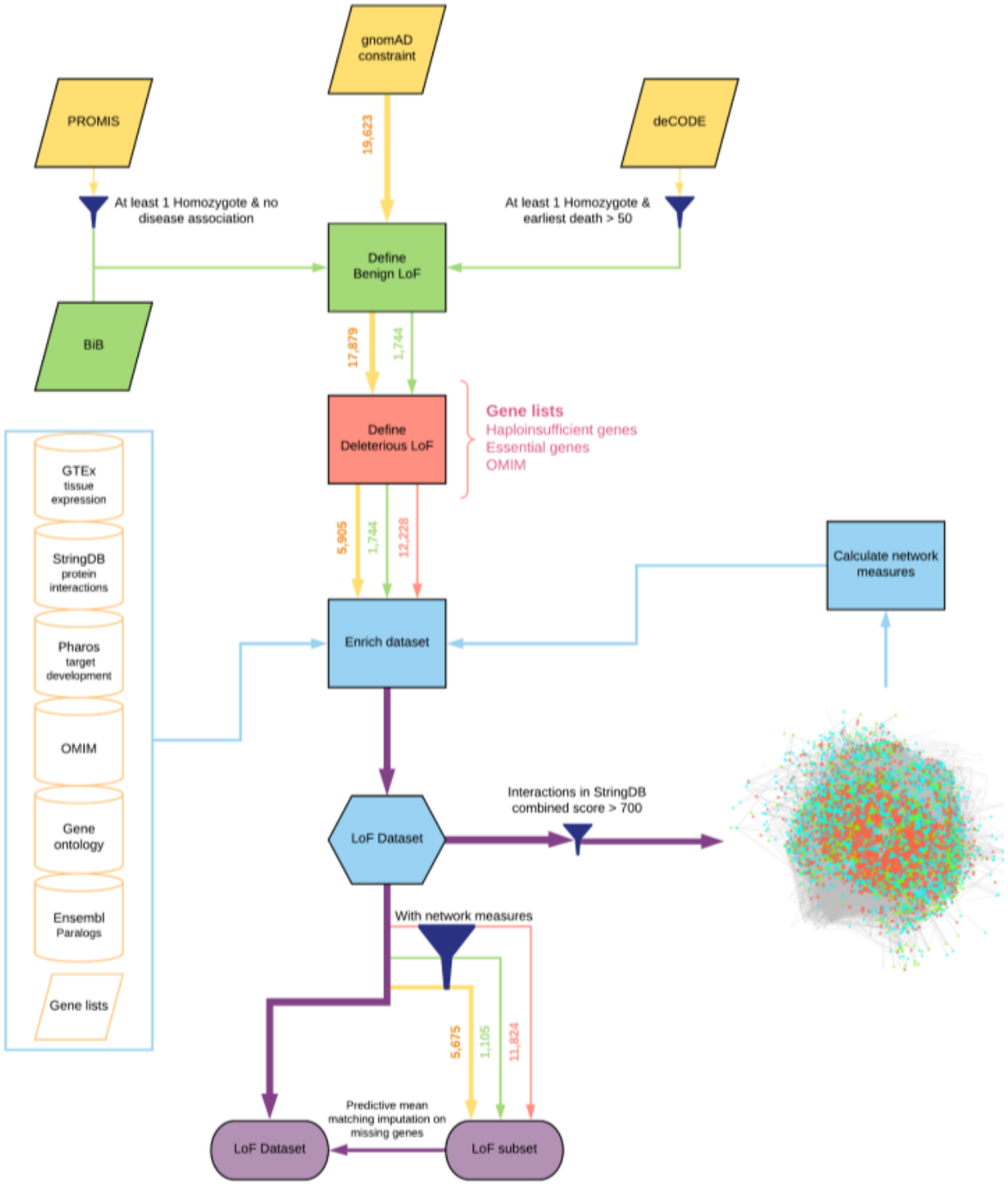



**Figure 3.1 - Schematic outlining the creation of the LoF dataset.** We took the total complement of protein coding genes from gnomAD, and classified genes as benign based on filters specific to each study. Genes were then further classified as deleterious if they were not previously characterised as benign, and belonged to either haploinsufficient, essential gene or OMIM gene lists. This dataset was then enriched with annotations from various different databases, and network measures based on protein-protein interaction data were calculated for genes for which such information was available. The LOEUF score was then used to impute missing network measure information using predictive mean matching imputation.

### 3.2.2 Calculating network metrics

We queried StringDB to identify all protein-protein interactions for the genes in the LoF dataset. Low confidence interactions (combined score < 700) were removed. From this a network of 14,791 nodes and 312,099 edges was created comprising 105 components (disconnected subgraphs). Of these the largest component was kept resulting in a network of 14,546 nodes and 311,9213 edges comprising 815 benign, 3327 deleterious and 10,404 ND (s-dS dataset) nodes. Network metrics were subsequently computed using the R package Tidygraph (v1.2.2), including various measures of centrality such as degree, closeness, betweenness, eigenvector, integration and hub.

### 3.2.3 Visualising GO embedding

We visualised the GO embeddings generated using Opa2vec through t-Distributed Stochastic Neighbor Embedding (t-SNE), a method for non-linear dimensionality reduction. We generated a feature matrix containing all GO embeddings, and then ran t-SNE using Rtsne (v0.15), with theta set at the default 0.5 and dimensions set to 2 with no prior PCA step. Due to the stochastic nature of the method, embeddings were calculated 5 times each and using a range of perplexity values (2, 5, 10, 20, 30, 40, 60, 80 and 100). Outputs were plotted using ggplot2 and we visually inspected the plots to decide which perplexity value we would use for further visualisation.

### 3.2.4 Machine Learning protocol

#### 3.2.4.1 Data preparation

We dropped redundant columns, those containing unique identifiers, or those that were used to define the phenotype class. This included columns related to transcript ID, gene name, protein identifiers, OMIM accession numbers, data source and gene list membership.

GO terms were embedded into a lower dimension feature space using the python pipeline opa2vec [11]. In brief, this method uses the neural-network based tools Word2Vec [12] to generate vector representations of words from corpora (the whole gene ontology). It then combines the multiple annotations that may be assigned to each gene to create a per entity (gene) set of 200 features.
We applied predictive mean matching imputation (R package MICE, version 3.8.0) to fill missing network measures due to genes not being represented in the network. Due to the gnomAD results described previously showing correlation between centrality measures and LOEUF, we used the LOEUF score as the explanatory variable in the imputation model.
We then binarized categorical columns including those detailing the chromosome, the target development level data and network clustering based group membership.

We defined the target column for the ML algorithm as the phenotype column, with the positive target being ‘Benign’. We dropped all ‘ND’ phenotype rows such that only labelled data remained. We then split the data into training and test data (80% training, 20% test) by splitting each label group into 80/20 subsets to ensure we maintained class distributions. The training data was then used as the input data for TPOT.

#### 3.2.4.2 Model Identification using TPOT

As previously discussed, TPOT is a python based genetic algorithm designed to programmatically identify and test possible ML models. This pipeline includes feature selection and engineering, and hyperparameter optimization. We limited the number of possible generations to 300, with 100 offspring in each generation. We allowed a maximum run-time of 2 weeks, with an early break upon convergence of model performance (defined as 50 generations with no offspring improvement). We defined that models should be evaluated for accuracy after 10-fold cross validation. The best performing pipeline was then saved as a standalone python script. We evaluated this model by classifying the data in the 20% test set.  We report accuracy, precision and recall scores, along with the harmonic mean of the latter 2 scores, known as the F1 score.
Finally, we applied the model to the unlabelled ‘ND’ genes. Genes were assigned a 1 if they were predicted to be Benign (pBenign), and 0 otherwise (pNotBenign). We also report label confidence scores ranging from 0 to 1, in which values approaching 1

indicate high confidence in the label prediction. Values above 0.5 are indicative of a positive prediction.

A third dataset derived from the l-dS was created in which all gnomAD derived data was removed. This includes all allele frequency data and LOEUF scores. All other parameters were as described above.

In order to provide multiple model types for comparison, we also ran two iterations of TPOT in which the possible model types were restricted to logistic regression and decision tree. These models are included in the standard runs of TPOT, and therefore we only ran these options for 25 generations (additionally, the high numbers of generations are less important in cases where the model space is so restricted). Aside from these differences, the model was run as above, allowing for feature selection and engineering to be carried out to increase model performance. This was performed for the l-dS only.

Finally, in order to ascertain whether we were achieving an optimal model for the feature space, we ran multiple iterations of the entire pipeline. As TPOT is a stochastic pipeline, it is probable that any one solution may be reflecting a local minimum point, rather than a global minimum. Due to time and computational constraints, we ran 3 runs on the s-dS only. Label stability was measured by gene membership concordance between each of the solutions.

### 3.2.4.3 Feature importance estimation

We sought to estimate the importance of each of the model features as this can often shed light on the biology driving the predictions. Due to the models created above largely being ensemble models, with complex feature engineering and selection resulting in profound abstraction from the original dataset, we built an entirely separate model. To retain as much explainability as possible, we used all 316 features in the cleaned dataset (see Appendix 3.2.4.3 ), with no feature engineering steps. As the preferred models from the TPOT model were based on random forests, we also elected to build a random forest model using the R package randomForest (v1.4-2). We first performed a parameter grid search, in which the number of trees was allowed to vary from 1, 10, 20, 30, 40, 50, 100, 200, 300, 400, 500, 700, 800, 900 to 1000 trees.

These were then compared on the basis of their out-of-bag error rate (OOB). We selected the model with the lowest overall OOB. Following this, we ran a random forest cross-validation, in order to ascertain the optimal number of features to include within the model. We started with all features, and halved the number included at random for each iteration. From this we ascertained that the optimal number of features was 20. We subsequently calculated the feature importance of the chosen random forest model, and selected the top 20 features. Using just these features as input, we built another random forest model.

# 3.3 Results

## 3.3.1 Dataset generation

We generated a dataset of 19,623 genes based on protein-coding genes reported in gnomAD. These were then classified based on whether the genes have been found in human populations with homozygous pLoF in at least one individual with evidence of no associated phenotype. We report 1,744 such genes, labeled as ‘Benign’ due to the apparent lack of phenotype associated with their inactivation in humans. This is true for both the l-dS and s-dS as benign genes were defined in the same way across both datasets. The l-dS had a total of 12,228 deleterious genes leaving 5,905 ‘ND’ (Fig. 3.3A). The s-dS had 3,469 genes reported as ‘deleterious’ with 14,426 ‘ND’ (Fig. 3.2A). Two iterations of the TPOT pipeline were run to label both sets of ‘ND’ genes.

## 3.3.2 Group characteristics

As reported in [13], the median o/e constraint score across the transcriptome was 0.48. Within groups we report a median score 0.79, 0.37 and 0.47 for Benign, Deleterious and ND genes respectively for the s-dS (Fig. 3.2C) and 0.79, 0.39 and 0.57 for the l-dS (Fig. 3.3C). Comparison of the phenotype assignments to the gnomAD LOEUF bins show concordance, with 63% of Benign genes in the s-dS being found in the top 40th percentile of transcriptome-wide scores, and 54% of Deleterious genes in the bottom 40th percentile. ND genes are more evenly spread, with no significantly different percentage across any of the deciles (One-way anova p-value > 0.1, Fig. 3.2B). The l-dS has a near identical distribution for the Benign genes, but a more evenly distributed deleterious set, with 50% of Deleterious genes occurring in the bottom 40th percentile (Fig. 3.3B). This difference is most marked in the 90th percentile, with only ~3% of s-dS deleterious genes being found here, compared to ~6% of l-dS genes. The unknown genes are much more right-skewed with over 17% of genes belonging to the 90 percentile, compared to 11% in the s-dS.

Overall, between both sets there is concordance between the phenotype classifications and existing constraint metrics, however the difference in the assignment results in a marked shift in the ND class genes, with those being enriched for less constrained genes. This would indicate that the l-dS captures more of the disease causing, or disease related genes.

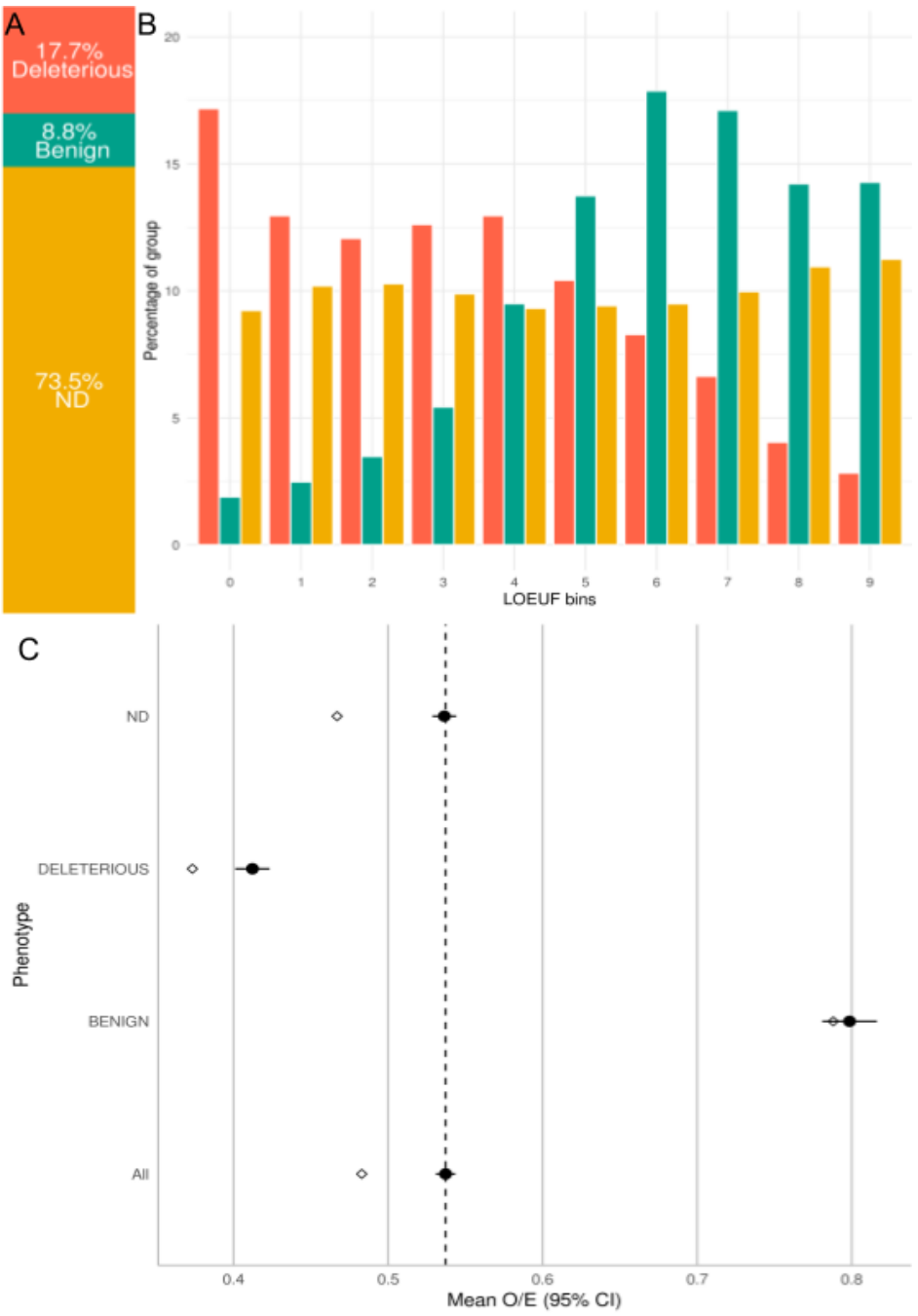



**Figure 3.2 - Dataset characteristics of the strict deleterious set.** A) Stacked bar plot showing the percentage of the total 19,623 genes in each phenotype category. The majority of genes are classified as ND, indicating that they are neither found as Mendelian disease causing genes within OMIM, nor are they found in homozygous pLoF form in the studies examined. B) A Bar plot showing the percentage of each phenotype found in each LOEUF bin. Few genes found in homozygous pLoF state are found within the more constrained bins, with the majority being found in bins 5 and above; the inverse is true for the deleterious genes. ND genes are relatively evenly distributed. Colours are as in panel A. C) Forrest plot of the mean O/E LoF with 95% confidence intervals for each phenotype and all genes collectively. The clear diamond denotes the median value for each.

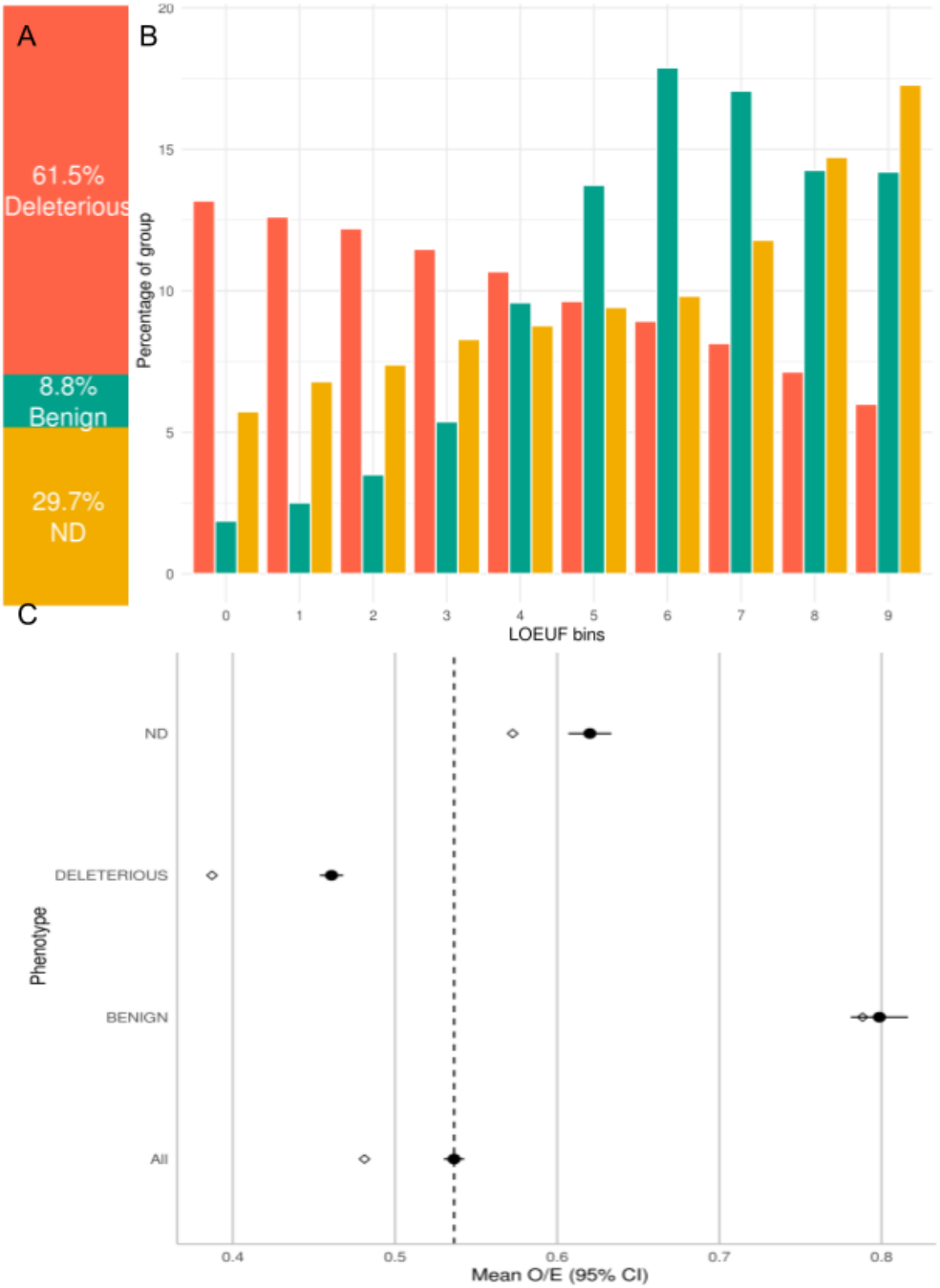



**Figure 3.3 - Dataset characteristics of the loose deleterious set**. A) Stacked bar plot plot showing the percentage of the total 19,623 genes in each phenotype category. The broadening of the genes considered deleterious has means deleterious genes now make the majority class. B) Bar plot showing the percentage of each phenotype found in each LOEUF bin. The expansion of the deleterious class means that more deleterious genes are found in higher LOEUF bins. However, fewer ND genes are found in the lower LOEUF bins, with the distribution mirroring that of the benign genes. Colours are as in panel A. C) Forrest plot of the mean O/E LoF with 95% confidence intervals for each phenotype and all genes collectively. Compared to Fig. 3.2, the average ND LOEUF score has increased, indicating that fewer constrained genes are found within this set. The clear diamond denotes the median value for each.

Comparison of our phenotypes to mouse essential genes and International Mouse Phenotype Consortium (IMPC) developmental phenotypes shows conservation of

essentiality across the species (Fig. 3.4 and Fig. 3.5). The Target Development Level (Fig. 3.4) from the Target Central Resource Database (TCRD) categorises targets based on the development of ligands for the related protein and other markers of knowledge about the target. The categories are Tclin - targets that have approved drugs with known mechanism of action, Tchem - targets with ligand activities that may not be approved, Tbio - targets with no known ligands that satisfy activity thresholds, and finally Tdark - targets about which little is known.

Tbio constitutes the largest group in all categories, although this group is enriched within the Deleterious targets. The Benign and ND targets are enriched for Tdark compounds, representing 39% and 30% of the groups respectively compared to only 4% of the Deleterious targets. Of the 613 Tclin targets, 222 are found within the Deleterious group, 42 in the Benign group and 332 in the ND group.

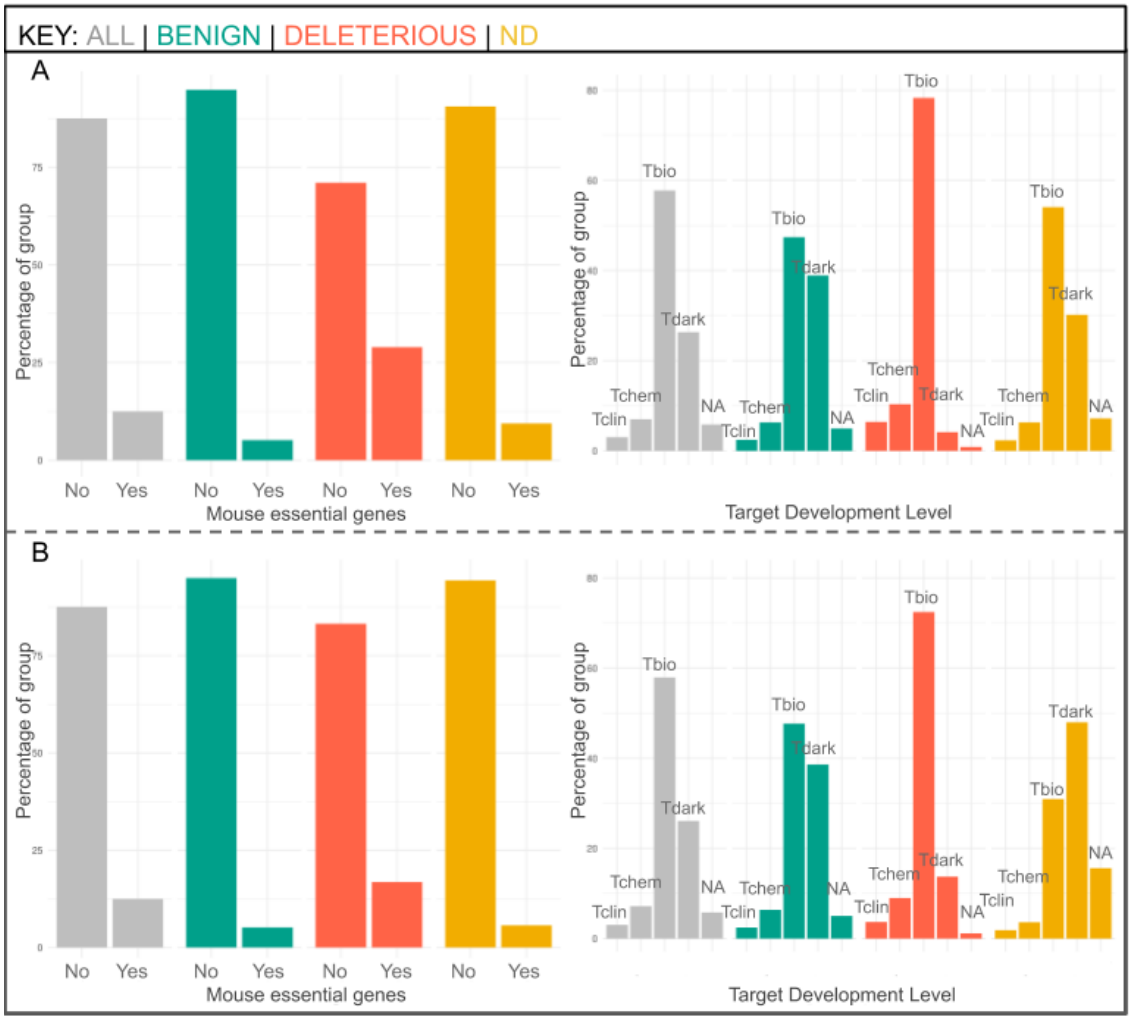



**Figure 3.4 - Biological and functional characteristics associated with the phenotype groups.** Panels A and B relate to the s-dS and l-dS respectively. The left

hand graphs show the percentage of each phenotype group that is also a mouse essential gene. The right hand graphs display the distribution of target development level by phenotype group.

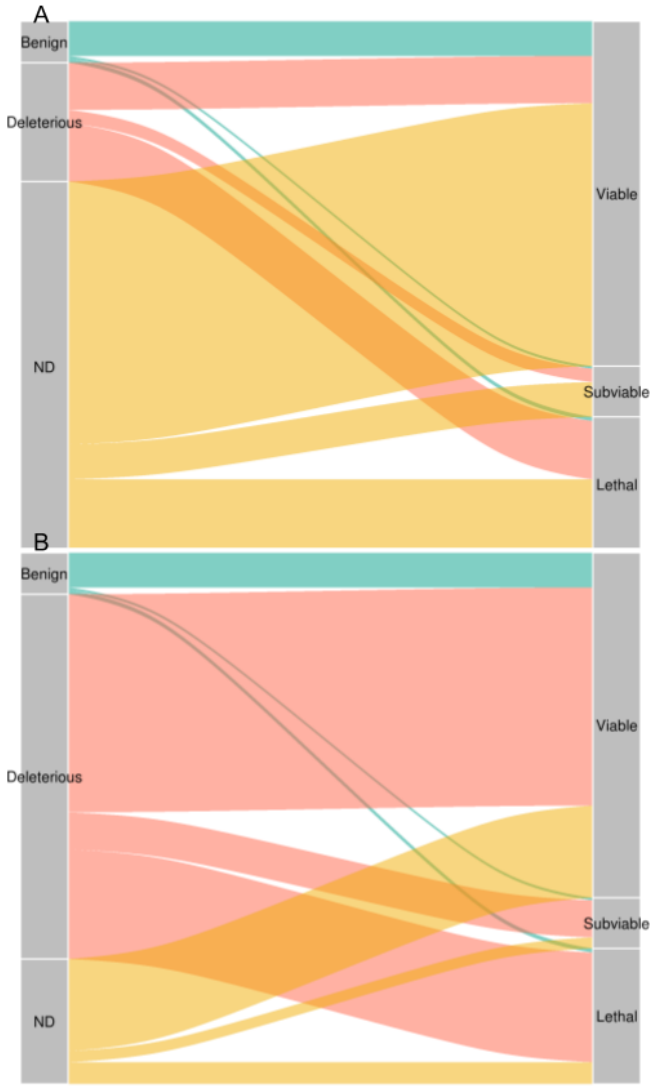



**Figure 3.5** - **Alluvial plot showing phenotype gene membership to IMPC developmental phenotypes for the s-dS (A) and l-dS (B) datasets**. Human genes were 1-1 matched to orthologous mouse genes, n=3,845. Mouse developmental phenotypes are in three categories as defined by the IMPC that serve as roughly analogous comparisons to our human phenotypes. We observe considerable conservation across genes compared, with Benign genes in humans predominantly falling within the Viable mouse phenotype.

### 3.3.2.1 Network characteristics

We evaluated several measures of network centrality (Figure 3.6). Generally all metrics show that the groups share similar shaped distributions indicating that it is unlikely that LoF drives network properties. The degree centrality, a measure of the number of connections a node has, reveals that the Deleterious group has a more heavy-tailed distribution with a median score of 9 vs 4 and 5 for the Benign and ND groups respectively. This echoes results reported in [14,15], although these refer more directly to genes with respect to their constraint. All other centrality measures show similar effects to varying degrees. This is true within both the s-dS and l-dS.

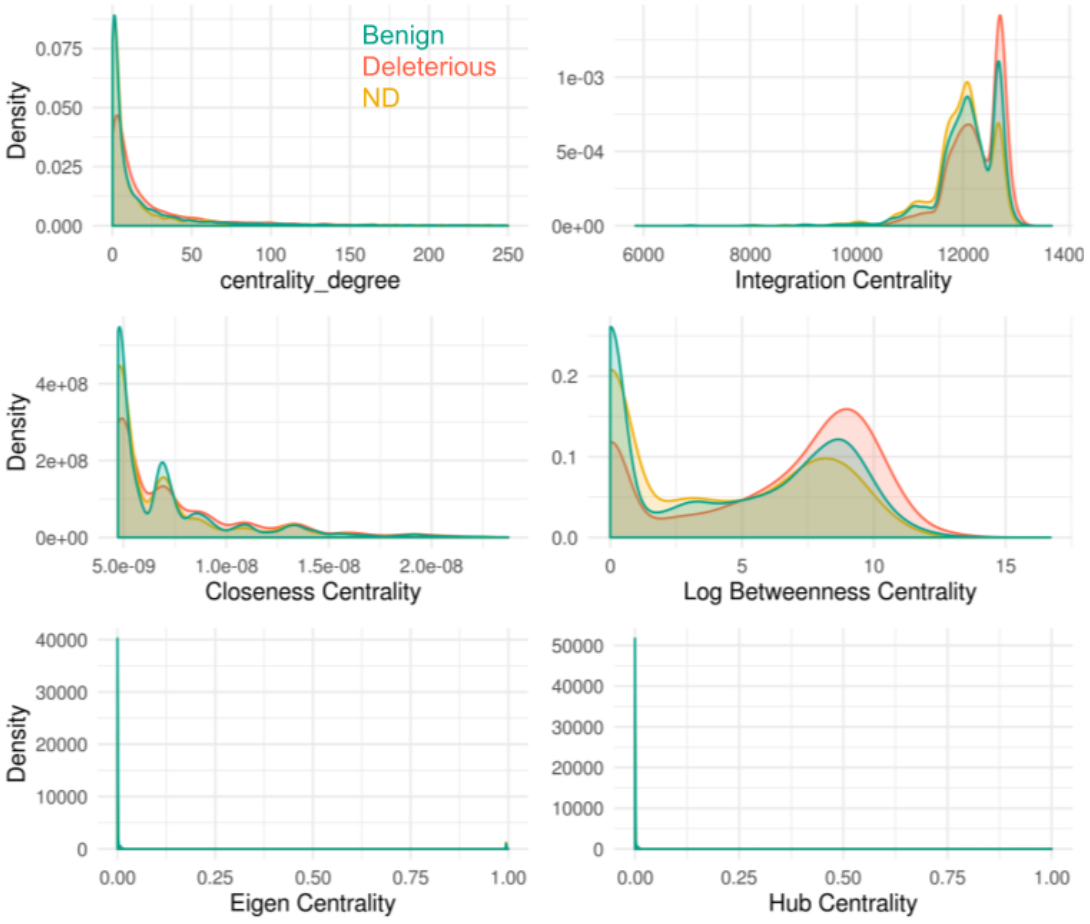



**Figure 3.6** - **Density diagrams displaying the distributions of network metrics according to phenotype**. Degree Centrality has been truncated at 250 degrees along the X axis. Colour codes remain consistent throughout each.

As shown in Fig. 3.7, all measures are positively correlated to each other, and inversely correlated to the O/E LoF. Whilst the degree of centrality has previously been reported

with regards to constraint, we report that the strength of correlation is relatively weak (Spearman correlation, $\rho$ -0.11), with the strongest inverse correlation being found with the integration centrality (Spearman correlation, $\rho$ -0.27). In fact the degree distribution is the most weakly correlated of the centrality measures.

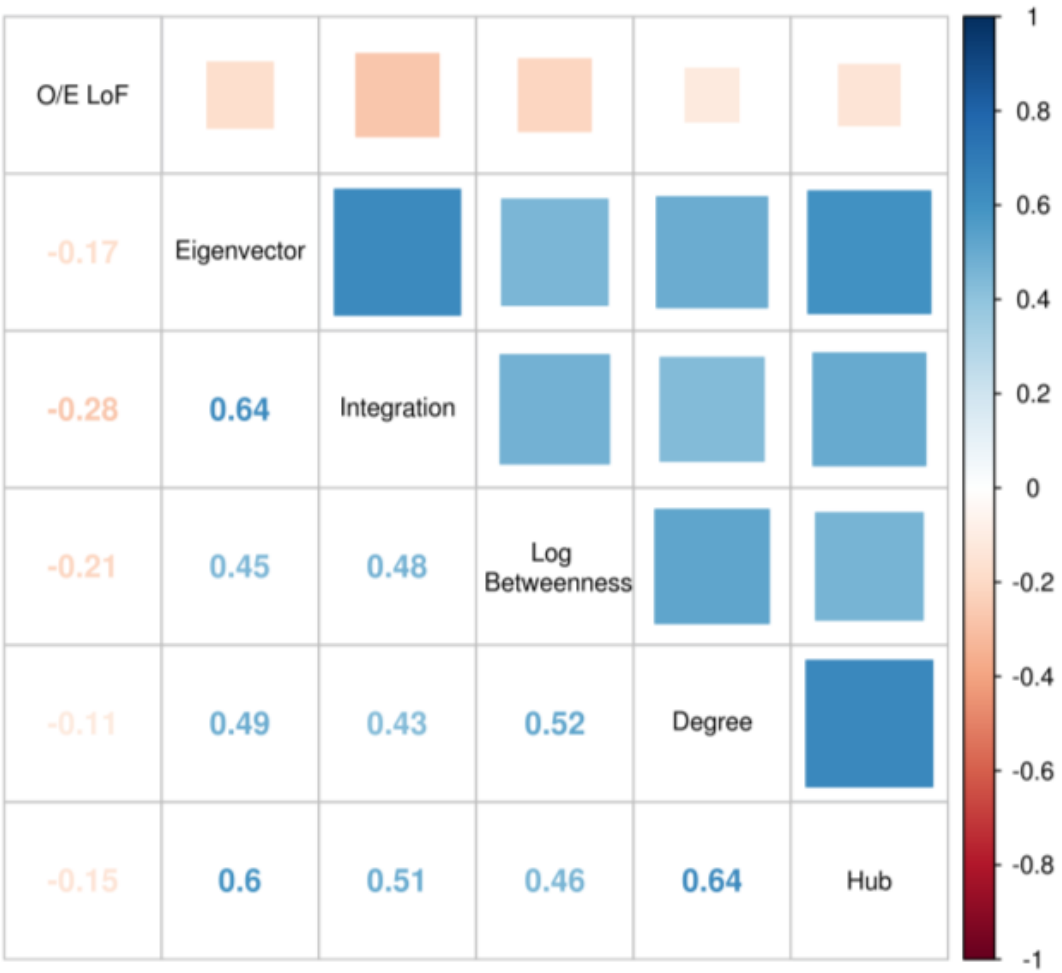



**Figure 3.7** - **Correlation plot comparing the Spearman rank correlation between measures of centrality and the O/E LoF score**. The lower triangle contains the correlation coefficients, positive correlations are in blue and inverse correlations in red, with alpha increasing with increased magnitude. All correlations were significant ($p<0.05$).

### 3.3.2.3 Imputation of network metrics

When creating the PPI network, we limited interactions to those with high combined confidence scores (> 700). This resulted in the loss of many genes, with 5,077 genes not being represented. To prevent the loss of these in the machine learning pipeline,

we performed multiple imputation to assign network metrics using the O/E LoF score as an explanatory variable.

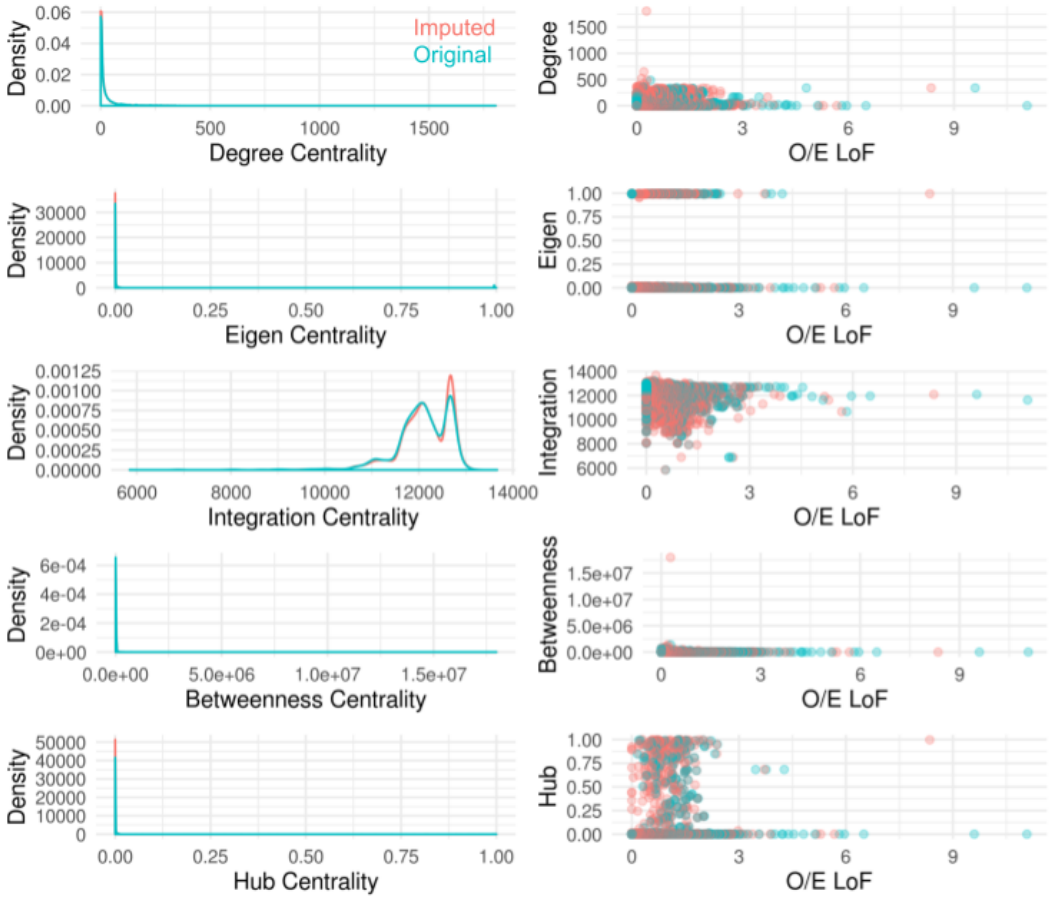



**Figure 3.8 - Comparison of imputed to non-imputed data across various network centrality metrics.** Graphs on the left are density plots of each of the metrics, on the right are scatter plots adding the dimension of the explanatory variable, O/E LoF. Imputed values are shown in red and original values in teal.

Figure 3.8 displays the distribution of imputed to original data both in isolation, and in reference to the explanatory variable, O/E LoF. Across all metrics we see close replication of the original data distribution, albeit with some increased density around peaks, such as the integration centrality distribution in which the second peak of the bimodal distribution is slightly accentuated upon imputation. This is consistent with the reduction in standard error commonly seen in mean imputation and it is not a cause for concern in this case.

### 3.3.3 Apparent phenotype disparities

Genes were labelled as deleterious based on:

- The gene was not previously labelled as Benign
  AND Present in the Clingen haploinsufficient (HI) list
  OR Present in the OMIM disease gene list
  OR Present in the essential culture list

Here we examine cases in which genes were Benign despite membership of the previous three lists. As there is no difference in the way Benign genes were classified between datasets, all following observations are true for both. 294 genes were listed as haploinsufficient (HI) in Clingen. Of these, 6 had a designation of Benign due to their being present in the BiB cohort (Table 3.1). Of these, FLG (Fillaggrin) and PXMP2 (peroxisomal membrane protein 2) have LOEUF scores indicating that they are unconstrained or under relatively mild selection. The remaining 4 all have LOEUF scores indicating LoF constraint, with ZEB2 being part of the most constrained bin of the transcriptome.

Following from this, of the 3062 OMIM disease genes, 174 were also labelled as benign, with 81, 45 and 48 genes found in a homozoygous state in the BiB, PROMIS and deCODE cohorts respectively. This subset is less constrained with a median LOEUF score of 0.99, compared to 0.91 for the whole transcriptome.

Finally, of the 682 genes report to be essential in cell culture, 16 were within the Benign set with a median constraint of 0.77. Combined, this represents a set of 190 unique genes in which there appears to be a disparity between gene list membership and given phenotype according to study membership. Further curation of these genes may indicate that the LoF variants are in fact spurious, but this is beyond the scope of this work.

| Gene Name | O/E LoF | LOEUF | LOEUF Decile | Clingen HI | Source |
|---|---|---|---|---|---|
| FLG | 2.42 | 1.96 | 9 | 1 | BIB |
| PXMP2 | 0.56 | 1.27 | 7 | 1 | BIB |
| MLH1 | 0.37 | 0.57 | 2 | 1 | BIB |
| NF1 | 0.22 | 0.29 | 1 | 1 | BIB |
| GLI2 | 0.18 | 0.31 | 1 | 1 | BIB |
| ZEB2 | 0.02 | 0.11 | 0 | 1 | BIB |

**Table 3.1** - **Examples of homozygous pLoF containing genes.** Genes are labelled as haploinsufficient in Clingen, or with a LOEUF score or point-estimate O/E LoF score indicating constraint. All pLoF variants were identified from the BIB cohort in this case. The LOEUF Decile are bins of equal size binnings of genes according to their LOEUF score.

Despite not being used as a measure of whether a gene should be reported as Benign or not, the O/E LoF score is a direct measure of constraint. Therefore we would *a priori* expect the number of genes found in the homozygous state or listed as benign to be depleted for genes with a score of < 1 with decreasing numbers being found as O/E approaches 0. Within this dataset we report 4247 genes to have been found in the homozygous state in at least 1 individual. Of these, 1,020 have an O/E LoF of < 0.5, of which 306 were classified as benign, 141 as deleterious and the remaining 573 as ND.

### 3.3.4 Visualising GO embedding

In order to generate a dataset suitable for machine learning algorithms, categorical variables must be binarised or converted into a continuous metric. Efforts to binarise the gene ontology resulted in the generation of > 17,000 features. To avoid such expansion of our feature set we performed feature embedding. The pipeline Onto2Vec was used to generate a set of 200 features as a lower dimensionality representation of the gene ontology across all levels (molecular function, cellular compartment and

biological process) [11]. We visualised this data using t-SNE, a non-linear technique that projects our featureset to 2 dimensions [16]. Due to the inherent variability in representing the data with t-SNE, we ran 5 iterations at multiple perplexity scores. As a rule t-SNE plots emphasise local structure, however, alteration of perplexity (defined as 2 to the power of the Shannon entropy, a measure of the amount of information within a system) changes the balance between the importance of local and global structure in the resulting t-SNE plot. This in effect changes the number of nearest neighbours for each point, a key component in collapsing our high dimensional data to 2 dimensions. Generally we expect data with high entropy to require a higher perplexity, but this is not necessarily easy to divine, and is best explored through running multiple different perplexity scores, as in Fig. 3.9. t-SNE is thought to produce the best results using perplexities ranging from 5-50, however numerous examples exist of greater perplexities being required [16].

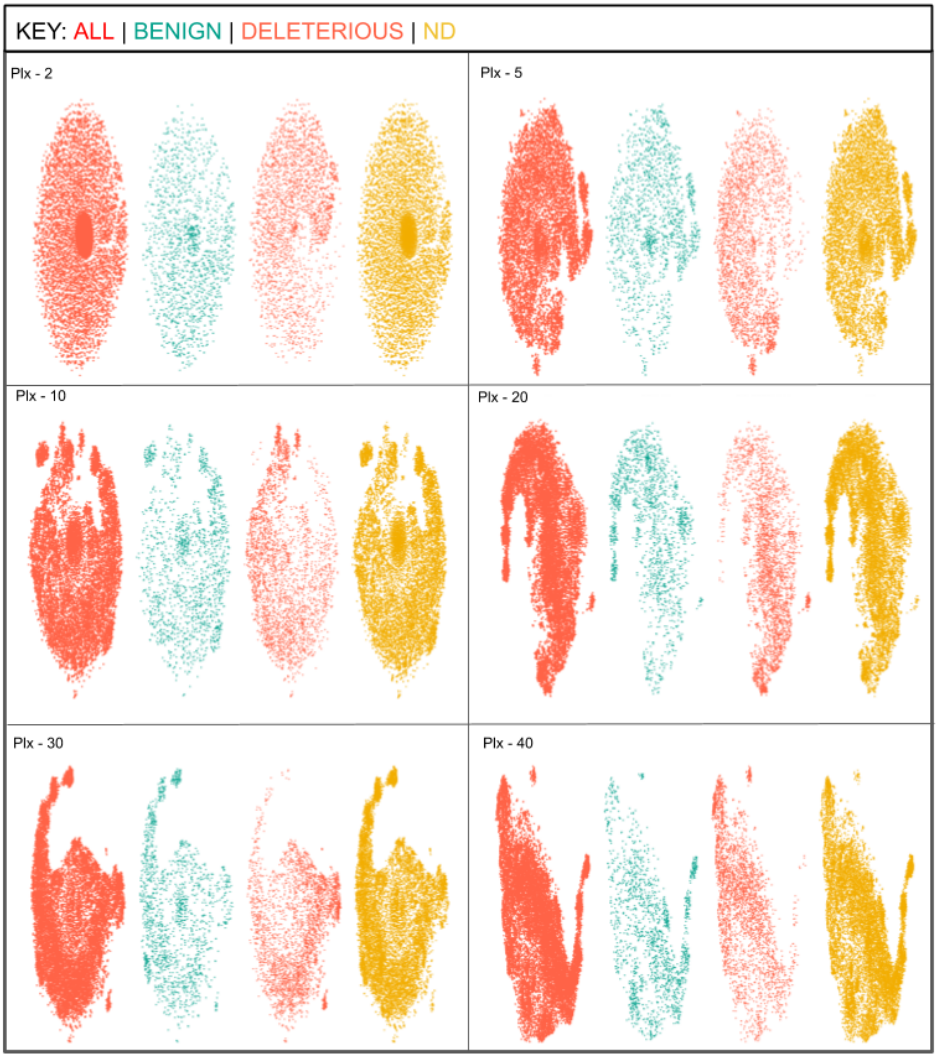



**Figure 3.9 - t-SNE visualisation of the Onto2Vec embedding of the Gene Ontology.** Gene ontology terms were embedded into 200 features, which are then projected as a 2-dimensional representation. Each panel displays the same data visualised with differing perplexities (plx). Only perplexities ranging from 2-40 are displayed, although perplexities up to 100 were explored. Other hyper-parameters were run with default settings. The colour relates to the phenotype as labelled, the order is maintained from left to right, as in the key.

Figure 3.9 contains the first run of each of the tested perplexity scores. We display them as the overall shape across all genes, and then a separate layer for each of the phenotypes. We find that as we increase perplexity the plots map the topology of the data differently, with shared motifs appearing between perplexities 2, 5 and 10; 20, 30, 40 and 100; and finally 60 and 80 (not all perplexities are shown). Overall, we observed relatively good stability between the varying parameter settings, with the shapes being similar (once inversions are accounted for), indicating that we are likely capturing some element of real topology, rather than arbitrary noise. With this in mind, a perplexity of 30

was chosen for further visualisation. Repeat runs and plots of the same perplexity scores runs were relatively stable, similarly indicating that the underlying data topology is not overly complex. Exceptions to this occurred at perplexities of 60 and 80, where sizeable differences in the output plots could be observed between multiple runs.

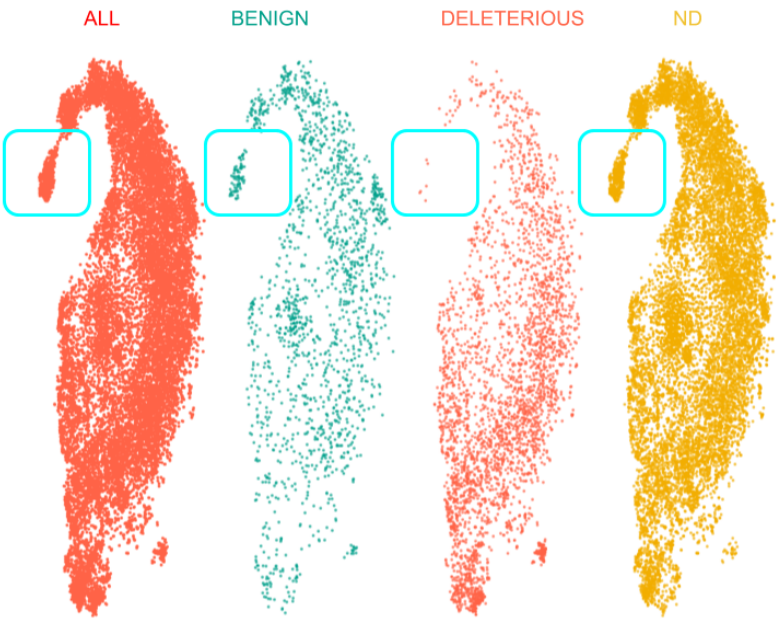



**Figure 3.10 - t-SNE visualisation of the Onto2Vec embedding of the Gene Ontology**. The different density of points across the different layers of the embedding show that functional annotation mirrors constraint, with areas of high constraint occupying different regions of this representation of gene function. Gene ontology terms were embedded into 200 features, which are then projected as a 2-dimensional representation. Perplexity was set at 30, with other hyper-parameters set to default settings. The colour relates to the phenotype as labelled. Teal boxes outline an exemplar of the differing density of points across the phenotypes.

Focusing on the 30 perplexity plot (Fig. 3.10), it is possible to see areas of different local density between phenotypes, such as in the areas outlined in Fig. 3.10 (the head of the 'sea-horse' structure). In these teal boxes we observe a dense collection of genes that are almost totally devoid of deleterious genes. The inverse of this is visible in the bottom-most right of the plot (the tail of the sea-horse), where relatively few genes are observed in the benign layer, with the greatest density of genes being observed in the deleterious layer. Each of the groups conform to the global shape of the ontology, indicating that all genes are present in all areas of the ontology, but local differences are visible. Similar plots were produced using LOEUF bin to create the

layers (see Fig. 3.11). This shows similar results, with local densities of the GO changing across bins from more to less constrained. As in Fig. 3.10, the head of the sea-horse is less populated in the more constrained layers, with density steadily increasing until the 5th LOEUF bin. The overall topology of the data for these plots looks different, however this is largely due to compression of the available plotting space in order to fit the multiple layers in.

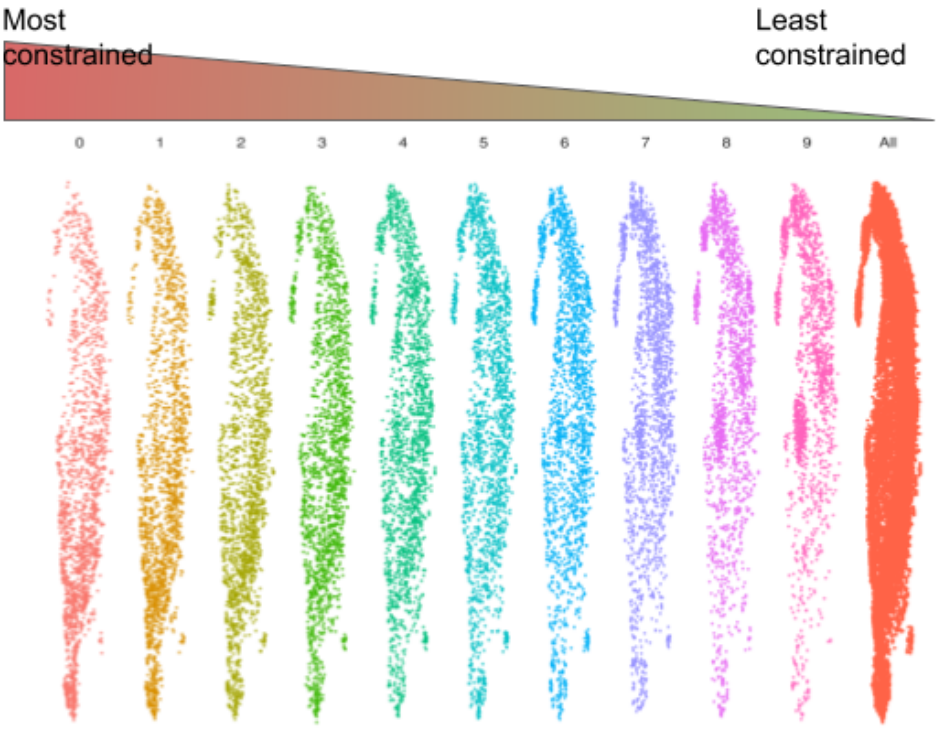



**Figure 3.11 - t-SNE visualisation of the Onto2Vec embedding of the Gene Ontology**. As in Fig. 3.10, we observe different densities of points as we move from more constrained bins to less constrained bins. This indicates a link between constraint and wider function. Gene ontology terms were embedded into 200 features, which are then projected as a 2-dimensional representation. Perplexity was set at 30, with other hyper-parameters set to default settings. Layers refer to LOEUF bins, with bins 0-9 followed by All bins combined from left to right.

The key message of these visualisations is that despite the examination of conceptually orthogonal data sources, the genetically derived LOEUF score and the functionally derived GO; we observe structure that is indicative of the linkage of these two data types. This indicates that the addition of such features to our model may help to explain variation in our data that are not purely genetically derived.

### 3.3.5 Classification models

We ran TPOT to classify ND genes within both the s-dS and l-dS datasets. Each model run took up to two weeks to complete, due to this only the iterations of the dataset previously described were tested. For each model we report the precision, recall and F1 score, a balanced average of the former scores. ROC AUC scores will not be reported due to the unbalanced nature of the classes in the dataset [17]. Please note that numbers we report for the l-dS and s-dS are dependent on data cleaning steps, therefore genes with incomplete data may be dropped resulting in fewer of any of the phenotypes in the reported results.

#### 3.3.5.1 Baseline models

We ran logistic regression and decision tree [18] algorithms as comparators to any eventual model suggested by TPOT. We did this explicitly for the benefit of downstream comparisons, but we highlight that these models are run as standard within the TPOT pipeline used. For this reason we ran TPOT for 25 generations on the l-dS only.
The logistic regression pipeline produced is a feedback classifier in which multiple iterations of logistic regression are stacked and added to the cleaned data before undergoing one final round of logistic regression (see Code block 3.1).

```
exported_pipeline = make_pipeline(
    make_union(
        StackingEstimator(estimator=make_pipeline(
            make_union(
                FunctionTransformer(copy),
                FunctionTransformer(copy)
            ),
            LogisticRegression(C=0.1, dual=False, penalty="l1")
        )),
        FunctionTransformer(copy)
    ),
    StackingEstimator(estimator=LogisticRegression(C=0.0001,
dual=True, penalty="l2")),
    LogisticRegression(C=5.0, dual=False, penalty="l1")
)
```

**Code block 3.1** - The output pipeline of a TPOT run in which possible models were restricted to logistic regression only.

The average balanced-accuracy cross-validation score for this pipeline was 0.908. Upon validation, scores of 0.626 and 0.627 were achieved for accuracy and recall respectively with an F1 score of 0.627 (Table 3.2).

Similarly, the decision tree pipeline produced is a stacking estimator in which multiple trees are run with outputs being fed back to the data before running a final decision tree for classification (see Code block 3.2). The average cross-validation balanced accuracy score achieved was 0.92, with accuracy and recall scores of 0.62 and 0.90 respectively. The F1 score was 0.737 (Table 3.2).

```
exported_pipeline = make_pipeline(
    make_union(
        make_union(
            StackingEstimator(estimator=make_pipeline(

StackingEstimator(estimator=DecisionTreeClassifier(criterion="entrop
y", max_depth=4, min_samples_leaf=18, min_samples_split=8)),
                DecisionTreeClassifier(criterion="gini",
max_depth=4, min_samples_leaf=2, min_samples_split=16)
            )),
            FunctionTransformer(copy)
        ),
        FunctionTransformer(copy)
    ),
    DecisionTreeClassifier(criterion="gini", max_depth=3,
min_samples_leaf=5, min_samples_split=8)
)
```

**Code block 3.2** - The output pipeline of a TPOT run in which possible models were restricted to decision trees only.

### 3.2.5.2 s-dS model

After 158 generations, TPOT converged on a Random Forest classifier [19]. The pipeline is fully outlined in Code block 3.3, but in brief first applies two recursive feature elimination (RFE) steps followed by a Random Forest classifier. Both RFEs use an ExtraTrees Classifier as the estimator, which in turn use 100 estimators (i.e. the number of trees) each considering 10% of the features. The bottom scoring 45% of the features are then dropped. The second round of RFE then uses a similar model but with 20% of the features tested in each estimator. The worst performing 50% of these

features are then removed. The remaining features are then used to train the Random Forest classifier.

```
exported_pipeline = make_pipeline(
    RFE(estimator=ExtraTreesClassifier(criterion="entropy",
max_features=0.1, n_estimators=100), step=0.45),
    RFE(estimator=ExtraTreesClassifier(criterion="entropy",
max_features=0.2, n_estimators=100), step=0.5),
    RandomForestClassifier(bootstrap=False, criterion="gini",
max_features=0.3, min_samples_leaf=5, min_samples_split=9,
n_estimators=100)
)
```

**Code block 3.3** - The model output of a TPOT run to identify a classifier for the s-dS.

The average accuracy resulting from 10 cross-fold validation with the training data was 0.95. We generated model metrics by applying the model to the 20% test set. We report an accuracy of 0.95, with precision and recall of 0.87 and 0.988 respectively. The F1 score was 0.925 (Table 3.2).

### 3.3.5.3 l-dS model

Model convergence was achieved after 173 generations, the suggested model for the l-dS dataset was a Gradient Boosting Classifier. The pipeline comprises several transformation and scaling steps, followed by the gradient boosting classifier (see Code block 3.4).

```
exported_pipeline = make_pipeline(
    make_union(
        make_pipeline(
            ZeroCount(),
            StandardScaler()
        ),
        FunctionTransformer(copy)
    ),
    GradientBoostingClassifier(learning_rate=0.1, max_depth=3,
max_features=0.15000000000000002, min_samples_leaf=2,
min_samples_split=12, n_estimators=100,
subsample=0.8500000000000001)
)
```

**Code block 3.4** - The model output of a TPOT run to identify a classifier for the l-dS.

The average 10-fold cross-validation accuracy score was 0.927. We again validated the model using the 20% validation set, achieving a precision and recall of 0.69 and 0.71 respectively, with a balanced F1 score of 0.702 (Table 3.2). Of the 4906 ND genes in this set, 442 were classified as benign, with a mean probability of 0.7 (Table 3.3). The mean LOEUF for this set was 1.33 vs 1.10 for ND genes predicted as not benign and 0.95 for the cleaned dataset as a whole.

### 3.3.5.4 No gnomAD model

We constructed a final model in which all gnomAD related data were removed from the input data. This reduced the number of features to 296. After 179 generations, the pipeline converged on an ExtraTreesClassifier based model (see Code block 3.5). Preprocessing of the features in this pipeline include the scaling of features according to their maximum absolute value followed by the filtering of low variance features and the rescaling of remaining features.

```
exported_pipeline = make_pipeline(
    MaxAbsScaler(),
    VarianceThreshold(threshold=0.005),
    MinMaxScaler(),
    ExtraTreesClassifier(bootstrap=True, criterion="gini",
max_features=0.3, min_samples_leaf=8, min_samples_split=19,
n_estimators=100)
)
```

**Code block 3.5** - The model output of a TPOT run to identify a classifier for the l-dS.

The average 10-fold cross-validation accuracy score was 0.915. We achieved a Precision and recall of 0.65 and 0.55 respectively with a balanced F1 score of 0.59 upon validation (Table 3.2).

| Model | Dataset | Metric | Score |
|---|---|---|---|
| Logistic regression | l-dS | Precision<br>Recall<br>F1 | 0.625<br>0.629<br>0.627 |
| Decision tree | l-dS | Precision<br>Recall<br>F1 | 0.627<br>0.896<br>0.737 |
| Ensemble model | l-dS | Precision<br>Recall<br>F1 | 0.691<br>0.714<br>0.702 |
| Ensemble/Random forest model | s-dS | Precision<br>Recall<br>F1 | 0.87<br>0.988<br>0.925 |
| ExtraTrees Classifier (no gnomAD) | l-dS | Precision<br>Recall<br>F1 | 0.65<br>0.55<br>0.59 |

**Table 3.2** - **The output metrics of the 5 different models resolved for the l-dS and s-dS.** Precision and recall are reported, alongside the F1 score, the harmonic mean of the precision and recall. The overall highest performing model is the Ensemble/Random forest model for the s-dS. Of the l-dS models, the best performing is the Decision tree, followed by the Ensemble model. Across all metrics, a score of 1 would indicate a perfect score. Each of the models referred to are described in more detail in Code blocks 3.1,3.2, 3.3, 3.4 and 3.5 respectively.

## 3.3.6 Gene predictions

Each of the l-dS models had the same number of ND genes with the exception of the 'no gnomAD' set. This is due to a number of genes having missing gnomAD data and therefore being filtered out of any dataset containing gnomAD data due to incompleteness. All l-dS models predict that 10% (range 9-12.1%) of the ND genes are benign (pBenign) with a mean certainty of 0.7 (range 0.7-0.72). The s-dS ensemble model labelled 1782 genes of a possible 13,081 genes as pBenign. The mean probability for prediction was 0.83 indicating much greater certainty in labelling (Fig. 3.12). A set of 692 genes were predicted with a probability of >0.9, of which 74% of l-dS ensemble model pBenign genes were present.

| Model | Dataset | Not Determined | Prediction pBenign/pNotBenign | Total (p)Benign Genes |
|---|---|---|---|---|
| Logistic regression | l-dS | 4906 | 522/4384 | 2266 |
| Decision tree | l-dS | 4906 | 595/4311 | 2339 |
| Ensemble model | l-dS | 4906 | 442/4464 | 2186 |
| Ensemble/Random forest model | s-dS | 13081 | 1782/11308 | 3510 |
| ExtraTrees Classifier (no gnomAD) | l-dS | 5196 | 506/4690 | 2250 |

**Table 3.3 - The breakdown of predicted benign genes**. For each of the five models, the 'Not Determined' column shows size of the test set of genes on which the models predicted pBenign status. The prediction column displays the outcomes of the prediction, with pBenign genes followed by pNotBenign genes. The 'Total (p)Benign Genes' column is the sum of model assigned pBenign genes, and initial label classification (as in Fig. 3.1).

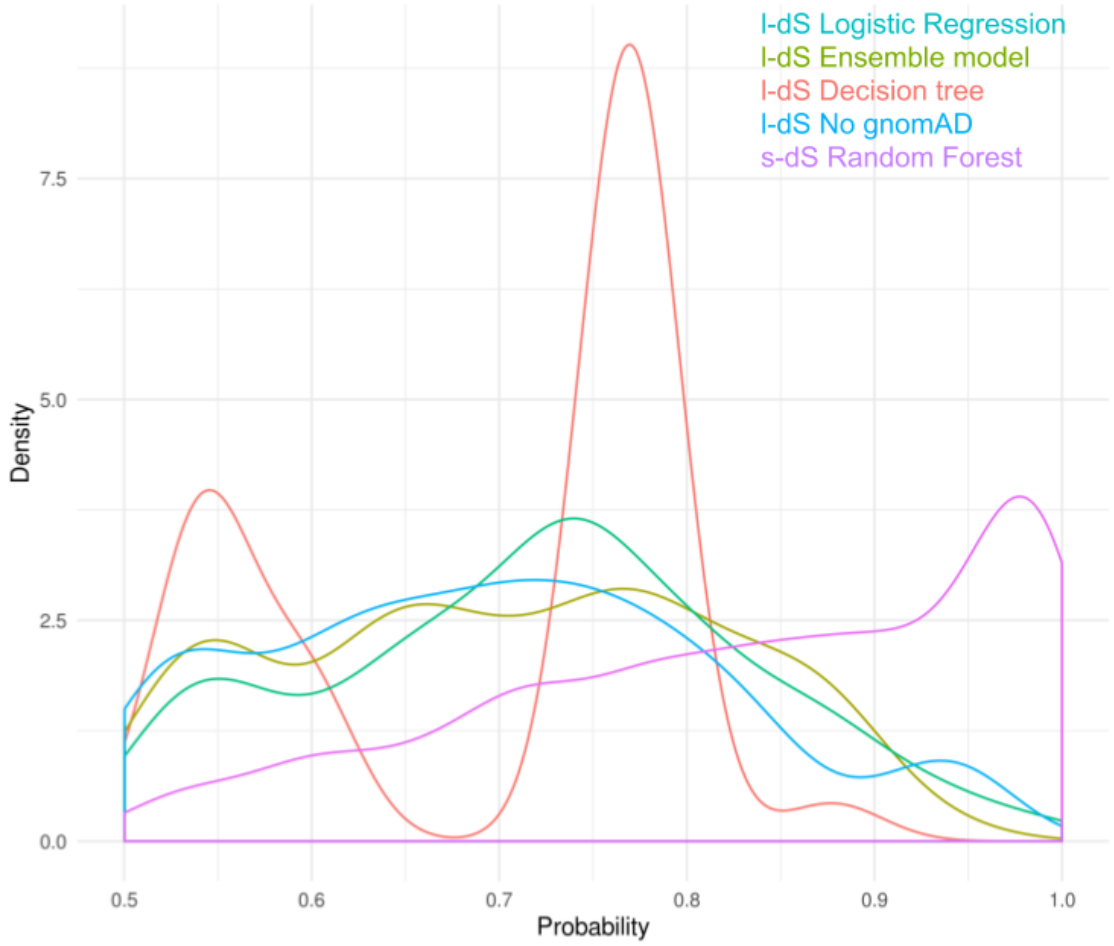



**Figure 3.12 - Density distributions of positively labelled predicted genes**. All distributions are derived from predictions made on ND genes. Probabilities of > 0.5 are classed as Benign.This reflects the certainty of a label classification, where genes of probability 1 show the model has complete confidence in the assignment. The variety of distributions show that modifying thresholds between models would yield different output numbers of genes. Few models outside of the s-dS Random Forest model have high certainty (>0.9) in the majority of their predictions, as is reflected in the output metrics (Table 3.2).

Gene parity is very high between the two highest performing models of the l-dS dataset, the Decision tree and the Ensemble model. 96% of pBenign genes from the Ensemble model are also found in the Decision tree (Fig. 3.13). The l-dS Ensemble model keeps relatively high similarity with all models, with 82% parity with the Logistic regression and No gnomAD models. We also see high similarity between all the l-dS models and the s-dS Random Forest model, with 441 of 442 pBenign genes in the l-dS Ensemble model also labelled as pBenign in the s-dS model. A further 75% of these are found within the high probability (>0.9) pBenign genes of the s-dS model. We find this to be the case across all l-dS models.

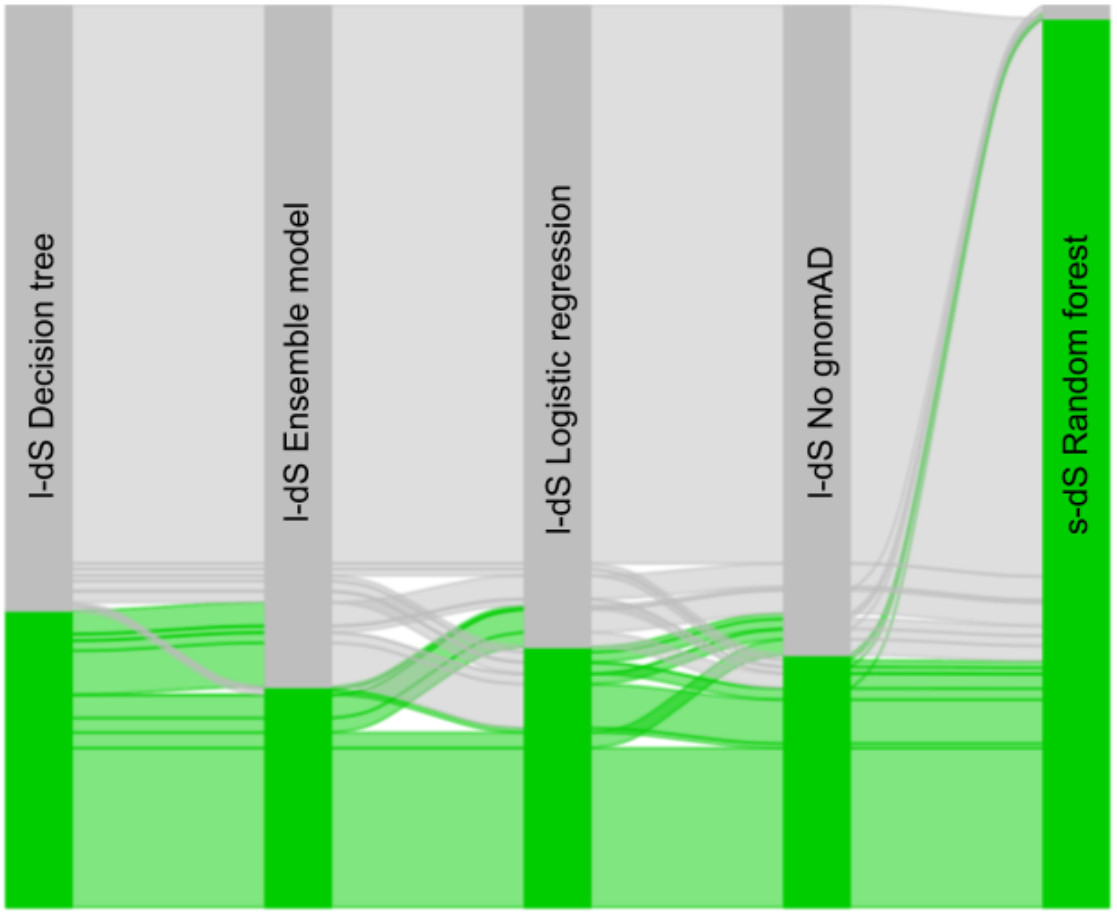



**Figure 3.13 - Alluvial plot showing shared gene membership between models.** Each line represents a gene where green lines are pBenign genes as predicted from the model on the left of the line. Genes not present in the left hand model are marked as grey. All genes represented in this plot were labelled as pBenign in at least 1 model. There is high consistency between models, with nearly all genes predicted as pBenign also being predicted as such in the s-dS model. Within the l-dS models, the most conservative model is the Ensemble model, with the most pBenign genes predicted by the Decision tree.

## 3.3.7 l-dS Ensemble Model gene characteristics

Of the 442 pBenign genes in the l-dS Ensemble Model, 120 are members of the druggable genome. Reactome pathway analysis of the 172 pBenign genes that could be mapped to the Reactome database show relatively widespread coverage of pathways within the gene sets (Fig. 3.14.1). However significant enrichment (FDR < 0.05) is much more localised to just 5 immune system specific pathways (Table 3.4, Fig. 3.14.2). Examination of the same gene set using The Database for Annotation, Visualization and Integrated Discovery (DAVID) shows significant enrichment for olfactory receptors across several other pathway-based databases including KEGG and Interpro (FDR < 0.001).

| Pathway name | Entity pathway overlap | p-value | FDR |
|---|---|---|---|
| Endosomal/Vacuolar pathway | 15/82 | 5.4E-07 | 2.3E-04 |
| Antigen Presentation: Folding, assembly and peptide loading of class I MHC | 15/102 | 7.5E-06 | 1.6E-03 |
| Antigen processing-Cross presentation | 18/187 | 2.3E-04 | 3.2E-02 |
| Interferon gamma signaling | 21/250 | 4.3E-04 | 3.2E-02 |
| ER-Phagosome pathway | 16/165 | 4.6E-04 | 3.2E-02 |
| Interferon alpha/beta signaling | 17/164 | 5.3E-04 | 3.2E-02 |

**Table 3.4 - The 5 Reactome pathways for which an FDR < 0.05 is achieved.** The entity pathway overlap shows the number of genes from our dataset that overlap with the pathway superset of genes.

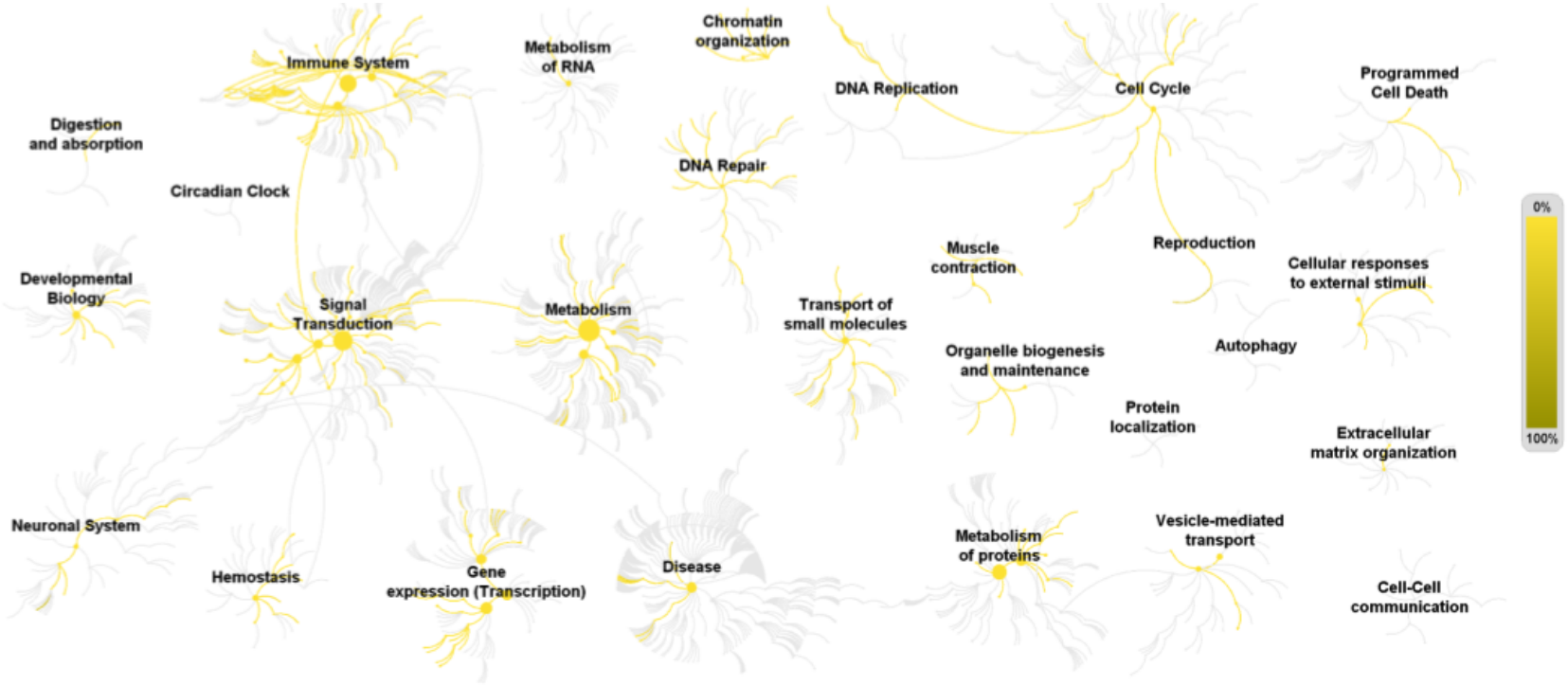



**Figure 5.14.1 - Reactome pathway overview of overrepresented pBengin genes**. Reactome pathways, labelled by overarching functional groups are labelled yellow if pBenign genes are enriched within the pathway

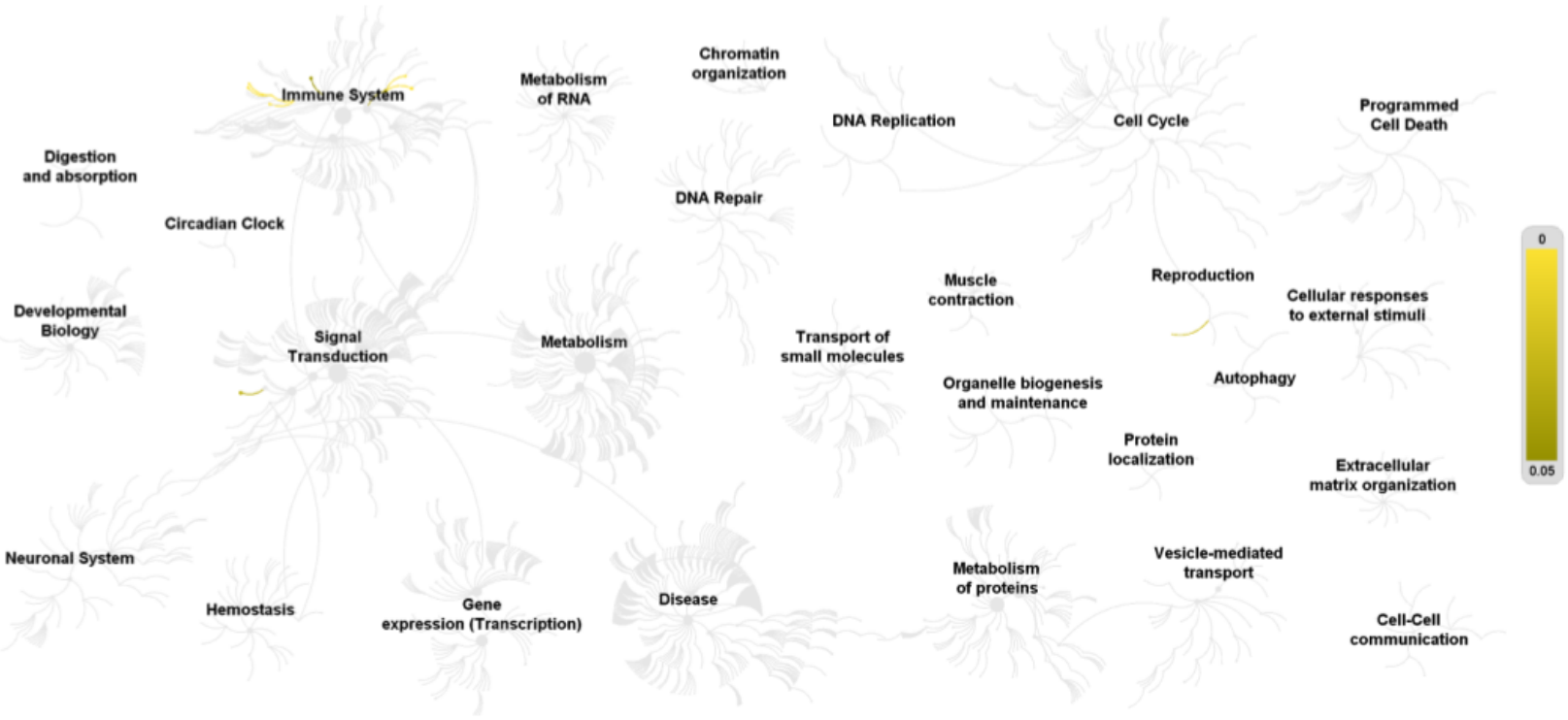



**Figure 5.14.2 - Reactome pathway overview of overrepresented pBengin genes**. Reactome pathways, labelled by overarching functional groups are labelled yellow if pBenign genes are significantlyt enriched within the pathway with an FDR < 0.05

68% of genes are labelled as ‘Tdark’ by the Pharos database, indicating that little information is known about these targets. Of the remainder, 21% have limited evidence of some compound interaction or meet the threshold of having either GO leaf terms or an associated OMIM phenotype. Only ~5% of pBenign genes have existing therapies targeting them, or strong evidence of compounds with therapeutic action. The final 6% of pBenign genes could not be mapped.
Mean centrality measures indicate that genes in this subset are less connected than the pNotBenign genes and the global (all genes across the dataset) median degree (degree centrality 3 v 5 and 5 respectively). This is true for all measures of centrality tested (data not shown).
The median O/E score reflected this pattern, with pBenign genes having a considerably higher median value of 0.84 compared to 0.54 and 0.48 for pNotBenign and global respectively.
Across all databases with the exception of the PharosDB, considerable proportions of genes were lost due to lack of representation within the database.

### 3.3.8 s-dS Random Forest gene characteristics

The s-dS Random Forest model labels a total of 1,782 genes as pBenign. Of these, 615 are known to be members of the druggable genome. Analysis of this gene set using Enrichr reveals no significant associations (after Benjamini-Hochberg correction) to either GWAS catalog 2019 traits, UK biobank GWAS v1 traits, DisGeNet diseases or any OMIM diseases.
The median O/E score was 0.77 and 0.837 for the pBenign and high confidence pBenign genes respectively, and 0.402 for pNotBenign.

### 3.3.9 Feature importance estimation

Following the creation of a simplified random forest approach for the l-dS dataset, the feature importance was estimated. This showed the most important feature to be ‘atleast1Hom’, an indicator of whether a gene is found in a homozygous pLoF state or not. This was the most important feature by a wide margin. Running a similar model, omitting this variable leads to a decrease in accuracy, with a notable increase in misclassification of benign genes as deleterious. However, more variants are mislabelled from deleterious to benign when not including “atleast1Hom”, indicating

that we actually reduce false positives when ignoring this feature (Appendix table 1). Due to the fact that this is a different model that has undergone different feature selection and engineering steps, it is difficult to link this to the models previously described in this chapter. Plots displaying summaries of the feature importance across both model types (with and without atleast1Hom), for the 30 most important features can be found in the supplementary data (Appendix figure 3.3.9).

# 3.4 Discussion

## 3.4.1 Predicting pLoF pathogenicity

Whilst there are strong indications that LOEUF and various biological factors relating to pathogenicity are correlated, it is clear that LOEUF alone does not tell the whole picture. This is exemplified by the fact that 310 genes are found in a homozygous pLoF state within various sequenced cohorts, despite being constrained (o/e LoF < 0.5), with 45 being severely so (o/e LoF < 0.1). There are numerous reasons for why such examples exist. Firstly, without deep curation of each of these genes, it is not possible to say that these are definitely confirmed LoF variants. For example, the gene Filaggrin is identified as a rhLoF harbouring gene from the BiB cohort. Two variants are found, with a total of 2 and 26 ALT alleles counted. Both of these are reported in the gnomAD cohort, with both being presented as low confidence pLoF, due to proximity to the gene terminus (recall from chapter 2, pLoF found in a terminal region are less likely to induce functional LoF). Additionally one of the variants is only found in a single instance in heterozygous form, and whilst the latter is actually common within the Finnish population (MAF ~ 2%), it is found on a non-coding transcript. Both of these variants are examples of pLoF that are unlikely to be functional LoF inducing. Secondly, there may be compensatory mechanisms at play, such as additional rescue variants, or novel nonsense-mediated decay mechanisms [20]. In addition to this, the ability to fully examine the pLoF burden within human populations is challenging due to the sheer scale of the necessary sample size. The LOEUF metric is currently underpowered to detect constraint in around 30% of coding genes, primarily due to gene length [15]. As previously mentioned, targeting consanguineous and bottlenecked populations will increase the probability of identifying the full spectrum of constraint [4]. However, such projects are costly and will take years to complete. It is for this reason that finding other measures that may indicate which genes are tolerant to inactivation may be beneficial.

To do this, it is important to try to consider as broad a range of possible contributing data sources as is feasible. It is probable that other layers of the biological dogma will be indicative of what is occurring at the genetic level. Such indicators, from areas such as protein-protein interactions and tissue expression could provide a rough estimate of the likely effect of pLoF in genes for which the ground truth is unavailable. However it is clear that such efforts may suffer from some of the same inherent biases in the data driven by undersampling of an array of ethnic groups. Or put differently, the overrepresentation of a single group.

In order to simplify prediction slightly, we sought solely to attempt to identify proteins for which no phenotype would be expected in spite of total LoF. We emphasise that the binary classification of Benign and Deleterious is reductive, and especially, that genes not predicted to be Benign should *not* be seen as truly Deleterious. It is for this reason that we have defined negative labels as pNotBenign.
As the overarching intention in this project is to find suitable and safer drug targets, it is important to note that the reality is that a degree of negative impact from drug perturbation of a target is tolerable, as long as this effect represents a net gain in a patient's quality of life and/or disease management. We chose such a narrow group of genes, namely those for which there is strong evidence of tolerated LoF observed within 1 or more humans, in order to try to identify those genes that are least likely to cause side effects. However, it should not be inferred from this that the remaining genes should not be examined for drug targeting and, as we show in [4], most successful drug targets are in fact constrained, or associated with negative phenotypes.

Defining Benign pLoF within the data is not necessarily trivial. We had to draw several assumptions in order to proceed. First of all, it is certain that a number of the variants found within any of the cohorts are spurious [14,15]. Without follow-up of detected pLoF, including manual curation of pLoF variants, functional assays and deep-phenotyping [21], it is not possible to be certain of the assertion that these pLoFs are real. Our work in Minikel et al. (Nature, 2020) provides a clear example in the examination of MAPT pLoF variants. MAPT gain of function variants are implicated in tauopathies, and drugs are in active development for this target. Therefore this serves as a case study of where knowledge of LoF may lead to a natural experiment to assess the viability of this target. However, when examining variants found within MAPT, all pLoF variants either occurred in exons not expressed in the brain (based on data from GTEx), or were the result of identifiable annotation errors [4].

This example illustrates the issue with the generation of our LoF phenotypes categories. A mitigating factor in this is that the papers used to define Benign pLoF all used LOFTEE, and therefore the probability of false positives was reduced [15]. However versions of LOFTEE used were older than that reported in Karcweski et al. (2020), and therefore it is likely that error rates of greater than 25% still persist. One of the largest differences is the lack of pext score based filtering [22]. In spite of this, such an approach is the only available way to attain a list of human-derived homozygous pLoF data.

The inclusion of the GO data allows for the contextualisation of single gene data into the greater surrounding system, such as cellular pathways and functional groups [23]. This is exemplified by the t-SNE visualisations in Fig. 3.10, where the differing phenotypes exhibit differing local patterns of density whilst maintaining the same overall global shape. We infer from this that all areas of the GO are covered by genes of each phenotype, but it is clear that there are some functional groups with much less tolerance to perturbation. This is further exemplified by the LOEUF bin view of this data (Fig. 3.11), where we observe more gradual gradation in said densities. As previously discussed, GO annotations have been used in a wide range of predictive contexts, such as gene function prediction [24–26], drug discovery [27] and disease gene identification [28–30]. This draws us to conclude that GO and possibly other functional annotations may be informative in discerning genes tolerant to inactivation. In future efforts, it may be worthwhile to embed other ontologies and include these as features within the input feature set, however steps would have to be taken to prevent excessive collinearity between features (or at least account for them in any feature selection and engineering steps).

Ensemble models can be powerful ways of increasing performance in complex datasets, however they are inherently poor at providing explainability in a model. Explainability refers to the ability to trace back from a model prediction to the features that led to that prediction being made. This is something that can glean useful insights into our dataset. Take for the example the case where we are trying to predict which plants will grow best in a garden based on knowledge of related species of plant. The model telling us that the amount of sunlight in the afternoon is a strong predictor of the plant's health is a useful insight that reveals further information about the data that might not have been immediately obvious (albeit in this example you would imagine that *was* an obvious one).
Our dataset comprises 316 features, and it is almost certain that these features will not contribute equal weight in the prediction problem at hand. It is for this reason that we

decided to create another, simplified random forest. Here we didn't include feature engineering steps, which construct abstractions of the original data. However, this resulted in reduced model performance, and the results from this model have to be treated with caution. The features that are important in this model will likely be related to those that are important in the random forest ensemble model, but we do not assert in any way that they are the same. With these caveats stated, it is interesting to note that the single most important feature of this simplified model was the 'atleast1Hom' feature, a binary indication of whether a homozygous LoF variant had been identified in any of the datasets considered. This is a relatively intuitive finding, as the rarity of such variants is likely a driving confounding variable. As previously described, homozygous LoF variants are rare, and all of our benign LoF labelled genes by definition must be a 1 for 'atleast1Hom'. This is more of a problem within the l-DS than the s-DS due to the imbalance nature of the l-DS set. However, in considering this, it is also important to note that genes predicted benign by these models are generally also predicted by so by the s-DS random forest approach. This model, with an F1-score of 0.93 predicts 441 of the 442 genes predicted as benign by the l-DS ensemble model. Additionally, examination of the confusion matrices produced in our simplified random forest models suggests that error drives misclassification of benign genes as deleterious. Due to our emphasis being on the correct identification of benign genes, false negatives are less problematic.

Future iterations of this work would surely require us to first update the list of benign genes to mirror the increasing sequencing-based knowledge available, and specify more features that may be of use. We would especially like to include features such as network specificity, and identify approaches to integrating group network clustering algorithms data into the prediction model. We would also seek to include more up to date sources of LoF data. We limited the choices to papers already published prior to January 2019, and therefore there would be scope to expand upon this dataset with more up to date data.

### 3.4.2 Network analysis

As significant genetic data is imparted by the use of the LOEUF score, we sought to broaden the range of data by including network metrics. We built these measures based on stringDB protein-protein interactions. Through this approach we see that some measures are better at differentiating between the different phenotypes to others.

Previous studies have shown the utility of degree centrality metrics in biological inference, whether it pertains to cell essentiality [31–34], disease gene identification [35] and correlation to constraint [3,14,15]. However, the applicability of many other network approaches to biology have also been demonstrated. This includes other measures of centrality such as closeness [36] and betweenness [37] [32,35,38,39].
Whilst we have explored a greater range of network centrality measures - it remains clear that more measures remain untouched. We believe that these should be explored, and that special attention should be paid to measures such as integration centrality, that lend weight to total network dynamics, rather than purely locally based measures such as degree centrality. In addition to these, measures of network modularity require proper integration into the featureset, although the value of these is currently unclear, and therefore this would be on a more exploratory basis.

### 3.4.3 Gene predictions

We generated several models based on three iterations of data. From these iterations we report five models, three based on the l-dS, 1 on the s-dS and 1 on the l-dS in which gnomAD data has been removed. As the classes are unbalanced, we reported precision and recall scores with a balanced F1-score to summarise these (Table. 3.2). The s-dS ensemble model clearly out-performs all other models, however this is likely indicative of the nature of the problem it is being asked to solve. The strict definition of the deleterious genes in the s-dS means that we are essentially defining our positive and negative labels as the extremes of the distribution. That is to say, the positive labels, tolerated homozygous pLoF genes are the best case scenario, and the negative labels, Mendelian disease genes and essential genes are the worst, in which damaging effects are all but certain upon knockout of the gene. This is reflected in the probability of prediction for the labels where we see ~40% of genes predicted with a probability of > 0.9 and ~70% at > 0.75 (Fig. 3.12). Such certainty in labelling is in stark contrast to what we see when examining the l-dS Ensemble classifier data, where only 39% of genes have a probability of > 0.75. This implies that the data are less clear here, as we would expect from introducing a wider range of damaging effects. However we feel doing so is important due to the question at hand; to identify those genes that will not result in negative phenotypes upon total or partial inhibition.

As is exemplified by the poor scores achieved upon removal of gnomAD data (Table 3.2), the inclusion of O/E data (and allele frequencies and population specific

frequencies) is informative for classification. Using this score as an indicator of how closely we replicate the original set Benign genes serves as a rough measure of how successfully we have labelled the unlabelled genes. Whilst we observe close similarity between the deleterious sets for the l-dS and s-dS, there is a marked difference in the O/E scores for the ND sets, with the s-dS ND closely mirroring the global O/E score of 0.48, versus the l-dS ND set which is markedly less constrained. As we have previously stated, we posit that this indicates better capturing of genes which would cause negative phenotypes upon knockout.

Both models predict pBenign genes with a median O/E score that is similar to that of the Benign genes (as defined by actual observation of homozygous knockout in humans). Whilst the O/E ratio is a useful indicator, we caution against overinterpretation of information derived from this, as the ratio is explicitly a measure of tolerance to heterozygous, not homozygous pLoF inactivation.

## 3.5 Conclusion

The identification of homozygous LoF genes in healthy individuals could lead to a new source of otherwise unexplored targets for drug development. Such genes may provide safe targets with fewer side effects and a greater chance of reaching the market. The accumulation of current knowledge shows that we have already uncovered more than 1,700 such genes of which 204 are predicted druggable or Biopharmable, with more to be discovered as sequencing efforts are expanded globally.

In this and the previous chapter we have covered many areas relating to LoF specifically. We have seen how the measure of tolerance to heterozygous pLoF, LOEUF, is a functionally relevant metric with effects that can be seen at the protein level in addition to the genetic.

Following from this, we have explored the possibility of using features such as LOEUF amongst many others to try to predict genes in which homozygous pLoF will be tolerated. The underlying assumption in this endeavour is that such genes will make attractive drug targets. We describe several such models, based on different versions of data. The best performing model overall is a Random Forest model, although we feel that this model and the accompanying dataset is less reflective of the hypothesis we are aiming to test. For this reason, we promote the Ensemble model built on the data with a less stringent definition for deleterious genes. This model proposes an additional

442 pBenign genes. Of this group, a further 180 genes are predicted druggable, nearly doubling the pool of theoretically ‘safe’ drug targets.

The information preceding is built on the premise that the identification of homozygous LoF containing genes will lead to opportunities in drug development. The following chapter will examine the utility of these assignments as predictors of drug development success, and therefore aim to clarify whether there is merit in this approach.

# 3.6 References


1. King, E. A., Wade Davis, J. & Degner, J. F. Are drug targets with genetic support twice as likely to be approved? Revised estimates of the impact of genetic support for drug mechanisms on the probability of drug approval. *bioRxiv* 513945 (2019) doi:10.1101/513945.
2. Nelson, M. R. *et al.* The support of human genetic evidence for approved drug indications. *Nat. Genet.* **47**, 856–860 (2015).
3. Narasimhan, V. M. *et al.* Health and population effects of rare gene knockouts in adult humans with related parents. *Science* **352**, 474–477 (2016).
4. Minikel, E. V. *et al.* Evaluating potential drug targets through human loss-of-function genetic variation. *Nature* 459–464 (2020).
5. Olson, R. S. *et al.* Automating Biomedical Data Science Through Tree-Based Pipeline Optimization. in *Applications of Evolutionary Computation* (eds. Squillero, G. & Burelli, P.) vol. 9597 123–137 (Springer International Publishing, 2016).
6. Forrest, S., Nguyen, T., Weimer, W. & Le Goues, C. A Genetic Programming Approach to Automated Software Repair. in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation* 947–954 (ACM, 2009).
7. Spector, L., Clark, D. M., Lindsay, I., Barr, B. & Klein, J. Genetic Programming for Finite Algebras. in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation* 1291–1298 (ACM, 2008).
8. Karczewski, K. J. *et al.* Variation across 141,456 human exomes and genomes reveals the spectrum of loss-of-function intolerance across human protein-coding genes: Supplementary Information. *Genomics* (2019).
9. Sulem, P. *et al.* Identification of a large set of rare complete human knockouts. *Nat. Genet.* **47**, 448–452 (2015).
10. Saleheen, D. *et al.* Human knockouts and phenotypic analysis in a cohort with a high rate of consanguinity. *Nature* **544**, 235–239 (2017).

11. Smaili, F. Z., Gao, X. & Hoehndorf, R. Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations. *Bioinformatics* **34**, i52–i60 (2018).
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *arXiv [cs.CL]* (2013).
13. Karczewski, K. J. *et al.* Variation across 141,456 human exomes and genomes reveals the spectrum of loss-of-function intolerance across human protein-coding genes. *bioRxiv* 531210 (2019) doi:10.1101/531210.
14. MacArthur, D. G. *et al.* A systematic survey of loss-of-function variants in human protein-coding genes. *Science* **335**, 823–828 (2012).
15. Karczewski, K. J., Francioli, L. C., Tiao, G. & Cummings, B. B. The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature* 434–443 (2020).
16. Maaten, L. van der & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
17. Saito, T. & Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One* **10**, e0118432 (2015).
18. Quinlan, J. R. 10.1023/A:1022643204877. *Machine Learning* vol. 1 81–106 (1986).
19. Breiman, L. 10.1023/A:1010933404324. *Machine Learning* vol. 45 5–32 (2001).
20. El-Brolosy, M. A. *et al.* Genetic compensation triggered by mutant mRNA degradation. *Nature* vol. 568 193–197 (2019).
21. Zhao, Z. *et al.* Molecular characterization of loss-of-function mutations in PCSK9 and identification of a compound heterozygote. *Am. J. Hum. Genet.* **79**, 514–523 (2006).
22. Cummings, B. B. *et al.* Transcript expression-aware annotation improves rare variant interpretation. *Nature* **581**, 452–458 (2020).

23. The Gene Ontology Consortium & The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research* vol. 47 D330–D338 (2019).
24. Zhao, Y., Fu, G., Wang, J., Guo, M. & Yu, G. Gene function prediction based on Gene Ontology Hierarchy Preserving Hashing. *Genomics* **111**, 334–342 (2019).
25. Li, Z. *et al.* Gene function prediction based on combining gene ontology hierarchy with multi-instance multi-label learning. *RSC Advances* vol. 8 28503–28509 (2018).
26. Cheng, L., Lin, H., Hu, Y., Wang, J. & Yang, Z. Gene Function Prediction Based on the Gene Ontology Hierarchical Structure. *PLoS ONE* vol. 9 e107187 (2014).
27. Mutowo, P. *et al.* A drug target slim: using gene ontology and gene ontology annotations to navigate protein-ligand target space in ChEMBL. *J. Biomed. Semantics* **7**, 59 (2016).
28. Asif, M., Martiniano, H. F. M. C. M., Vicente, A. M. & Couto, F. M. Identifying disease genes using machine learning and gene functional similarities, assessed through Gene Ontology. *PLoS One* **13**, e0208626 (2018).
29. Chen, J., Aronow, B. J. & Jegga, A. G. Disease candidate gene identification and prioritization using protein interaction networks. *BMC Bioinformatics* **10**, 73 (2009).
30. Ata, S. K. *et al.* Integrating node embeddings and biological annotations for genes to predict disease-gene associations. *BMC Syst. Biol.* **12**, 138 (2018).
31. Levy, S. F. & Siegal, M. L. Network hubs buffer environmental variation in Saccharomyces cerevisiae. *PLoS Biol.* **6**, e264 (2008).
32. Zotenko, E., Mestre, J., O'Leary, D. P. & Przytycka, T. M. Why do hubs in the yeast protein interaction network tend to be essential: reexamining the connection between the network topology and essentiality. *PLoS Comput. Biol.* **4**, e1000140 (2008).
33. Estrada, E. Virtual identification of essential proteins within the protein interaction network of yeast. *Proteomics* **6**, 35–40 (2006).

34. Jeong, H., Mason, S. P., Barabási, A. L. & Oltvai, Z. N. Lethality and centrality in protein networks. *Nature* **411**, 41–42 (2001).
35. Ozgür, A., Vu, T., Erkan, G. & Radev, D. R. Identifying gene-disease associations using centrality on a literature mined gene-interaction network. *Bioinformatics* **24**, i277–85 (2008).
36. Sabidussi, G. The centrality index of a graph. *Psychometrika* **31**, 581–603 (1966).
37. Freeman, L. C. A Set of Measures of Centrality Based on Betweenness. *Sociometry* **40**, 35–41 (1977).
38. Hahn, M. W. & Kern, A. D. Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks. *Mol. Biol. Evol.* **22**, 803–806 (2005).
39. Joy, M. P., Brock, A., Ingber, D. E. & Huang, S. High-betweenness proteins in the yeast protein interaction network. *J. Biomed. Biotechnol.* **2005**, 96–103 (2005).

# Chapter 4: Leveraging loss-of-function genetic variant data from population cohorts for drug target prioritisation

# Chapter commentary

The work in chapter 3 sought to catalog benign pLoF in sequenced cohorts, and then predict more benign genes.The initial motivation for this work was that such benign and pBenign genes would make better drug targets due to their tolerable safety profiles. This hypothesis was guided by the observation that genes harbouring LoF mutations were twice as likely to succeed from phase 1 to clinical approval [1]. Seeking to expand on this finding, and in order to test the validity of the machine learning labels we introduced in the last chapter, we sought to ascertain the probability of success of drugs in clinical trials based on LoF phenotype. We also expanded this into a more general exploration of the use of pLoF genetic information for drug discovery. The work following was completed in collaboration with the dermatological research team at AbbVie in Worcester, MA, with all work described herein being the product of that collaboration. Work from this chapter were also combined to produce a paper, sharing the name of this chapter (manuscript in preparation).

# 4.1 Introduction

## 4.1.1 Using loss-of-function variation data to aid drug target selection

Genetics has proved to be a powerful tool to inform drug discovery, offering direct biological insights from humans to complement those from model organisms. The analysis of historic drug data development data by Nelson et al. (2015) and King et al. (2019) both show that drug-targets genetically associated with disease are more than twice as likely to obtain market approval [2,3]. More specific evaluation of drug-targets harbouring rare homozygous loss of function (rhLoF) variants reveals a similar association, with such targets being twice as likely to reach approval from phase 1 trials (11.4% versus 6.7%, $\chi 2$ test; p = 0.046) [1]. Here, we further delineate this relationship, and explore whether the phenotype labels we generated in chapter 3 yield similar results. We use the data from the l-dS model for the duration of this chapter, and any reference to predicted benign or pBenign data refer to these data.

LoF variants, arising from stop-gained, essential splice and frameshift variants, provide insight to the essentiality of a drug-target. The deleterious nature of such variants means they are generally rare as variants that reduce fitness are selected against. However, with increasingly large-scale sequencing projects the prospect of an accumulating and ever more comprehensive catalogue of such variation is imminent [4–6]. LoF variation has a high potential to inform drug discovery, by serving as a proxy for lifelong partial or complete inhibition of a protein target [7], thus providing us with evidence of potential safety profiles. Genes harbouring LoF variants with no associated negative phenotype, are consistently over-represented among targets with a successful history of drug development and hence are more likely to represent safer targets for long term modulation [1].

Here we investigate the value of rhLoF information in the context of drug discovery, using our preliminary catalog of the known homozygous predicted LoF (pLoF) variation observed in sequenced populations (see chapter 3 for details) in addition to our predicted genes that may harbour benign rhLoF variation. We examine the probability of success in drug development associated with these data, and report that benign pLoF data is predictive of drug target success in non-oncological indications (ordered logistic regression, beta 0.41, p-value = 3.4e-4) to a greater extent than genetic data previously studied, and other measures of constraint [2,3,7]. We also highlight that

underexplored groups of protein-coding genes, in particular olfactory receptors (ORs), may provide functionally relevant targets with favourable safety profiles. Our findings support the wider use of rhLoF data for the selection of targets with an improved chance of success in drug discovery.

## 4.1.2 Looking beyond canonical G-coupled protein receptors (GPCRs) as drug targets

GPCRs are the most widely studied and drugged target family, serving as the primary targets of 34% of all FDA approved drugs [8]. The interest in this family is driven by their involvement in a wide-range of physiological processes as key signalling proteins. GPCRs are the largest of the membrane-protein families, with over 800 characterised so far, and interact with a large range of ligand types [8–10]. Of these proteins, a large proportion are olfactory receptors (ORs), accounting for around 400 genes (with a further 600 pseudogenes) [11]. ORs gained their name due to their role in our sense of smell and the historic belief that they were solely expressed in olfactory tissues [12,13]. However, further study into ORs has revealed widespread ectopic expression, as well as involvement in human disease (this will be examined in more detail within this chapter). Of the remaining GPCRs, 108 are the targets of approved drugs, with each target having a median of 4 unique approved agents [8,14]. This signifies a saturation of these targets has been achieved, and that other GPCRs should be explored [15,16].

One of the most significant hurdles in designing drugs for GPCRs is cross-reactivity, caused by the drug binding with other GPCRs resulting in adverse effects [17]. This issue is also observed in the targeting of other protein super-families such as kinases and is largely driven by the high degrees of sequence homology observed in these highly diversified families. The core mechanism of action of GPCRs remains the same; the extracellular domain of the protein binds to a ligand, causing a conformational change in the protein that exposes a guanine nucleotide exchange factor. This then activates the associated G-protein, which propagates downstream signalling. Whilst the ligands and G-proteins are hugely varied, the fundamentals are not, and therefore the active sites responsible for this activity are highly homologous. In addition, functional effects of GPCRs are difficult to delineate, a point highlighted by the increased observation of pleiotropy across multiple signalling pathways [18–21]. These similarities in structure and

overlap of function make targeting the active sites of the protein challenging, as selectivity is difficult to accomplish. Clinical data suggests that GPCRs are the most enriched of the 5 major target protein-families for side effects [22]. A significant proportion of these effects are caused by inter-paralog cross-reactivity, much as with kinases [17]. In order to ameliorate this problem, more specific drugs must be created, however doing this in the narrow band of GPCRs currently targeted will be challenging [15,16]. Therefore other, less studied, more functionally divergent GPCRs should be investigated in the hopes that allosteric inhibitors exploiting sequence differences between homologs can be found [23].

# 4.2 Methods

All methods relating to the generation of the LoF dataset are fully described in chapter 3. This includes the generation of the pLoF variation dataset, with additional enrichment of features and the generation of predictive benign pLoF models (see Chapter 3.2).

## 4.2.1 Statistical modeling of LoF benign genes and drug target approval

We used the drug target approval dataset and genetic evidence dataset assembled by King et al. (2019). In brief, a latest historical development phase was inferred for target-indication pairs that are not in active development, as listed in the Pharmaprojects database. Information on causal links between targets and diseases were obtained from GWAS catalog (MacArthur et al. 2017) and OMIM and were linked to Pharmaprojects indications based on medical subject heading (MeSH) similarity. To evaluate the clinical performance of LoF benign genes we selected all targets with the latest development stages of Phase I, Phase II, Phase III, and Approved. 589 unique indications were represented in this dataset, including 78 oncology indications. We then annotated all genes in this drug target dataset of pLoF benign status. Based on manual annotation we have 435 and 5592 target-indication pairs with a benign or deleterious gene respectively. Based on statistical prediction of benign pLoF labels we have 65 and 1589 target-indication pairs with a benign or deleterious gene respectively.

We first separated oncology from non-oncology indications and manually annotated benign pLoF genes from the supervised model predictions. To model the ordered progression from Phase I to Approval we fitted ordered logistic regression models on these four subsets of the data using benign LoF status, genetic support and LOEUF as covariates (Table 4.1). To improve statistical power, we also ran an ordered logistic regression on the full dataset, which included indicator terms for oncology indications and genes labeled manually versus predicted by the statistical model as LoF benign (Table 4.1). Ordered logistic regressions are a sub-class of logistic regression that account for the ordinal nature of the outcome data by calculating the probability that a value will fall between a defined threshold. No intercept term was added in this

instance. To model the progression between individual clinical trial phases we fitted twelve separate logistic regression models (Table 4.2), again separating oncology from non-oncology indications and manually annotated benign LoF genes from the supervised model predictions, and including benign LoF status, genetic support and LOEUF as covariates.

### 4.2.2 Gene expression analysis of Benign and pBenign genes

Gene expression data for the olfactory receptors OR2T10 and ORT211 were assessed using GTEx (GRCh37) and associated data contained within the ArrayStudio human DiseaseLand dataset. Data were also queried for  gene expression in atopic dermatitis and psoriasis specifically using data from GSE121212. This dataset was chosen specifically as it was generated by the collaborating group at AbbVie.

### 4.2.3 Gene set enrichment analysis

We conducted gene set enrichment analysis to further understand the properties of the gene lists. This was conducted using the Enrichr web interface [24]. Gene lists were uploaded and pathway, ontology and disease data were queried. This was done across the pBenign gene list.

# 4.3 Results

## 4.3.1 Estimated effect of pLoF benign status on drug target success

With the ever increasing cost of drug design and development, exacerbated by the high attrition rate in early and late-stage development [25], the need to select good targets is clear. Nelson et al. 2015 and King et al. 2019 illustrated a beneficial effect of GWAS and OMIM genetic evidence on drug target approval probability [2,3]. We replicate this finding in our dataset (Tables 4.1 and 4.2): presence of genetic support is significantly associated with higher drug target success probability in both oncology and non-oncology indications. Minikel et al. identified a small increase in LoF constraint (as reflected in lower observed/expected LoF ratios) among targets of approved drugs versus all protein-coding genes [7]. We replicate this finding in our dataset: lower LOEUF scores are associated with increased probability of drug target success, even after controlling for genetic evidence.

We found that our benign pLoF annotation is also significantly associated with drug target success in the ordered logistic regression models even after controlling for genetic support and LoF constraint based on the LOEUF metric (Table 4.1). Under these models, genes with benign LoF variants have an increased probability of clinical success. In the analysis of individual phase transitions (Table 4.2) we also find that genes with benign pLoF variants (based on manual labeling) have a higher probability of transitioning from Phase II to Phase III.

| Regression feature | Regression model | | | | |
|---|---|---|---|---|---|
| | Benign LoF labels: manual | | Benign LoF labels: predicted | | Combined model: 7681 T-I pairs 1103 unique genes |
| | Non-oncology indications: 3302 T-I pairs 697 unique genes | Oncology indications: 2725 T-I pairs 437 unique genes | Non-oncology indications: 1142 T-I pairs 177 unique genes | Oncology indications: 512 T-I pairs 99 unique genes | |
| Benign LoF status | beta = 0.41<br>p = 3.41e-04 | beta = 0.01<br>p = 9.26e-01 | beta = -0.21<br>p = 4.16e-01 | beta = -0.03<br>p = 9.56e-01 | beta = 0.22<br>p = 1.12e-02 |
| Genetic support | beta = 0.29<br>p = 2.35e-02 | beta = 0.51<br>p = 1.66e-02 | beta = 0.67<br>p = 2.81e-05 | beta = 0.17<br>p = 6.04e-01 | beta = 0.44<br>p = 4.64e-07 |
| LOEUF value | beta = -0.27<br>p = 1.58e-04 | beta = -0.48<br>p = 8.31e-08 | beta = -0.43<br>p = 1.02e-03 | beta = 0.06<br>p = 7.70e-01 | beta = -0.34<br>p = 5.08e-12 |
| Manual or predicted LoF labels | NA | | | | beta = -0.22<br>p = 2.33e-05 |
| Oncology or non-oncology indication | NA | | | | beta = -0.94<br>p = 3.47e-97 |

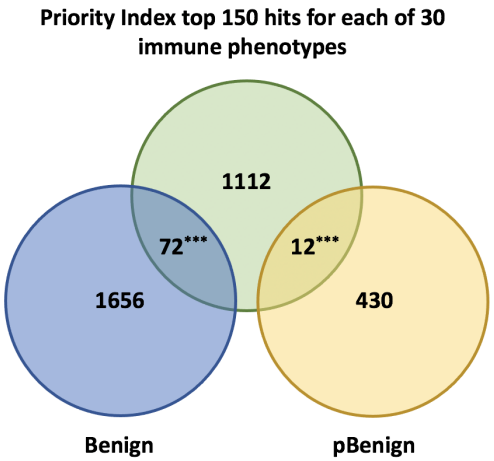
**Table 4.1 - Estimated effect of benign LoF status on the probability of advancing in clinical development.** Indications are partitioned into oncology and non-oncology indications. Genes are partitioned based on the source of their benign LoF annotation: manual labeling versus statistical prediction. An ordered logistic regression model is fitted to estimate the effect of the benign LoF annotation on clinical trial success, while controlling for genetic support and intolerance to LoF variation (as measured by the LOEUF metric). A combined ordered logistic regression model is also fitted to improve statistical power by analyzing the full dataset jointly. Indicator variables are used in this combined model to label the oncology versus non-oncology indications, and the manual versus statistically predicted benign LoF annotations. Regression terms that pass the 0.05 significance level are highlighted in each model. Statistical significance can be affected by sample size, so we also record the number of unique target genes and unique target-indication pairs for each regression. The sign of the beta coefficient for each feature records whether the feature has a positive or negative impact on drug target success probability. When statistically significant, benign LoF status increases the probability of target success, even when controlling for genetic support and LOEUF scores.

| Phase transition | Regression feature | Regression model | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Benign LoF labels: manual | | | | Benign LoF labels: predicted | | | |
| | | Non-oncology indications | | Oncology indications | | Non-oncology indications | | Oncology indications | |
| | | Regression results | Unique genes | Regression results | Unique genes | Regression results | Unique genes | Regression results | Unique genes |
| Phase I – Phase II | Benign LoF status | beta = 0.30 p = 0.06 | 697 | beta = 0.12 p = 0.47 | 437 | beta = -0.36 p = 0.28 | 177 | beta = 0.57 p = 0.49 | 99 |
| | Genetic support | beta = 0.14 p = 0.40 | | beta = 0.68 p = 0.01 | | beta = 0.34 p = 0.16 | | beta = 0.41 p = 0.33 | |
| | LOEUF value | beta = -0.17 p = 0.07 | | beta = -0.41 p = 2.09e-05 | | beta = -0.35 p = 0.06 | | beta = -0.04 p = 0.86 | |
| Phase II – Phase III | Benign LoF status | beta = 0.46 p = 0.001 | 595 | beta = -0.39 p = 0.12 | 331 | beta = 0.02 p = 0.94 | 148 | beta = -1.20 p = 0.28 | 77 |
| | Genetic support | beta = 0.33 p = 0.03 | | beta = -0.06 p = 0.82 | | beta = 0.68 p = 0.0004 | | beta = -0.40 p = 0.41 | |
| | LOEUF value | beta = -0.30 p = 0.0007 | | beta = -0.49 p = 0.0006 | | beta = -0.39 p = 0.01 | | beta = 0.18 p = 0.56 | |
| Phase III – Approved | Benign LoF status | beta = -0.01 p = 0.92 | 318 | beta = 0.58 p = 0.23 | 149 | beta = 0.004 p = 0.99 | 100 | beta = -16.75 p = 0.99 | 32 |
| | Genetic support | beta = 0.25 p = 0.28 | | beta = 2.71 p = 0.008 | | beta = 0.67 p = 0.02 | | beta = 0.47 p = 0.60 | |
| | LOEUF value | beta = -0.19 p = 0.18 | | beta = -0.46 p = 0.059 | | beta = -0.30 p = 0.21 | | beta = 1.26 p = 0.03 | |

**Table 4.2 - Effect of benign LoF status on the probability of advancing from individual clinical trial phases.** Indications are partitioned into oncology and non-oncology indications. Genes are partitioned based on the source of their benign LoF annotation: manual labeling versus statistical prediction. Twelve logistic regression models are fitted to estimate the effect of the benign LoF annotation on transition from individual clinical trial stages, while controlling for genetic support and intolerance to LoF variation (as measured by the LOEUF metric). Regression terms that pass the 0.05 significance level are highlighted in each model. Statistical significance can be affected by sample size, so we also record the number of unique target genes for each regression. The sign of the beta coefficient for each feature records whether the feature has a positive or negative impact on clinical trial transition probability. When statistically significant, benign LoF status increases the probability of target success, even when controlling for genetic support and LOEUF scores.

We then queried our Benign and pBenign genes for overlaps with current successful drug targets, as well as genes predicted to be promising future drug targets. We leveraged the genetics-based target prioritization framework developed in Fang et al. (2019, Priority Index) [26]. The Priority Index approach has curated historic drug target success data for 30 immune traits, generating prioritized novel target gene lists based on integrated analysis of multiple genetic lines of evidence. We identified highly statistically significant overlaps between the top Priority Index hits across immune indications and Benign and pBenign genes (p = 8.26e-07 and p = 9.55e-05 respectively based on hypergeometric tests, Fig. 4.1A). In addition, we have identified multiple Benign and pBenign genes that are current successful drug targets in the immunology space (Figure 4.1B).

**A**




**B**

| Indication | Drug | Gene | LoF status |
|---|---|---|---|
| Asthma | AMINOPHYLLINE, THEOPHYLLINE | ADORA3 | benign |
| Asthma | ISOPROTERENOL HYDROCHLORIDE | ADRB3 | pbenign |
| Ankylosing Spondylitis | ESOMEPRAZOLE MAGNESIUM | ATP4A | benign |
| Osteoarthritis | ESOMEPRAZOLE MAGNESIUM | ATP4A | benign |
| Rheumatoid Arthritis | ESOMEPRAZOLE MAGNESIUM | ATP4A | benign |
| Allergy | DUPILUMAB | IL4R | benign |
| Crohn's Disease | NATALIZUMAB, VEDOLIZUMAB | ITGA4 | benign |
| Multiple Sclerosis | NATALIZUMAB | ITGA4 | benign |
| Ulcerative Colitis | VEDOLIZUMAB | ITGA4 | benign |
| Ankylosing Spondylitis | NAPROXEN, NAPROXEN SODIUM, DICLOFENAC | PTGS1 | benign |
| Gout | NAPROXEN SODIUM | PTGS1 | benign |
| Osteoarthritis | DICLOFENAC, DICLOFENAC SODIUM, IBUPROFEN, NAPROXEN, NAPROXEN SODIUM, PIROXICAM | PTGS1 | benign |
| Rheumatoid Arthritis | DICLOFENAC, DICLOFENAC SODIUM, NAPROXEN, NAPROXEN SODIUM, PIROXICAM, MESALAMINE | PTGS1 | benign |
| Primary Biliary Cholangitis | TRIAMTERENE | SCNN1A | benign |
| Gout | SULFINPYRAZONE | SLC22A12 | benign |
| Gout | LESINURAD | SLC22A12 | benign |
| Gout | ALLOPURINOL SODIUM, FEBUXOSTAT, ALLOPURINOL | XDH | benign |

**Figure 4.1 - Overlap between LoF benign genes and genes highlighted by the Priority Index pipeline as higher probability of clinical success (A) or already approved drug targets (B) across 30 immune indications.** Overlaps with already approved targets highlight the potential of LoF benign genes to serve as successful drug targets. Overlaps with top gene hits from the Priority Index model highlight potentially new target opportunities in the immunology space that lie at the interface between the Priority Index genetic and epigenetic evidence and the benign LoF annotation. Asterisks denote statistical significance based on a hypergeometric test, p-value < 0.0001.

## 4.3.2 Gene set enrichment

Gene set enrichment analysis of the 442 pBenign genes reveals significant enrichment of ORs (see table 4.3) in both KEGG and BioPlanet pathways. Similarly, the only significantly enriched ontology terms are biological processes related to olfactory receptor activity (GO:0004984).

| Term | Overlap | p-value | Adjusted p-value | Odds Ratio |
|---|---|---|---|---|
| Olfactory transduction | 25/444 | 2.06E-05 | 2.59E-03 | 2.74 |
| Herpes simplex virus 1 infection | 20/492 | 6.89E-03 | 4.34E-01 | 1.92 |
| Ether lipid metabolism | 3/47 | 8.53E-02 | 1.00E+00 | 3.03 |
| alpha-Linolenic acid metabolism | 2/25 | 1.05E-01 | 1.00E+00 | 3.86 |
| Phenylalanine, tyrosine and tryptophan biosynthesis | 1/5 | 1.06E-01 | 1.00E+00 | 11.08 |

**Table 4.3 - Gene set enrichment analysis of KEGG pathways.** Table of the top five enriched KEGG pathways across 442 pBenign genes, ordered by adjusted p-value.

Similar results are shown when examining ontology data, with OR activity from GO biological pathways showing enrichment with an adjusted p-value of 0.005 (Table 4.4). Ontologies queried included the other levels of the GO (cellular compartments and molecular function), the Jensen tissues, compartments and diseases ontologies, and the Human Phenotype Ontology. These results are presented in addition to the DAVID based enrichment analysis conducted in chapter 3.3.8.

| Term | Overlap | p-value | Adjusted p-value | Odds Ratio |
|---|---|---|---|---|
| olfactory receptor activity (GO:0004984) | 20/311 | 2.17E-05 | 0.0051 | 3.14 |
| epinephrine binding (GO:0051379) | 2/8 | 1.25E-02 | 0.85 | 1.48 |
| triglyceride lipase activity (GO:0004806) | 3/25 | 1.72E-02 | 0.85 | 14.8 |
| organic anion transmembrane transporter activity (GO:0008514) | 5/70 | 1.93E-02 | 0.85 | 3.43 |
| complement receptor activity (GO:0004875) | 2/11 | 2.35E-02 | 0.85 | 9.87 |

**Table 4.4 - Gene set enrichment analysis of GO biological pathways.** The top five most enriched gene ontology biological pathway terms across the 442 pBenign genes, ordered by adjusted p-value.

No other significant associations were found using the full set of pBenign genes.

Similar analysis of the manually labelled benign targets reveals enrichment for terms related to drug metabolism, with seven of the top ten pathways relating to ligand related processes. The results shown in table 4.5 related to the BioPlanet pathway data, however similar results were also found in the WikiPathways Human data and KEGG data. No results reached adjusted p-value significance (adjusted p-value < 0.05), however this search involved a large number of genes, with 1,728 tested, and therefore we consider the nominal p-values to be worth acknowledging.

| Term | p-value |
|---|---|
| Cytochrome P450 metabolism of xenobiotics | 0.00012 |
| Generic transcription pathway | 0.00016 |
| Drug metabolism: cytochrome P450 | 0.0063 |
| Linoleic acid metabolism | 0.0023 |
| Hydrolysis of lysophosphatidylcholine (LPC) | 0.004 |
| Retinol metabolism | 0.0033 |
| Tamoxifen metabolism | 0.0042 |
| Drug metabolism: other enzymes | 0.0043 |
| Taste transduction | 0.0043 |
| ABC transporters | 0.006 |

**Table 4.5 - Gene set enrichment analysis of BioPlanet pathways.** 1728 benign genes were assessed, all p-values are nominal, adjusted p-values did not reach significance.

# 4.4 Discussion

Retrospective analyses of drug-development efforts highlight that genetic information is predictive of development success [2,3]. In particular, OMIM and GWAS data supporting target-indication links are positively associated with successful clinical development. We sought to expand on this work by examining the value of homozygous LoF variation in predicting drug-development success. Analysis of pLoF within drug targets reveals elevated levels of constraint, despite drug-targets found to harbour rare-homozygous pLoF being twice as likely to make successful targets [1,7]. Our work clarifies the relationship between LoF variation and drug-development success, showing it provides the strongest predictive value of features tested in non-oncology indications.

## 4.4.1 LoF labelling to predict trial probability of success

First we catalogued data relating to rhLoF, from large-scale genomic studies in diverse populations, classifying variants as either benign, deleterious or not determined. The rarity of homozygous LoF variation means finding all such variants will require orders of magnitude more sequencing, and in particular, the targeting of bottlenecked and consanguineous populations [7]. Thus we built an ensemble random forest approach to predict 492 benign genes, in addition to the cataloguing of 1744 benign rhLoF genes within sequenced cohorts. Of these, 818 are likely druggable (of which 120 are pBenign).

Our findings recapitulate that genetic information adds to the probability of drug-development success. However we uncover greater subtlety in this association by applying a more granular approach. Reasoning that more severe safety profiles are tolerated for oncological indications, we split target-indication pairs into non-oncology and oncology groups. Our findings indicate that our benign LoF annotation is more informative than both LOEUF and genetic support annotations for non-oncology target-indication pairs (beta = 0.41 vs -0.27 and 0.29 respectively, Table 1) . Additionally, through a phase-specific ordered logistic regression, we observe that none of the information classes predict success in phase I-II or phase III-approval in non-oncology indications. Due to a lack of data on the reason for drug failures, we can only broadly infer that efficacy and safety may be positively predicted using these data.

Divergence occurs when examining oncology classes, where benign pLoF adds no value for oncology-target pairs either in the combined model, or in any constituent phase-transition. Conversely, LOEUF scores and genetic support labels are informative in oncology indications. This difference in effect, also observed in Naramsimhan et al. (2016) and Minikel et al. (2020), appears contradictory. It is important to note that whilst benign status and LOEUF value are both ascertained through the identification of pLoF variation, the former measures tolerance to homozygous LoF inactivation, whereas the latter is primarily a measure of intolerance to heterozygous LoF. Further study may reveal that effects are in part due to incomplete penetrance, or unidentified rescue effects, however this remains outside the scope of this work.
Despite relatively high performance in predicting pBenign pLoF genes, we observed no effect on drug target probability of success. However, we highlight that the number of genes within these sets are small, and as these genes are by definition understudied, we likely suffer from a lack of power.

## 4.4.2 Olfactory receptors and the dark genome

Within both our benign and predicted gene sets, we see a heavy enrichment for understudied genes ('Tdark' genes, Fig.3B, Tables 3.3 and 3.4). Of note within this group are the OR genes. Historically understudied, more recent work has shown that over half of ORs are expressed beyond olfactory tissues. With only 10% of ORs being functional characterised, they have already been shown to be a potentially untapped pool of therapeutic targets or biomarkers in cancer [27–38], heart disease[39], blood pressure and kidney function[40] , hair loss [41] and more general metabolic processes (see [42] and [43] for detailed reviews). We add to this growing list OR2T10 and OR2T11. Despite low overall levels of expression across tissues, studies have previously linked these ORs with autism spectrum disorder [44–46]. OR2T10 and OR2T11 are both significantly downregulated in psoriasis and upregulated in the synovium of rheumatoid arthritis patients (log2FC 2.7 and 1.9 respectively, with adjusted p-value < 0.05, Fig. 4.2A). OR2T11 is additionally upregulated in B cells and neutrophils of Systemic lupus erythematosus patients, and in cell models of kidney disease (Fig. 4.2B).

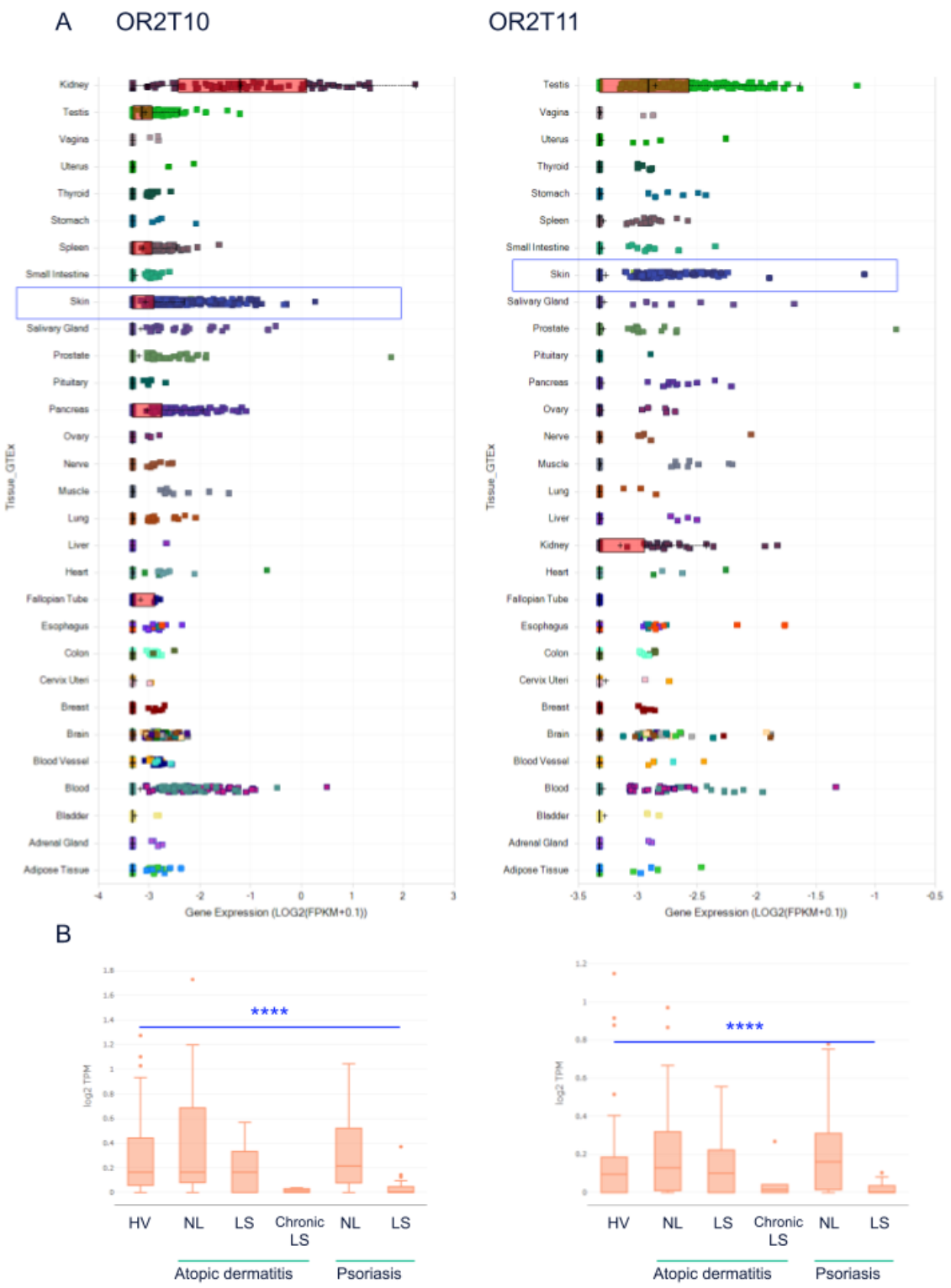



**Figure 4.2 - Gene expression of OR2T10 and OR2T11 in healthy and diseased tissue.**

A) Gene expression data for OR2T10 and OR2T11 across all tissues in the Tissue_GTEx B_37 dataset in ArrayStudio. Expression in skin is highlighted by blue boxes. B) Boxplots showing the significant levels of downregulated expression of OR2T10 and OR2T11

when comparing healthy volunteers (HV), and atopic dermatitis and psoriasis patients, from non-lesional (NL) to legional skin (LS).

We saw limited benefit in analysing the pBenign targets in this context, with no significant betas arising from the models fit. However it must be noted that the numbers of pBenign genes are much smaller, with only 177 unique genes at phase 1 versus 697 of the manually labelled genes in non-oncology indications. Additionally, the average number of indications for each gene was greater in the pBenign group, with 6.5 indications per gene vs 4/7 indications per gene in the manually labelled genes. Therefore, poor target selection, in which the target would not work regardless of the indication may affect the pBenign genes more. As covered in chapter 3, the genes originating in the ND set are less well studied, and less likely to be targets of approved drugs. This means that this sort of analysis, in which we review historic drug targeting may similarly suffer from this lack of knowledge. Targets that are less known may not be tested in a variety of other indications, as areas of the dark genome remain unexplored [15]. Similarly, as discussed in chapter 3, this lack of information may just mean that our predicted labels are not reliable, and that as sequencing efforts proceed, we will see that different features are important for prediction of LoF status. This knowledge may lead us to more clearly understand specifically what data relating to LoF is predictive of drug probability of success.

Gene set enrichment analysis is a widely used approach to associate biological and functional information with a list of genes. In this case, examination of the benign genes within our dataset shows the relative enrichment of drug-metabolism based pathways (see table 4.5), although this did not reach significance after multiple-testing correction. Given the evidence that suggests that these benign genes may be better targets, this association should be further investigated, as it may offer a functional explanation for the positive association between benign LoF status and drug development success.

To our knowledge, we are the first to examine drug development probability of success genetic associations in this detail. We demonstrate the heterogeneity of predictive power associated with differing forms of genetic information at different stages of drug development and highlight that pLoF can inform both non-oncological and oncological drug discovery efforts. This work also highlights the reasons to further explore the dark genome, as many viable, safe and effective targets may be contained in these oft-ignored genes.

# 4.5 Conclusion

In this chapter we demonstrate the utility of our LoF phenotype classification. The classification of genes based on observed homozygous LoF is predictive of probability of success in historic drug data. We also show that different types of genetic information contribute value at different stages of drug development, likely indicating that our LoF labels are indicative of tolerable safety in clinical trials. This supports the notion that gene LoF is a model for lifelong inhibition, and that it could impart information that would greatly aid in target selection.  Exploring the targets that we hypothesise to be safer targets, we see an enrichment for GPCRs and oGPCRs, and particularly within this group we find the understudied olfactory receptors. As members of the druggable genome, there is value in studying these genes further, especially as their involvement in disease is more fully appreciated.

Our work contributes to the growing body of evidence on the value of human genetics in all stages of drug development, in particular highlighting the value of cataloguing homozygous LoF variation.

# 4.6 References


1. Narasimhan, V. M. *et al.* Health and population effects of rare gene knockouts in adult humans with related parents. *Science* **352**, 474–477 (2016).
2. Nelson, M. R. *et al.* The support of human genetic evidence for approved drug indications. *Nat. Genet.* **47**, 856–860 (2015).
3. King, E. A., Davis, J. W. & Degner, J. F. Are drug targets with genetic support twice as likely to be approved? Revised estimates of the impact of genetic support for drug mechanisms on the probability of drug approval. *PLoS Genet.* **15**, e1008489 (2019).
4. MacArthur, D. G. & Tyler-Smith, C. Loss-of-function variants in the genomes of healthy humans. *Hum. Mol. Genet.* **19**, R125–30 (2010).
5. MacArthur, D. G. *et al.* A systematic survey of loss-of-function variants in human protein-coding genes. *Science* **335**, 823–828 (2012).
6. Karczewski, K. J., Francioli, L. C., Tiao, G. & Cummings, B. B. The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature* 434–443 (2020).
7. Minikel, E. V. *et al.* Evaluating potential drug targets through human loss-of-function genetic variation. *Nature* 459–464 (2020).
8. Hauser, A. S., Attwood, M. M., Rask-Andersen, M., Schiöth, H. B. & Gloriam, D. E. Trends in GPCR drug discovery: new agents, targets and indications. *Nat. Rev. Drug Discov.* **16**, 829–842 (2017).
9. Weis, W. I. & Kobilka, B. K. The Molecular Basis of G Protein–Coupled Receptor Activation. *Annu. Rev. Biochem.* **87**, 897–919 (2018).
10. Lagerström, M. C. & Schiöth, H. B. Structural diversity of G protein-coupled receptors and significance for drug discovery. *Nat. Rev. Drug Discov.* **7**, 339–357 (2008).

11. Gilad, Y. & Lancet, D. Population differences in the human functional olfactory repertoire. *Mol. Biol. Evol.* **20**, 307–314 (2003).
12. Buck, L. & Axel, R. A novel multigene family may encode odorant receptors: a molecular basis for odor recognition. *Cell* **65**, 175–187 (1991).
13. Lancet, D. & Ben-Arie, N. Olfactory receptors. *Current Biology* vol. 3 804 (1993).
14. Insel, P. A. *et al.* G Protein-Coupled Receptor (GPCR) Expression in Native Cells: 'Novel' endoGPCRs as Physiologic Regulators and Therapeutic Targets. *Mol. Pharmacol.* **88**, 181–187 (2015).
15. Oprea, T. I. *et al.* Unexplored therapeutic opportunities in the human genome. *Nat. Rev. Drug Discov.* **17**, 377 (2018).
16. Edwards, A. M. *et al.* Too many roads not taken. *Nature* **470**, 163–165 (2011).
17. Campillos, M., Kuhn, M., Gavin, A.-C., Jensen, L. J. & Bork, P. Drug target identification using side-effect similarity. *Science* **321**, 263–266 (2008).
18. Bock, A., Kostenis, E., Tränkle, C., Lohse, M. J. & Mohr, K. Pilot the pulse: controlling the multiplicity of receptor dynamics. *Trends Pharmacol. Sci.* **35**, 630–638 (2014).
19. Spehr, M. & Munger, S. D. Olfactory receptors: G protein-coupled receptors and beyond. *J. Neurochem.* **109**, 1570–1583 (2009).
20. Kim, S. Y. *et al.* Phosphoinositide and Erk signaling pathways mediate activity-driven rodent olfactory sensory neuronal survival and stress mitigation. *J. Neurochem.* **134**, 486–498 (2015).
21. Casey, L. M. *et al.* Small molecule disruption of Gβγ signaling inhibits the progression of heart failure. *Circ. Res.* **107**, 532–539 (2010).
22. Kuhn, M. *et al.* Systematic identification of proteins that elicit drug side effects. *Mol. Syst. Biol.* **9**, 663 (2013).
23. Davis, M. I. *et al.* Comprehensive analysis of kinase inhibitor selectivity. *Nat. Biotechnol.* **29**, 1046–1051 (2011).
24. Chen, E. Y. *et al.* Enrichr: interactive and collaborative HTML5 gene list enrichment

analysis tool. *BMC Bioinformatics* **14**, 128 (2013).

25. Hwang, T. J. *et al.* Failure of Investigational Drugs in Late-Stage Clinical Development and Publication of Trial Results. *JAMA Intern. Med.* **176**, 1826–1833 (2016).
26. Fang, H. *et al.* A genetics-led approach defines the drug target landscape of 30 immune-related traits. *Nat. Genet.* **51**, 1082–1091 (2019).
27. Neuhaus, E. M. *et al.* Activation of an olfactory receptor inhibits proliferation of prostate cancer cells. *J. Biol. Chem.* **284**, 16218–16225 (2009).
28. Maßberg, D. *et al.* The activation of OR51E1 causes growth suppression of human prostate cancer cells. *Oncotarget* **7**, 48231–48249 (2016).
29. Kalbe, B. *et al.* Helional-induced activation of human olfactory receptor 2J3 promotes apoptosis and inhibits proliferation in a non-small-cell lung cancer cell line. *Eur. J. Cell Biol.* **96**, 34–46 (2017).
30. Weber, L. *et al.* Activation of odorant receptor in colorectal cancer cells leads to inhibition of cell proliferation and apoptosis. *PLoS One* **12**, e0172491 (2017).
31. Gelis, L. *et al.* Functional Characterization of the Odorant Receptor 51E2 in Human Melanocytes. *J. Biol. Chem.* **291**, 17772–17786 (2016).
32. Weber, L. *et al.* Characterization of the Olfactory Receptor OR10H1 in Human Urinary Bladder Cancer. *Front. Physiol.* **9**, 456 (2018).
33. Ranzani, M. *et al.* Revisiting olfactory receptors as putative drivers of cancer. *Wellcome Open Res* **2**, 9 (2017).
34. Weber, L. *et al.* Olfactory Receptors as Biomarkers in Human Breast Carcinoma Tissues. *Front. Oncol.* **8**, 33 (2018).
35. Weng, J. *et al.* PSGR2, a novel G-protein coupled receptor, is overexpressed in human prostate cancer. *International Journal of Cancer* vol. 118 1471–1480 (2006).
36. Cao, W., Li, F., Yao, J. & Yu, J. Prostate specific G protein coupled receptor is associated with prostate cancer prognosis and affects cancer cell proliferation and

invasion. *BMC Cancer* **15**, 915 (2015).

37. Rigau, M. *et al.* PSGR and PCA3 as biomarkers for the detection of prostate cancer in urine. *Prostate* **70**, 1760–1767 (2010).
38. Sanz, G. *et al.* Promotion of cancer cell invasiveness and metastasis emergence caused by olfactory receptor stimulation. *PLoS One* **9**, e85110 (2014).
39. Jovancevic, N. *et al.* Medium-chain fatty acids modulate myocardial function via a cardiac odorant receptor. *Basic Res. Cardiol.* **112**, 13 (2017).
40. Shepard, B. D. & Pluznick, J. L. How does your kidney smell? Emerging roles for olfactory receptors in renal function. *Pediatr. Nephrol.* **31**, 715–723 (2016).
41. Chéret, J. *et al.* Olfactory receptor OR2AT4 regulates human hair growth. *Nat. Commun.* **9**, 1–12 (2018).
42. Lee, S.-J., Depoortere, I. & Hatt, H. Therapeutic potential of ectopic olfactory and taste receptors. *Nat. Rev. Drug Discov.* **18**, 116–138 (2019).
43. Chen, Z., Zhao, H., Fu, N. & Chen, L. The diversified function and potential therapy of ectopic olfactory receptors in non-olfactory tissues. *J. Cell. Physiol.* **233**, 2104–2115 (2018).
44. Codina-Solà, M. *et al.* Integrated analysis of whole-exome sequencing and transcriptome profiling in males with autism spectrum disorders. *Mol. Autism* **6**, (2015).
45. Gonzalez-Mantilla, A. J., Moreno-De-Luca, A., Ledbetter, D. H. & Martin, C. L. A Cross-Disorder Method to Identify Novel Candidate Genes for Developmental Brain Disorders. *JAMA Psychiatry* **73**, (2016).
46. Iossifov, I. *et al.* The contribution of de novo coding mutations to autism spectrum disorder. *Nature* **515**, (2014).

# Chapter 5: Creating a rare disease database for the 100,000 Genome Project

## Chapter commentary

The data included in the 100,000 genome project (100KGP) are broad and diffuse, with many different sources of information from many different fields being collected. Representing such an array of data in a consistent manner is a challenge across many areas of biology. This is further compounded by the need for said data to be machine readable, thus allowing application of a full computational toolkit to fully explore the data. In this chapter I will describe efforts to structure and standardise data into a simple GEL disease database. Following this, I will outline the ways in which I have facilitated visualising the database in a number of different ways with useful helper functions and output pages to provide salient information about a disease and its phenotypes and genes. Much of this work reflects the original direction of this thesis, in which downstream repurposing would occur from this database. However due to the opportunity to complete a placement at the Broad Institute, the focus of this thesis pivoted to an in depth analysis of how human genetics can be used to inform drug discovery. As I outlined in the introduction, human genetics has become key to the increasing success of drug development in recent years, and plays an important role in deciding which targets we should prioritise for development. As such, the data and insights generated from the preceding chapters are directly applicable to the database described herein.

# 5.1 Introduction

## 5.1.1 The use of ontologies as a tool for disease understanding

The explosion of data generated within the biological sciences since the turn of the century has led to the need to develop new data handling and storage approaches. The emergence of ‘Omics’ has generated a wide array of data, ranging from genome and transcriptome sequencing data to metabolome mass-spectrometry, imaging and electronic health record data. The proper representation of such heterogeneous data in a form that allows for knowledge to be gleaned from not just one area, but between different fields, is an important and challenging task. A successful approach to such a problem is the generation of ontologies.

Definitions for ontologies are relatively broad (in itself a point of irony), which is largely reflective of their universal application to any field. In the broadest sense, they are a structured vocabulary of a knowledge-base - a formal way of representing knowledge in which concepts are described both by their meaning and their relationship to each other. Generally shared key features of ontologies include a domain-specific vocabulary, objects (individual objects, for example a gene) classes and relations (sets of objects and the way they can be related to each other) and attributes (a feature of the objects and classes). The source of data used to derive this knowledge-base can be diverse, representing meta-data, experimental data and data from disparate organisms, among others. Reflecting this, the data forms themselves can be recalcitrant to harmonisation, with some taking forms such as strings, integers, and floating-points. However, what can be more easily derived is the relationship between them, such as the statement that DBN1 ‘is a’ gene, the product of which ‘colocalises with’ actin. The words in inverted commas are standardised relationship terms and are the links between different sources of information.

The main aim in the construction of such data forms is the ability to link diverse datasets in a machine-readable way. The Semantic Web, an extension of the World Wide Web in which data structures are tagged to allow for machine integration, represents a key development in the field of knowledge integration.  The more specific use in the field of Biological Science is referred to as Semantic Biology [1]. Here

knowledge is reduced to the form of a semantic “triple” of a subject - predicate - object (DBN1-colocalises with-actin). The classical example of a biological ontology is the Gene Ontology (GO) [2]. First developed at the turn of the millenium, this ontology structures data as directed acyclic graphs (graphs in which the direction of relationship between nodes is defined and unidirectional) relating to gene and protein roles across many organisms (Fig. 5.1). In the intervening years, many more ontologies have emerged covering many different areas of interest, including the Human Phenotype Ontology [3] (HPO) and the Online Mendelian Inheritance in Man [4] (OMIM) ontology.

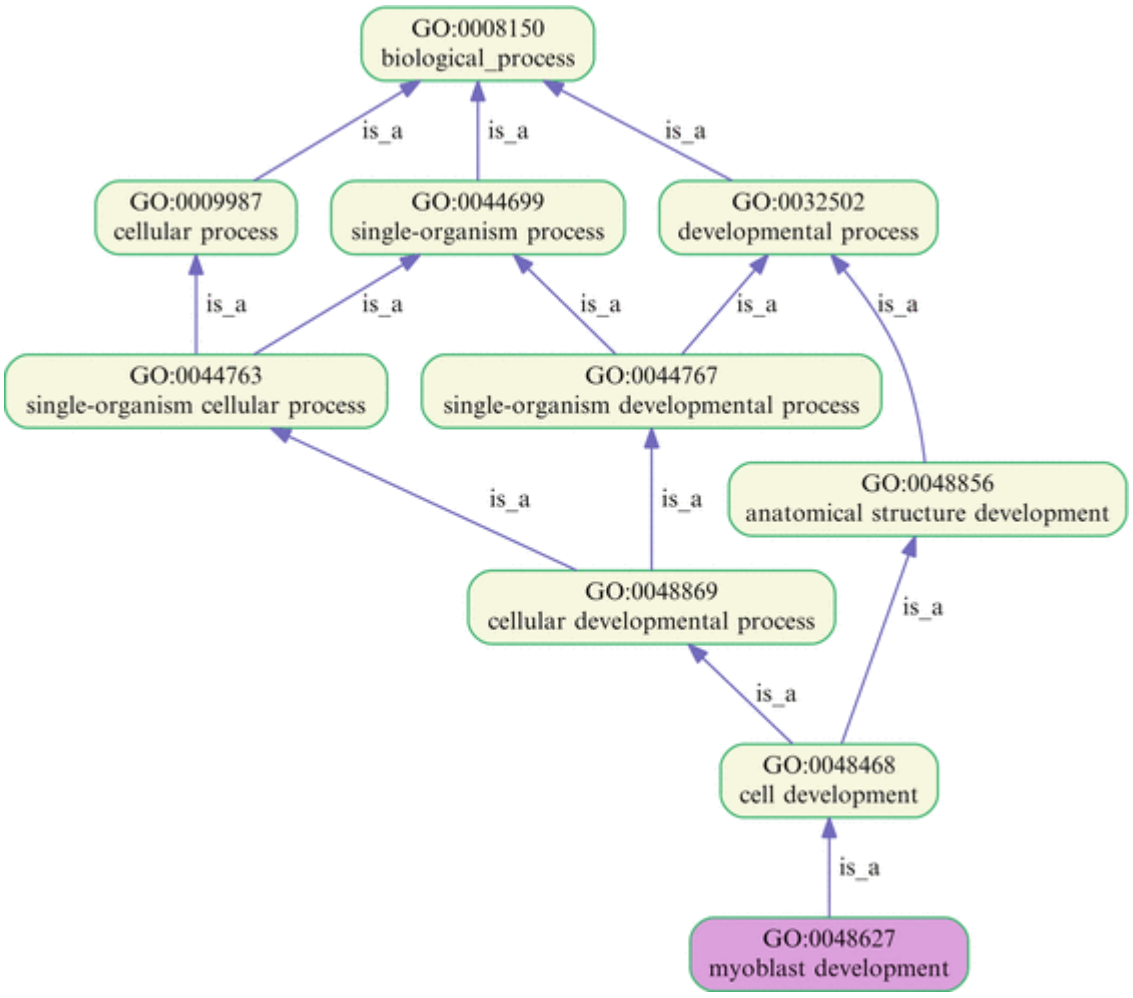



**Figure 5.1 - An example of a DAG from the Gene Ontology**. Here the term ‘myoblast development’ is described by its relationships (denoted by arrows between entities) to other entities [5]. Relationships are directed, and lead from the most specific term to the least specific.

## 5.1.2 Databases as repositories of biological knowledge

### 5.1.2.1 Database design

Databases are used to store data in a structured manner in order to increase both query and computation efficiency, while reducing duplication of data, thus minimising storage requirements. Databases are ubiquitous in the modern world, with all forms of data being stored in this manner. However in reality there are a large number of different databases that can be chosen, with structures varying depending on how the database may be used or what form the original data take. For this project, we used the most commonly used database type [6], the relational database management system (RDBMS). In an RDBMS, data are stored in tables as rows and columns, with each table representing an entity, columns representing an attribute of the entity, and rows are data of an entity [7]. Querying of RDMSs is achieved using a Structured Query Language (SQL), where commands and statements are used to build, query, edit or delete data stored within the database. An open source variant, known as MySQL, was our language of choice for this project.

In order to build our database, we needed to first normalise our data (see Fig. 5.2). In the context of database construction, normalisation is the process of imposing order on the data, such that data integrity is maintained, and that storage and querying efficiency is maximised. In brief, the **first normal form (1NF)** ensures that repeated groups are reduced to atomic (indivisible) representations, that each set of data has its own table, and that each of those sets can be identified with a primary key. So for example, a list of genes and their proteins would be separated into table 1 - Genes, and table 2 - Proteins. Only one instance of each gene or protein would be included within their respective tables. Each gene or protein would be assigned a key, for example G1 and P1 for the first gene and protein in the table respectively. Now to identify a row from our original table of genes and proteins, we would combine a gene and protein ID, which due to their being atomic, would be unique. The next step, or the **second normal form (2NF)**, creates a single column primary key. So for example, if you wish to include functional properties of a gene and protein, the only way to avoid repeated gene/protein lines and keep data atomic (i.e. not have multiple functions in a single row), would be to separate the functional data into a separate table. Following this is **third normal form (3NF)**, which breaks transitive dependencies. Values that may change need to be separated so that the value only needs to be changed in one place, rather than in each record where it is used. Each of these forms are dependent on the

form before, so 2NF depends on 1NF, and 3NF on 2NF. Whilst there are more normal forms than the 3 used in this database, we did not think them necessary for our particular use-case.

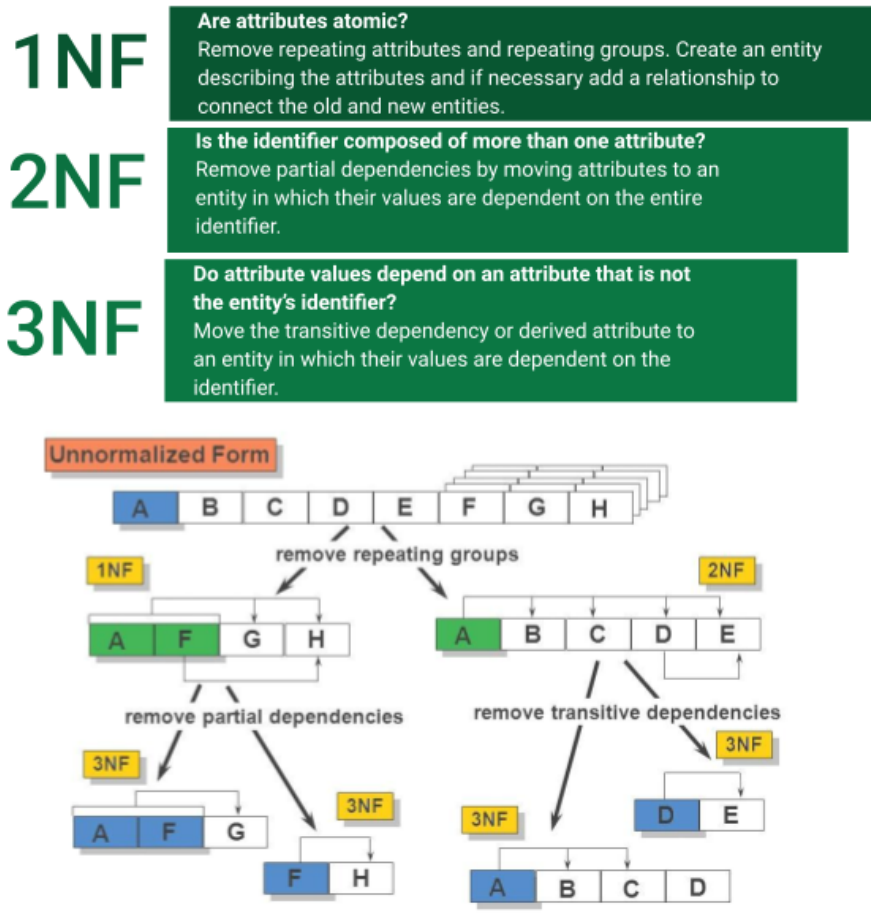



**Figure 5.2 - An overview of data normalisation.** Key considerations when normalising data are listed as questions, followed by the actions required to satisfy them. This is followed by a simple example to illustrate these principles in action. Adapted from Introduction to Modern Database Systems [8].

### 5.1.2.2 Key component databases

We use various different biological datasets to enrich the existing disease data from the GEL ontology. These are intended to add additional information, and possibly reveal other associations between the diseases. Examples of how such enrichment of data can be useful are seen in chapters 3 and 4, where we use these data as additional features for machine learning. Here we will outline the most important in terms of gene-disease associations, drug data and gene function.

#### 5.1.2.2.1 Gene Ontology (GO)

GO is a resource produced by the Gene Ontology Consortium, with the stated aim of comprehensively modelling biological systems. It is the largest database regarding the function of genes, with a current tally of ~44 thousand GO terms for over 1.5 million gene products across 4,666 species. GO terms describe biology in three ways; molecular function, cellular component or biological process. For example a cell surface transporter protein will have the molecular function of transport, alongside the cellular component of membrane, and the biological process anion transmembrane transporter activity. Relationships between terms are represented using logical definitions or equivalence axioms and are both computer and human-readable, allowing for inference through logical reasoning to be performed.

#### 5.1.2.2.2 PanelApp

This database is an integral part of our dataset, as it forms the foundation for gene-disease associations that we go on to enrich with other datasets. PanelApp is a database produced by Genomics England with the central aim of crowdsourcing diagnostic quality, standardised gene panels for diagnostic purposes [9]. Genes and other genetic units such as short tandem repeats and copy number variants that are suspected to have a causative association with a condition are submitted by researchers or sourced from groups such as the UK Genetic Testing Network. These are then reviewed by approved researchers and experts, further evaluated and reviewed by the Genomics England clinical team, and then published as a gene panel. Gene panels categorise submitted genes as either ‘green’ genes which may be used diagnostically in the clinic, ‘amber’ genes where there is a moderate level of confidence in disease association, and ‘red’ genes, indicating a gene should not be used for clinical interpretation.

There are currently 326 panels, of which the majority are related to 100KGP diseases, comprising 5870 genes and genomic entities. Examining 100KGP diseases exclusively, there are 969 green genes averaging to roughly 5.6 genes per disease [9]. The data are all accessible via a publicly accessible API and are browsable via the PanelApp website.

Whilst this resource is intended for use in diagnostic clinical settings, it also provides a valuable set of high confidence genes upon which to build an expanded disease knowledge-base.

#### 5.1.2.2.3 Drug-Gene interaction Database (DGiDB)

The DGiDB brings data together from ~30 different sources focussing on drug-gene interaction and druggability data. These data are normalised and provided with a web UI and an API for programmatic access. This resource is valuable for target related data, such as identifying targets that are members of the druggable genome and targets of approved drugs.

# 5.2 Methods

We sought to automate the entirety of the database building and updating process. This required the use of R for data wrangling, Python for specific API calling, and Bash wrapper scripts to reduce this to a single command process. Here we introduce each of the scripts individually brief descriptions of intended function and a link to the Github repository for reference.

## 5.2.1 Building the database

### 5.2.1.1 Clinical phenotypes

A datasheet containing GEL disease phenotypes under study were received by personal communication from GEL. We built the GEL disease database on and around information contained within this document. We included other data with the aim of enriching and expanding upon this core of data. The data were as described previously (see Introduction). We normalised the data according to database architecture principles up to 3NF.

### 5.2.1.2 MySQL schema

We built the database using MySQL Workbench Community (v6.3, Oracle Corporation), a graphical user interface (GUI) for SQL development and database design. We dumped the final schema to file using the schema export tool for porting and backup purposes. The resulting schema can be viewed as an entity relationship diagram in Fig. 5.3, or as a SQL script in the appendix.

### 5.2.1.3 Database build scripts

#### 5.2.1.3.1 Db_update.sh

We created a bash script to build and maintain the database. We built this script with semi-interactive features to simplify maintenance of the database for novice command-line users. The script depends on there being an existing database set up in MySQL, and requires the presence of all other scripts mentioned below in the same working directory.

Upon invoking the script, the user is asked to enter the username and password for the MySQL database. The passwords are not hashed or encrypted as they are intended to be run in a secure and limited environment. The user is then prompted to enter a file path to a python environment they may already have configured. This process is optional, and is designed simply to prevent a user having to reinstall python packages required for API calling if they already have them. The user is then prompted to enter the file path to the clinical phenotypes file already described. Failure to provide a proper file path, or an incorrect MySQL DB user and password combination will result in a failure message and the script terminating. Following this, the script calls the clinical phenotypes script.

#### 5.2.1.3.2 Clinical_phenotypes_add.R

This script adds clinical phenotype data from the GEL supplied datasheet to the MySQL database instance. First the existing data from the database is pulled and compared to the new input table to check for new data. If there is no existing data, all data from the clinical phenotypes datasheet will be added. We reduce each column of data to unique values, and then create mapping tables to specify relationships between each object. We then upload these data to their relevant MySQL DB tables.

#### 5.2.1.3.3 PanelApp_prep.R

This script largely deals with pulling and cleaning data from the PanelApp website. We pull data in JSON format, and then clean it and add it to the MySQL DB. We add disease data found in PanelApp but not currently existing in the DB, and include relevant disease subgroup and group data. Following this, we filtered out low confidence genes, such that only the highest confidence associations were represented from this dataset. This script then generates gene to disease mappings and writes to the DB.

#### 5.2.1.3.4 dis_to_pheno_mapping.R

This script adds HPO data above what is already included in the GEL datasheet. This includes  adding genes associated to GEL phenotypes to the gene DB table. We

downloaded the data from the HPO website (file: ALL_SOURCES_ALL_FREQUENCIES_phenotype_to_genes.txt, date accessed 10-03-2017).
We sourced gene-disease associations from Panelapp, a crowd-sourced and expertly curated database managed by GEL. Subsequently, gene to drug associations were derived from DGIdb, an open source and open access database which brings data together from numerous sources such as DrugBank and PharmGKB. Further drug information was mined from OpenTargets and the National Cancer Institutes chemoinformatics tools.

#### 5.2.1.3.5 DGIdb_api.R

Prior to this script running, the python environment command is invoked if a file path is provided for said environment. The R script first calls the python script DBIdb_API.py, which handles API calls to the DGIdb. Drug-gene data are pulled in json format and written to a temporary '.txt' file. This is then read by the R script and cross-checked against existing data in the database. Drug interaction data are then simplified to simply 'agonist', 'inhibitor', 'unknown' or 'other' and drug names are harmonized. New drugs and their associated data are then written to the database.

#### 5.2.1.3.6 SMILE.R

This calls the National Cancer Institute chemical structure database for Simplified molecular-input line-entry system (SMILES) data. This data simplifies drug structure data into an ASCII string vector. Such data can be readily converted into 2D graphical representations of drug structure, but is more amenable to storage in string format.

#### 5.2.1.3.7 drugbank_parse.R

We first download structure data from DrugBank release 5.1.1. This script then cleans and subsets the structure data. Additional fields added from this data include structure data-file (SDF) drug representations, an alternate format to the SMILES format (SDF includes 3D coordinates and is less ambiguous than SMILES), FASTA data on biologics, and various IDs for mapping to other databases. Much of this data is

intended for interoperability, to allow for users to quickly locate more drug data in a database of their choice.

#### 5.2.1.3.8 drug_pheno_disease_mappings.R

This script simply creates mapping tables between disease-drug, and phenotype-drug tables. These links are inferred from other relationships established in previous scripts. I.e. A panelapp  gene that is targeted by a drug will  lead to an inferred link between the drug and the relevant disease for that gene. These are explicitly not approved indications, and should not be interpreted as such.

Following the completion of these processes, the shell script then engages foreign key constraints, to ensure data stability, and terminates. The script should only need to be invoked when new data are to be added, for example if there is an update in gene-disease associations, or new targets are added to a constituent database. However, due to the rapid runtime of the script, it could be feasibly updated everyday. First time use of this script to create the database will take ~1 hour to complete, with subsequent updates taking 1-5 minutes (although this is dependent on the size of the required update).

# 5.3 Results

Much of the work described herein is intended to facilitate research. Therefore there will not be in depth investigation of all disease areas in their various forms, but rather exemplars of the way in which this database can reveal disease relationships. To do this, we have concentrated on aspects of visualisation and summation of the database. We will however examine some examples of drug repurposing opportunities that are revealed via this approach later in the chapter. Many of the graphs that follow are static representations of interactive graphs, or the output of functions with simple inputs to allow for user defined customisations.

## 5.3.1 GEL disease database (GELDdb) schema

We created GELDdb with 16 tables totalling 56 columns and 295,453 rows. The entity relationship diagram shows the interlinking nature of the tables where we bring together data on phenotypes, genes and drugs. The gene categories table (gene_cats) and the corresponding mapping table (gene_to_genecat) are yet to be populated and therefore have 0 rows. Row numbers correspond to unique data entries, therefore conforming to normalisation standards. The colours of the rows in the table indicate the primary key (light green) and foreign keys (yellow/green), with all non-constrained data fields in white.

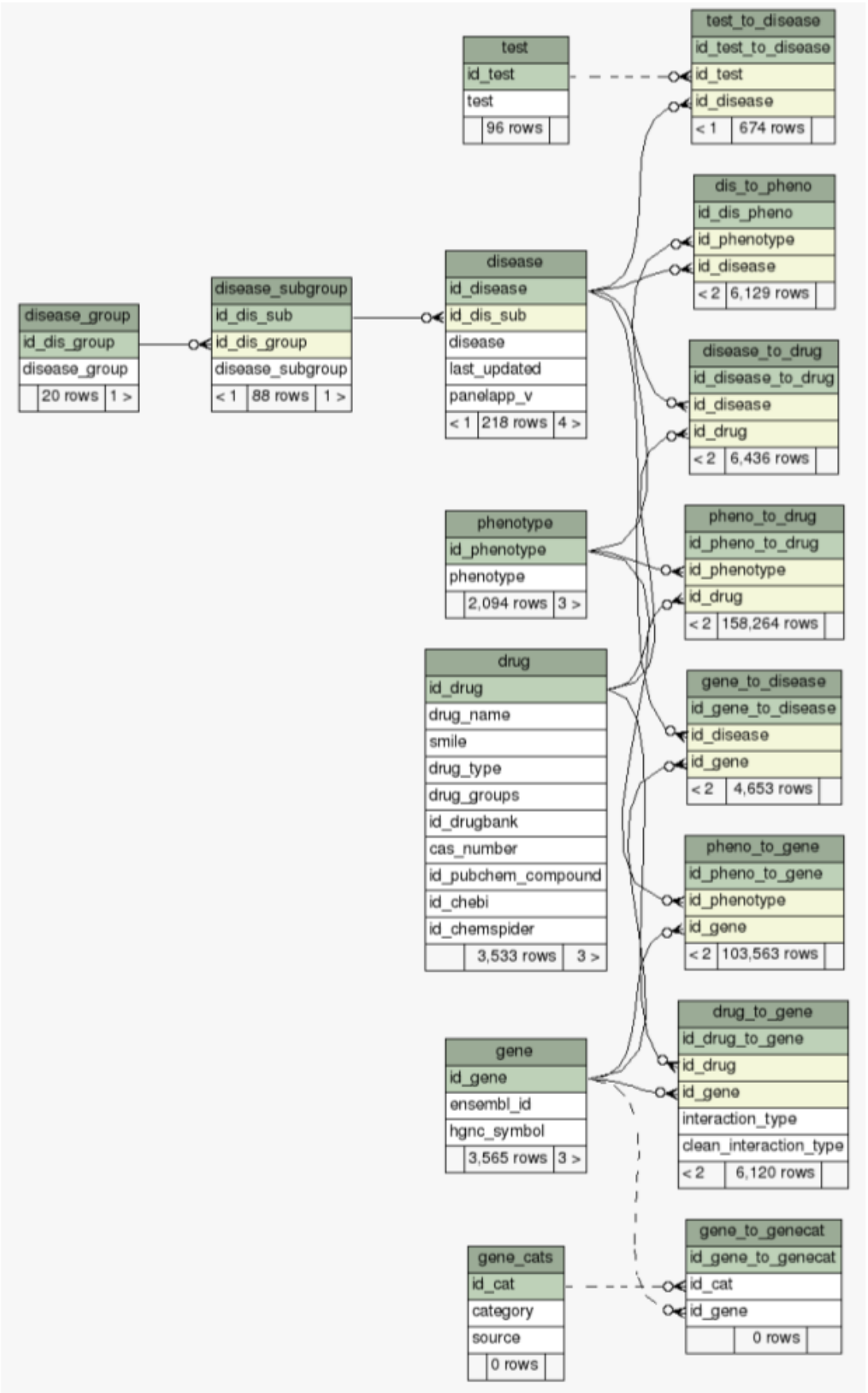

test
id_test
test
96 rows
test_to_disease
id_test_to_disease
id_test
id_disease
< 1
674 rows
dis_to_pheno
id_dis_pheno
id_phenotype
id_disease
< 2
6,129 rows
disease
id_disease
id_dis_sub
disease
last_updated
panelapp_v
< 1
218 rows
4 >
disease_subgroup
id_dis_sub
id_dis_group
disease_subgroup
< 1
88 rows
1 >
disease_group
id_dis_group
disease_group
20 rows
1 >
disease_to_drug
id_disease_to_drug
id_disease
id_drug
< 2
6,436 rows
phenotype
id_phenotype
phenotype
2,094 rows
3 >
pheno_to_drug
id_pheno_to_drug
id_phenotype
id_drug
< 2
158,264 rows
drug
id_drug
drug_name
smile
drug_type
drug_groups
id_drugbank
cas_number
id_pubchem_compound
id_chebi
id_chemspider
3,533 rows
3 >
gene_to_disease
id_gene_to_disease
id_disease
id_gene
< 2
4,653 rows
pheno_to_gene
id_pheno_to_gene
id_phenotype
id_gene
< 2
103,563 rows
drug_to_gene
id_drug_to_gene
id_drug
id_gene
interaction_type
clean_interaction_type
< 2
6,120 rows
gene
id_gene
ensembl_id
hgnc_symbol
3,565 rows
3 >
gene_to_genecat
id_gene_to_genecat
id_cat
id_gene
0 rows
gene_cats
id_cat
category
source
0 rows

**Figure 5.3 - GELDdb schema.** Entity relationships are described with table names in dark green at the top of the table, followed by primary keys in light green, and foreign keys in yellow/green.

## 5.3.2 Visualising disease spaces

The term “disease space” approximates the sum total of information contributed by all entities in our knowledge store, including genes, diseases and drugs, adhering to a given set of construction principles and boundary conditions. This space, much like that within an ontology, is limited by the observations we make, and therefore is not complete. The addition of new dimensions (for example - symptoms associated with a disease) adds to this disease space and increases the accuracy of this simulacrum of real-world phenomena.

In order to better understand the interaction of different elements of the database, we sought to visualise the conceptual disease space represented by the entities in GELDdb. We created various customisable functions to view different elements of the database in graph form. It is possible to make such graphs for any of the tables for which there are mapping tables (Fig. 5.3). N-partite graphs can be created simply by joining on the existing dataframe. We also allow for the projection of bipartite graphs such that one node is represented as a link between the other type of node. For example, a disease-gene network would be projected into a unipartite network in which diseases are joined by genes that are shared between them.

### 5.3.2.1 Disease-Gene Network

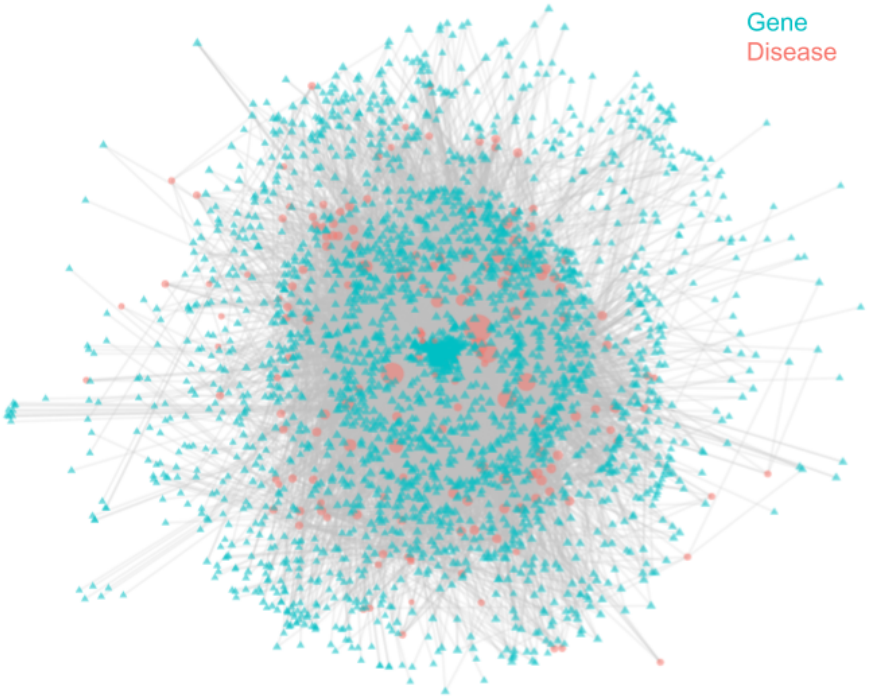



**Figure 5.4 - A network visualisation of the GEL disease space and their associated genes.** Genes (teal, triangles) and diseases (orange, circles) links are derived from GELDdb, with disease node size reflecting the degree of connectivity.

Figure 5.4 shows the mapping of diseases to their associated genes and how these interconnect. This is an example of a bipartite network, therefore genes have no direct relationship with each other, but instead are linked via the diseases in which they are implicated. Due to the size and density of this representation it is difficult to glean much human interpretable information, however the network itself is made interactive through the use of Plotly (https://plotly.com/). This means that the graph is zoomable, and hovering over a node will display a pop-up with key information about the node listed. As a set of entity relationships this data is machine readable and could potentially be used for further analysis using a range of statistical and machine learning approaches.

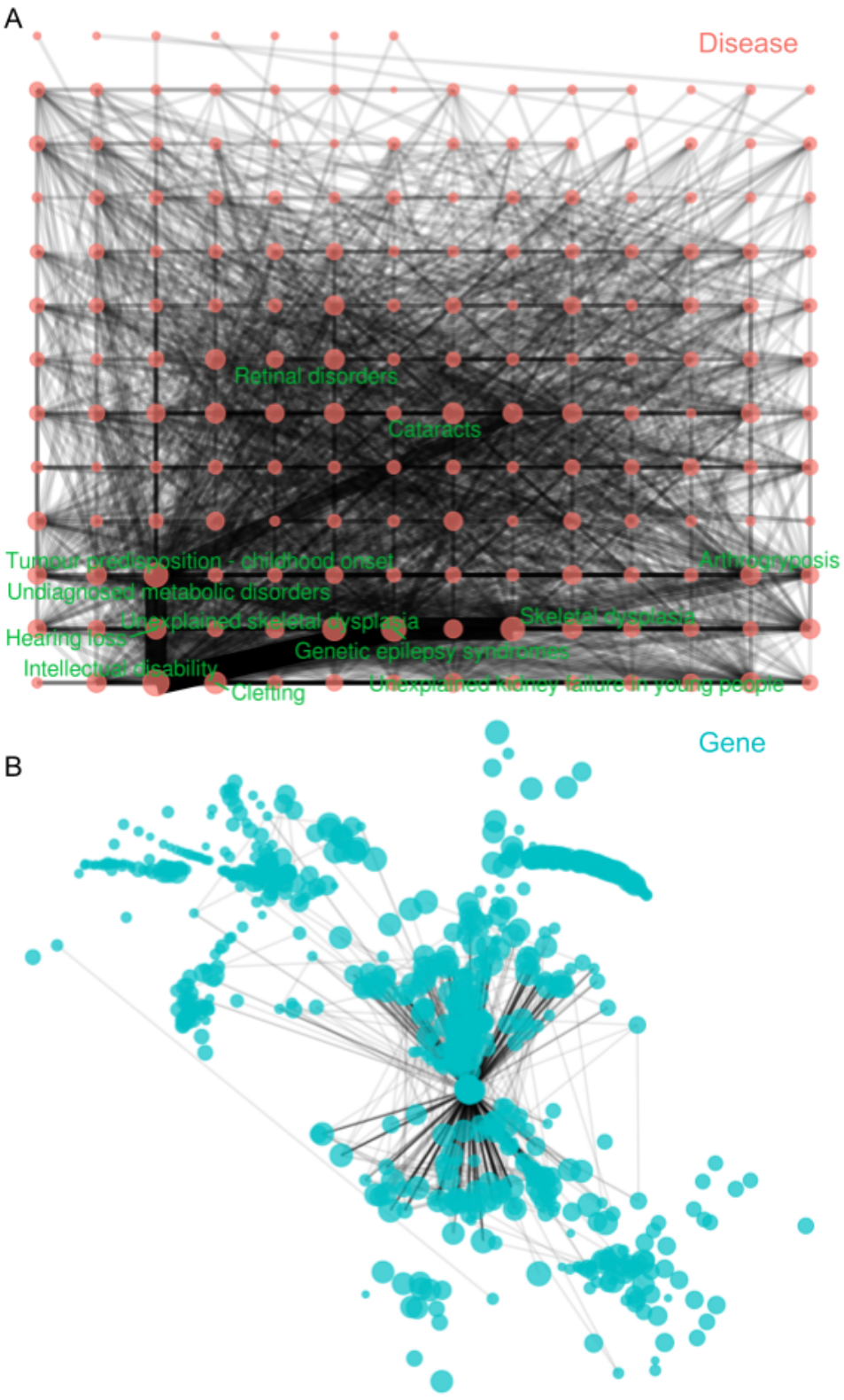



**Figure 5.5 - Unipartite projections of the GELDdb Gene-Disease bipartite network with either diseases as nodes (A) or genes as nodes (B).** A) The ten highest degree disease nodes are labelled, signifying the diseases with the most associated genes. B) Only edges for genes with more than 3 shared diseases are displayed.

Another way of simplifying the visualisation is to project the network into a unipartite network. From a bipartite network there are two possible projections, one in which the diseases are the nodes and the genes are the edges (Fig. 5.5A) and vice versa (Fig.5.5B). As there are close to 1 million edges for the latter graph, we only plotted edges of weight greater than 3, indicating at least 4 diseases are found in common between any two genes.

There exist many forms of layout that are available through existing R plotting packages (such as in ggraph). Depending on the intended use of the graph, this can be useful, for example when wanting to view the relationship between different highly connected nodes, such as in Fig. 5.5A. Most force-directed layout algorithms (such as Fruchterman-Reingold or Kamadi-Kawai) will place densely connected nodes in the centre of the graph, making viewing their interaction relatively challenging. However, imposing a grid structure on the graph allows for greater clarity with the added benefit of considerably faster rendering time. However, such a grid layout is less feasible when the number of nodes increases to a much greater degree, and if you are more concerned with less connected components of the graph, the alternative spacing algorithms are far better.

#### 5.3.2.2 Visualising specific disease environments

As previously mentioned, it is possible to zoom in on areas of the graph above, however it can also be advantageous to filter out unwanted nodes. For this, we have produced a function to limit any graph to a specific neighbourhood around a nominated node or nodes. This allows for a narrower view of a disease area and allows for more human readable/interpretable plots to be rendered. It is also possible to project any networks for subsets of greater than or equal to the 2nd order, however, as visible in Figure 5.6 panels A through C, increasing orders (especially if highly connected nodes are present) quickly leads to “hairball” plots. As previously discussed, this can be ameliorated through the projection of plots into their unipartite form. Additionally, we would expect a standard use-case of such graphs to be to examine small, concise gene/disease lists, therefore naturally producing clearer plots.

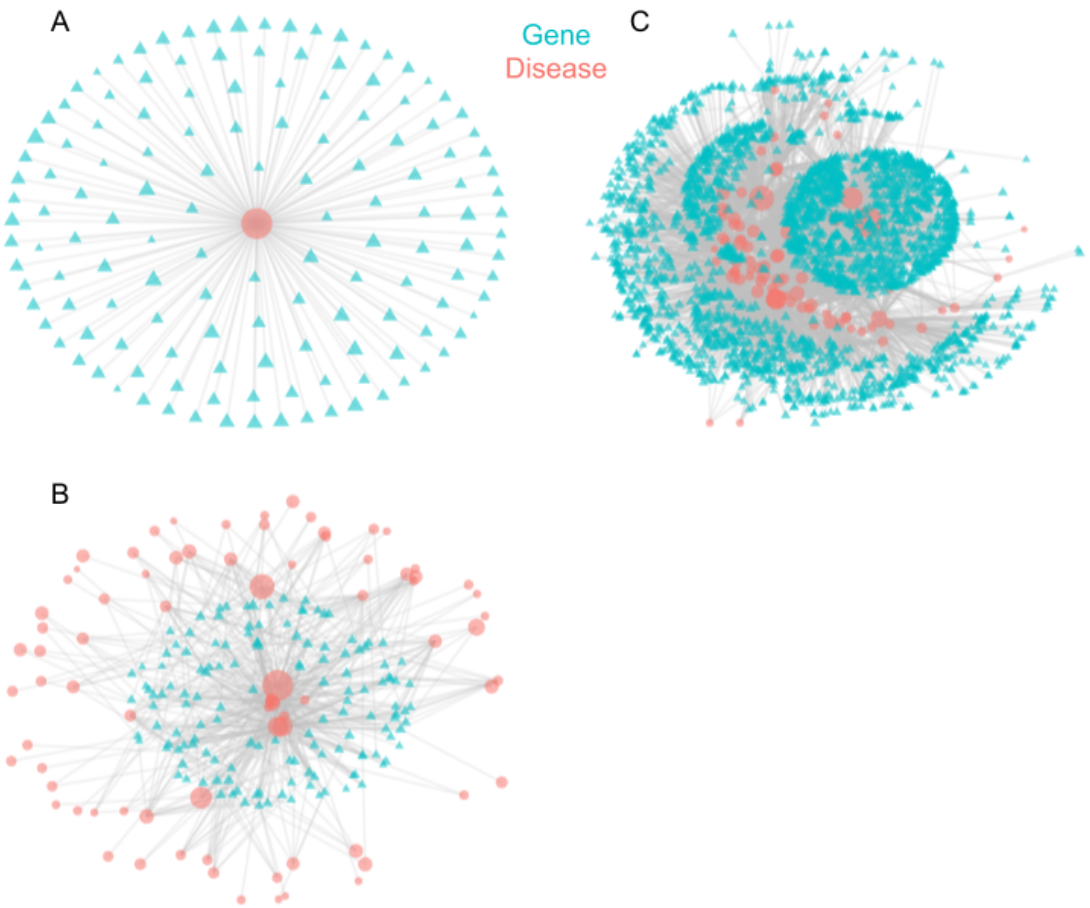



**Figure 5.6 - Bipartite network graphs of 1st (A), 2nd (B) and 3rd (C) order neighbourhoods of the dysmorphic disorder "Clefting".** Disease node (in orange) size is scaled according to the number of genes (in teal) associated with it and its associated phenotypes.

Further to this, it is possible to highlight areas of the network that are of interest. For example, in the case above, it may be advantageous to explore which genes may already be the targets of approved medications. These can then be recoloured within the network graph through each order of interaction. Figure 5.7A + B shows such examples, using data from the druggable gene database, Pharos DB (https://pharos.nih.gov/). These data are not stored within the database itself to reduce maintenance overhead, and as we consider this to be an unnecessary duplication of data. However, given the array of different identifiers stored in the GELdb, it is possible to map to other databases with ease.

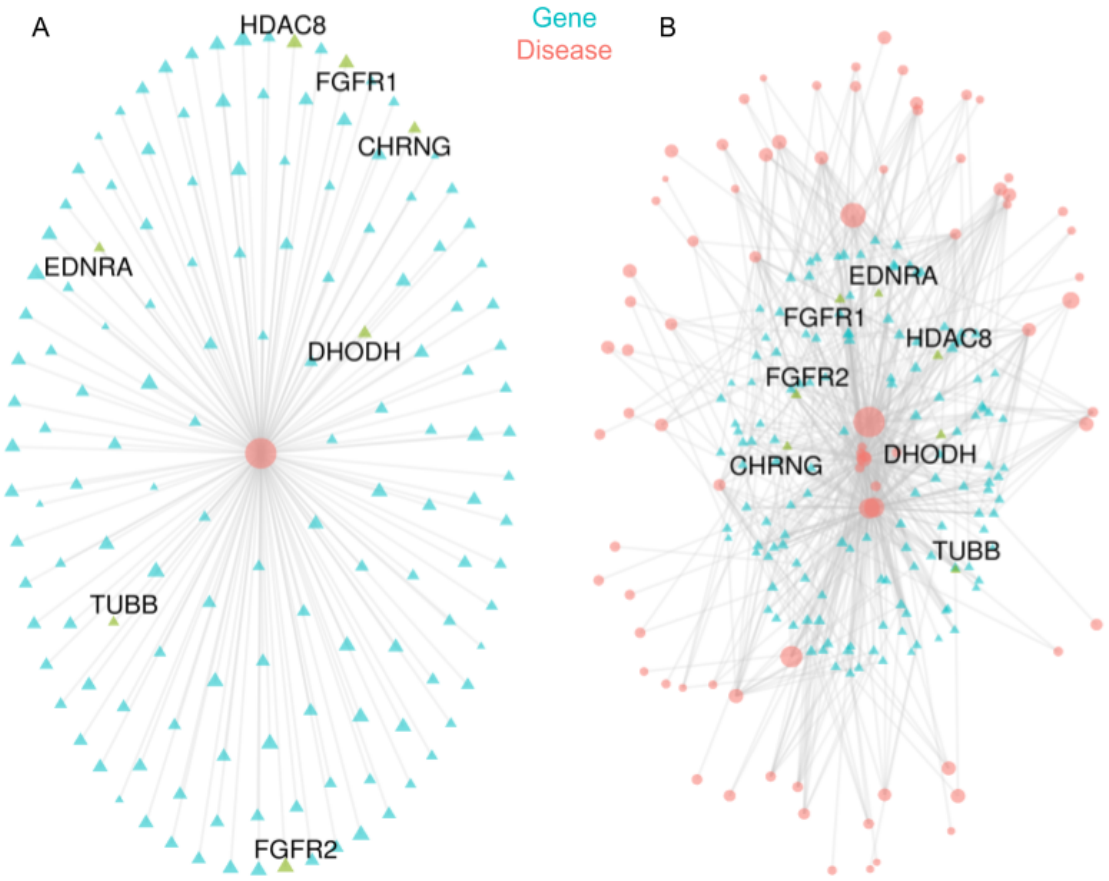



**Figure 5.7 - Bipartite network graphs of 1st (A), 2nd (B) order neighbourhoods of the dysmorphic disorder "Clefting".** As in Figure 5.6, disease node size is scaled according to the number of associated genes. Here, genes that are the target of approved drugs (according to Pharos DB) are named and labelled in green, the rest in teal.

### 5.3.2.3 Building summary data pages

Due to the difficulty in visualising the data in a graphical way across the entire dataset, we decided to create disease-based summary pages. These are HTML documents, in which interactive graphs are embedded with additional summary statistics and data regarding the disease in question. These are all early development stage documents, that would require additional work to be production ready, but serve as illustrative conceptual examples.

Figure 5.8 shows an example of such a page. Here we display information on the dental disorder Amelogenesis Imperfecta. The first pieces of information displayed in this data page are the disease groups and subgroups.
Following this, we display all ‘green’ genes from PanelApp in a scrollable table.

A

# Amelogenesis Imperfecta

## Setup and import

```
# Libs ---------------------------------------------------------------
pkgs <- c('dplyr',
          'RMySQL',
          'magrittr',
          'tidyr',
          'tidygraph',
          'tidyverse',
          'ggraph',
          'scales',
          'knitr',
          'knitrBootstrap',
          'kableExtra',
          'gridExtra',
          'htmlwidgets')

if (!require("pacman")) install.packages('pacman')
pacman::p_load(pkgs, character.only = T)

rm(pkgs)

# Functions ----------------------------------------------------------
source('~/GEL_DB/DB_vis/Scripts/graph_funcs.R')

# Data import --------------------------------------------------------
BashArgs = commandArgs(trailingOnly=TRUE)

source('~/GEL_DB/DB_setup/Scripts/db_connect.R')

con1 <- start_con(BashArgs)
db.tabs <- dbListTables(con1)


db.list <- pull_data(db.tabs, con = con1)

plots <- list() #to save output plots
titles <- list() #titles for plots
```

## Analysis start

### DB overview

```
tab.num <- dbGetQuery(con1, 'SELECT table_name, table_rows
  FROM INFORMATION_SCHEMA.TABLES
  WHERE TABLE_SCHEMA = "GEL";')

tab.num %<>% arrange(desc(table_rows))
names(tab.num) <- c('Table','Num rows')
```

```
knitr::kable(tab.num)
```

| Table | Num rows |
|---|---|
| pheno_to_drug | 163294 |
| pheno_to_gene | 115929 |
| disease_to_drug | 8080 |
| dis_to_pheno | 6805 |
| drug_to_gene | 4550 |
| drug | 3263 |

B

## Network generation

```
# Amelogenesis imperfecta ------------------------------------------------
```

### Gene-Disease network

```
gd <- GeneDisNet()
```

```
## Retrieving Gene-Disease interaction network
```

```
## Joining, by = "id_disease"
```

```
## Joining, by = "id_gene"
```

```
## # A tbl_graph: 2244 nodes and 4631 edges
## #
## # A bipartite simple graph with 5 components
## #
## # Node Data: 2,244 x 4 (active)
##   name                          degree size.degree type
##   <chr>                          <dbl>       <dbl> <chr>
## 1 Intellectual disability          882       6.00  Disease
## 2 Undiagnosed metabolic disorders  543       3.69  Disease
## 3 Unexplained skeletal dysplasia   338       2.30  Disease
## 4 Mitochondrial disorders          206       1.40  Disease
## 5 Hereditary ataxia                116       0.783 Disease
## 6 Arthrogryposis                   115       0.776 Disease
## # ... with 2,238 more rows
## #
## # Edge Data: 4,631 x 3
##    from    to hgnc_symbol
##   <int> <int> <chr>
## 1   411   788 KDM6A
## 2   412   788 KMT2D
## 3    52   236 SALL1
## # ... with 4,628 more rows
```

```
#Look at just local neighbourhood for disease of interest
nodes <- 'Amelogenesis imperfecta'


local_neighbourhood_tab(graph = gd, nodes = nodes, order = 1)
```

| name | degree |
|---|---|
| Amelogenesis imperfecta | 26 |
| ENSG00000127980 | 8 |
| ENSG00000124587 | 8 |
| ENSG00000177706 | 5 |
| ENSG00000141485 | 3 |
| ENSG00000067836 | 2 |
| ENSG00000167323 | 2 |

C

```
ln.gd <- local_neighbourhoods(nodes, gd, order = 3)

ln.gd
```

```
## # A tbl_graph: 1561 nodes and 2393 edges
## #
## # A bipartite simple graph with 1 component
## #
## # Node Data: 1,561 x 4 (active)
##   name                            degree size.degree type
##   <chr>                            <dbl>       <dbl> <chr>
## 1 Intellectual disability            882        6.00  Disease
## 2 Undiagnosed metabolic disorders    543        3.69  Disease
## 3 Unexplained skeletal dysplasia     338        2.30  Disease
## 4 Arthrogryposis                     115        0.776 Disease
## 5 Inherited white matter disorders   109        0.736 Disease
## 6 Epileptic encephalopathy           106        0.715 Disease
## # ... with 1,555 more rows
## #
## # Edge Data: 2,393 x 3
##    from    to hgnc_symbol
##   <int> <int> <chr>
## 1    17    18 CHD7
## 2    17   172 FGFR3
## 3    17    81 FAM20C
## # ... with 2,390 more rows
```

```
plots[['ln.gd']] <- plot_graph(ln.gd, OutTheme = 'light', summaryTab = F, showLabs = T, nodeType = 'Disease')
titles[['ln.gd']] <- 'amelogenesis_gdis'
```

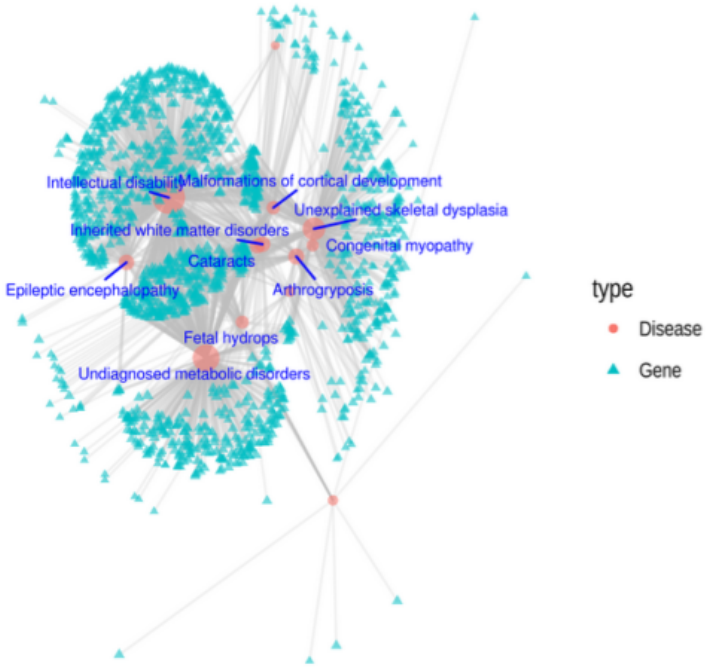



**Figure 5.8 - An example of a disease summary page.** A) The example shown is for Amelogenesis imperfecta. B) Summaries on the gene-disease network across the GEL ontology, and in relation to amelogenesis imperfecta are shown. C) A visualisation of the gene-disease network for this disease is displayed, with genes as teal triangles and diseases as orange circles. The size of the node denotes the centrality degree, and the ten most connected diseases are labelled in blue.

Figure 5.8 provides an example of the output in a simple report form generated about a disease. The user needs to simply change the input disease in Fig. 5.8B (labelled as 'nodes'), and then run the script as is. Panel A shows the packages and paths to helper functions that are required for the proper running of the script. When invoked, the script will ask for the user's username and password for access to the database. A brief overview of what is contained within the database is then displayed (this has been truncated to save space within this figure). Following this, a generic function is called to build a bipartite network comprising diseases and their associated genes (Fig. 5.8B). The "local neighbourhoods" functions restrict the network to the node of interest, in this case, the disease 'amelogenesis imperfecta', and produces related tables and networks. The "order" argument of the functions defines how restricted the relationship must be. E.g. - 1 will give only immediate neighbours to the disease, or the genes that are implicated with it, and 2 will expand to include the diseases that share these genes. The output table will list nodes by their degree, so in this case, the most connected gene, peroxisomal biogenesis factor 1 (PEX1,'ENSG00000127980') has eight connections, meaning it is linked to the disease 'amelogenesis imperfecta' and 7 other diseases within the GELDdb. Following this, Fig. 8C expands the network to include 3rd order relationships, with disease node size scaled according to the number of edges it has. In this example we can see that the "intellectual disability" and "undiagnosed metabolic disorders" provide the lion's share of gene connections. A further command could be included to remove these nodes from the network, thereby restricting to diseases that may be more specific than these relatively generalised disorders.

Other prebuilt networks that are included in a standard report are drug-disease networks, a table built on inferred relationships between drugs and diseases, and disease-phenotype networks. The aim of including these is to quickly identify potentially interesting compounds or phenotypes to investigate. In the case of amelogenesis imperfecta, the first order drug with the highest degree is Marimastat, a broad spectrum metalloproteinase inhibitor which phenocopies (mimics the disease presentation) amelogenesis imperfecta [10,11].

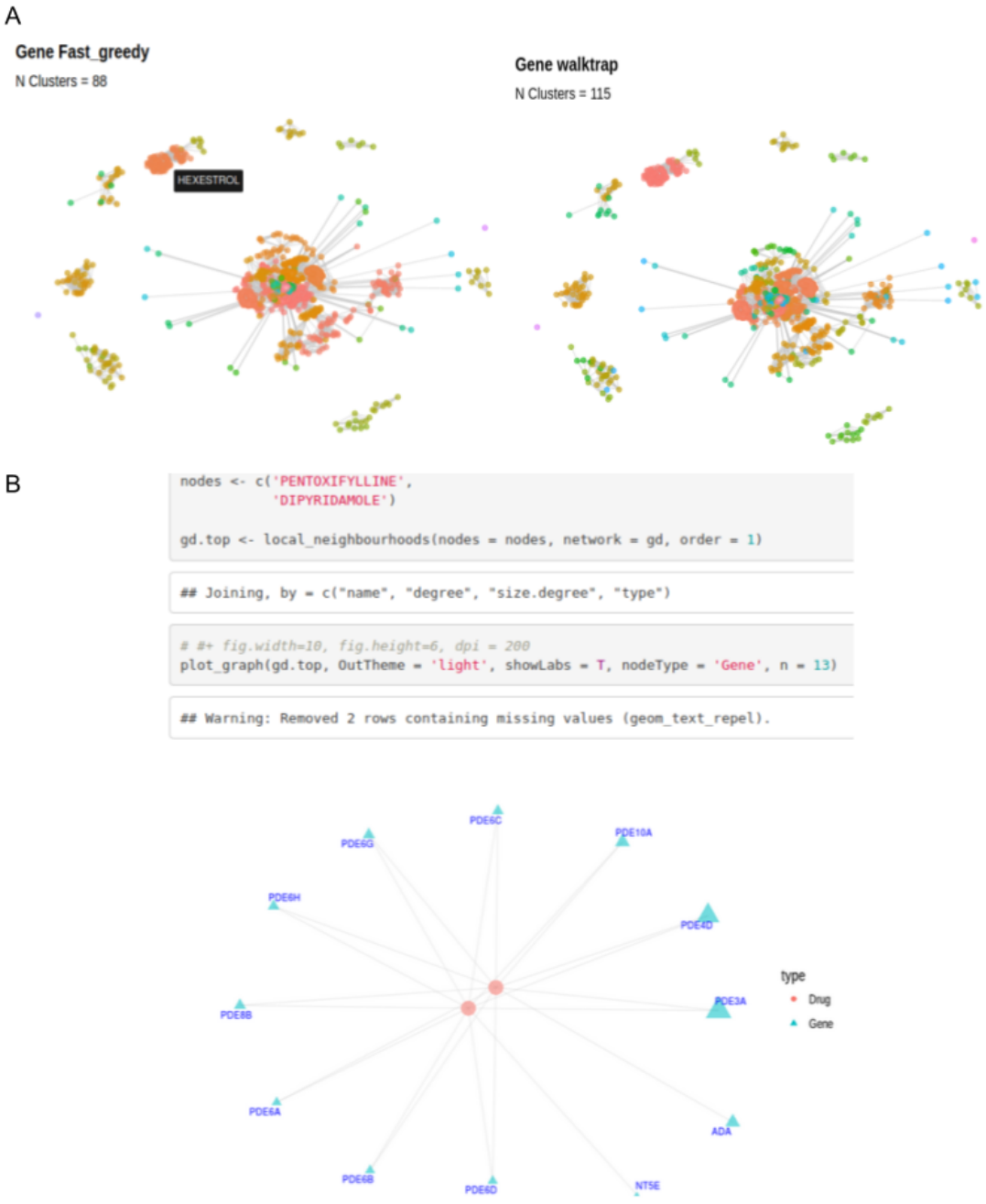



**Figure 5.9 - A continuation of the disease summary page - visualisations of drug-gene network clustering**. A) Two implementations of network clustering algorithms are displayed. The number of clusters along with the implementation name are given as titles. The user can hover over nodes to identify specific drugs and zoom on specific areas. Nodes are coloured according to their clusters. B) Upon identification of drugs of interest, a subgraph and resulting network can be plotted using predefined functions.

Following this, optional clustering algorithms can be run via their igraph (R package, v 1.2.1) implementations. For example the walktrap or fast greedy algorithm can be used to cluster groups of genes based on the drug interactions they share. All such algorithms are run exclusively on projected networks, as the majority of algorithms do not account for bipartite structures. Fig. 5.9A shows a generalised version of such a network, with all genes associated with diseases that have drug associations in DGIdb. Fig. 5.9B shows a subgraph derived from this graph. It is clear from this example that the drugs pentoxifylline and dipyridamole share many of the same interaction partners, namely, members of the phosphodiesterase family. However both drugs show unique partners too, in the form of ADA and NT5E. This could make either drug a good candidate to also target the 'missing spokes' in this wheel.

| GEL disease | Drug name | Number of genes |
|---|---|---|
| Mitochondrial disorders | metformin | 19 |
| Undiagnosed metabolic disorders | metformin | 19 |
| Inherited white matter disorders | metformin | 8 |
| Structural basal ganglia disorders | metformin | 7 |
| Arthrogryposis | rapacuronium | 5 |
| Arthrogryposis | suxamethonium | 5 |
| Congenital myaesthenia | rapacuronium | 5 |
| Congenital myaesthenia | suxamethonium | 5 |
| Epileptic encephalopathy | acamprosate | 5 |
| Epileptic encephalopathy | amifampridine | 4 |
| Epileptic encephalopathy | fampridine | 4 |
| Epileptic encephalopathy | guanidine | 4 |

**Table 5.1 - A Summary table of GeL disease-drug pairings ordered by the number of shared genes**. Genes shared are genes associated with GeL diseases through

PanelApp curation that are also targets of approved drugs with indications in other conditions. Only the first 12 sets of associations are displayed here.

Table 5.1 shows data not included within the example data page, namely, a cohort-wide view of drug-gene-disease associations. Here, diseases and drugs are linked if they share PanelApp ‘green’ genes that are targeted by approved drugs for other indications. This simple guilt-by-association (GBA) approach opens possibilities for drug repurposing, based on very simple queries of the database. The associations from Table 5.1 will be explored in the discussion.

### 5.3.3 LoF labelling of PanelApp genes

Linking of this data to the LoF data collated in chapters 3 and 4 has the potential to highlight targets that could be prioritised for drug development. We matched genes according to gene symbols, selecting all genes with associations to GEL diseases from GELDdb.

| Phenotype | N |
|---|---|
| Benign | 192 |
| Deleterious | 2737 |
| pBenign | 23 |
| pNotBenign | 450 |

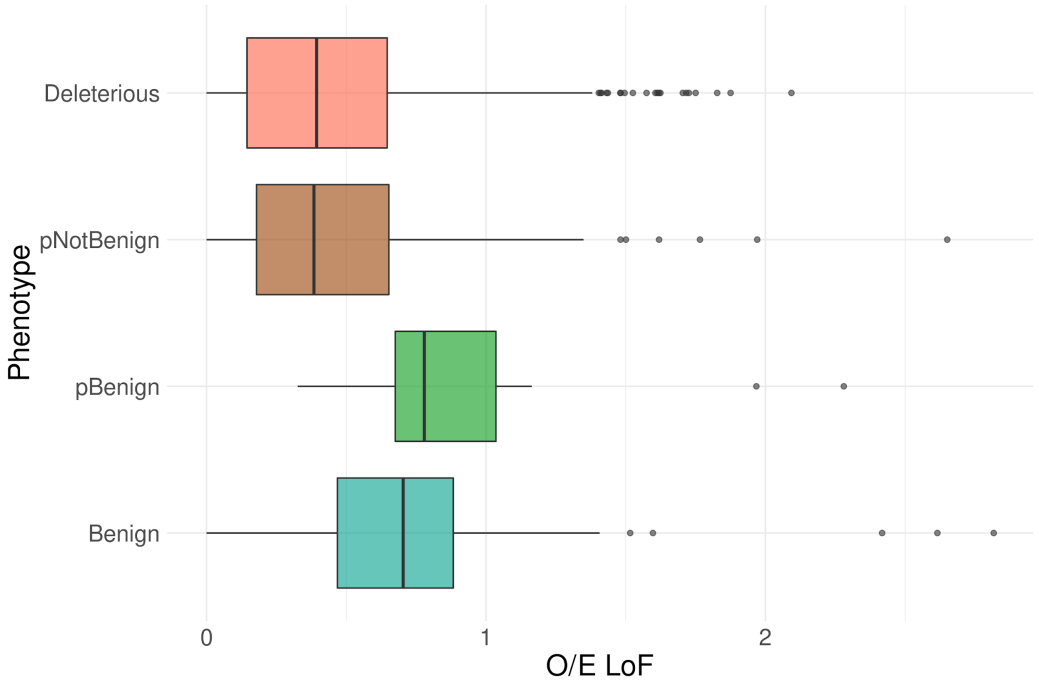



**Figure 5.10 - The breakdown of constraint within GELDdb disease genes.** Genes associated with GEL diseases from GELDdb were labelled according to phenotypes from Chapters 3 and 4. The majority of genes are deleterious genes, with increased levels of constraint.

As expected, Benign and pBenign genes make up the minority of genes in our gene-set. (Fig. 5.10) As genes are classified as benign if they are found within specific sequenced cohorts in healthy individuals, as previously discussed in chapter 3, it is possible for a benign gene to have a disease association. Nearly all of the benign and pBenign genes (e.g. ATR, BLM, SYNE1 and NPC2) are listed as green genes in multiple PanelApp disorders, for generally more complex disorders. This could explain the appearance of benign genes in a list of genes derived primarily from patients with monogenic rare diseases.

| Target | Approved Drug |
|---|---|
| ADRA2B | Carvedilol, Apraclonidine hydrochloride, Bromocriptine, Cabergoline, Yohimbine [...] |
| AKR1D1 | Finasteride |
| ALDH2 | Disulfiram, Guanidine |
| ALK | Ceritinib, Crizotinib, Gilteritinib, Fostamatinib, Alectinib [...] |
| AURKC | Fostamatinib |
| BCR | Ponatinib, Dasatinib, Imatinib, Bosutinib, Ponatinib hydrochloride |
| BDNF | Esketamine |
| CYP2C9 | Benzbromarone, Diacerein |
| GHRHR | Tesamorelin, Sermorelin acetate |
| GJB2 | Carbenoxolone |

**Table 5.2 - Opportunities for repurposing within the GELDdb.** The first ten pBenign or Benign targets with approved drugs (data from GDIdb) are listed. Targets with more than 5 approved drugs are concatenated for brevity.

The 215 observed or predicted benign genes present an opportunity to identify drug targets for which perturbation is more likely to be tolerated than other targets within the database.

Furthermore, of these targets, 100 are druggable and 27 are clinically actionable (data from DGIdb). Therefore there exists in this set of genes a large pool of tractable drug targets, which are likely safer than deleterious drug targets. Of these targets, some are already drugged; table 5.2 shows examples of these drug repurposing opportunities, with targets from our benign and pBenign categories listed if they have interacting, approved drugs already in the market. A total of 33 such targets exist and this could present an opportunity to rapidly bring medications to currently unserved patients and their diseases. However, as previously stated in chapters 2,3 and 4, this information is intended for hypothesis formation and target prioritisation only, and would require extensive validation.

# 5.4 Discussion

In this chapter, we have sought to create a dataset to facilitate drug repurposing. By pulling data together from different levels of the biological dogma, we hope to more closely reflect biological truth within a system, i.e. to capture a degree of the interconnectedness of biological processes. Such databases exist elsewhere, however we chose here to focus on the rare disease space, and more specifically on diseases that are part of the Genomics England 100KGP cohort. The additional benefit of this group of diseases is the ancillary database PanelApp, which has generated a curated list of high confidence disease-causing genes [9].

This dataset is intended to facilitate drug repurposing and therefore has focussed on drug related datasets such as SMILEs and drug interaction data, as well as the interaction networks of drug targets. Currently the methods of actual drug repurposing that have been applied are limited to GBA approaches. This simplistic method however can still yield results, as it seeks to find lines of evidence for associations between genes, their products, and diseases across multiple levels of biological dogma. As much of the results discussed thus far show the implementation and output of this database, the discussion will pick some examples of drugs that show promise due to their GBA. Additional work would build on this foundation to produce more sophisticated approaches to drug repurposing, however we aim here to merely illustrate a proof of concept.

In addition to this database, we have focussed on providing helper functions in order to query the database. We designed these functions with a focus on networks, as networks provide a powerful way of combining disparate data of this type.

## 5.4.1 Literature supporting GBA targets

### 5.4.1.1 Acamprosate

An association between epileptic encephalopathy and the antidipsotropic (anti-alcohol dependence) drug acamprosate is observed in Table 5.1. Genes shared are genes associated with GEL diseases through PanelApp curation that are also targets of approved drugs with indications in other conditions. Epileptic encephalopathies are a group of brain disorders in which continuing seizures during the course of brain maturation are thought to lead to a progressive cognitive deterioration [12]. Despite the devastating impact on neurological function incurred, there are only limited treatment options [13]. Overall, epilepsies are one of the most common neurological disorders; often characterised by recurrent synchronous discharges of the brain. With as many as eighty percent of cases thought to be due to genetic factors, increases in genetic testing has led to a deeper understanding of the underlying biology of this spectrum of disorders [14–16]. A large number of genes have associations with epilepsy (there are currently 425 green genes listed in the “Genetic epilepsy syndromes” panel (version 2.280) on PanelApp), reflecting the heterogeneity of the disorder. Numerous variants of interest have been found in N-methyl-D-aspartate receptors (NMDAR), particularly in childhood epilepsy [17–19]. These ionotropic glutamate receptors are generally composed of four subunits, encoded by 3 differing gene families. Each of these subunits and genes have been found to have likely causative variants in epilepsy [19–28]. Of these, the GRIN3b gene is labelled as benign according to our chapter 3 definition. This gene, with a LOEUF score of 0.98 also has several homozygous pLoF confirmed through manual curation (see chapter 2) and is the only gene target of acamprosate with this benign designation. This likely makes it the safest target of this functional group, and drugs with high affinity for this target could be prioritised. However we are unable to comment on what level of efficacy a drug targeting this GRIN3B as its primary target would have.

Despite being approved for the maintenance of abstinence in alcohol dependent individuals, the mechanism of action (MoA) of acamprosate is not fully understood. It is thought to reduce urges to drink through the reduction of alcohol withdrawal induced excitotoxic neuronal cell death [29,30]. This is likely achieved by blocking NMDAR as already discussed, although acamprosate is also thought to indirectly modulate γ-aminobutyric acid type A receptor transmission [31]. There are currently no clinical trials

investigating the use of acamprosate in epileptic encephalopathies, or any epileptic disorders, however it is currently being investigated for neurological disorders such as autism spectrum disorders (phase III, trial NCT01813318) and Fragile X syndrome (phase III, trial NCT01911455) and schizophrenia (phase IV biomarker study, trial NCT00688324) amongst others. Evidence also suggests acamprosate, in combination with baclofen (also a repurposed drug) may be promising in the treatment of Parkinson's disease [32–34].

Acamprosate is the most prescribed drug by the NHS for the treatment of alcohol dependence in combination with counselling [35] . Furthermore, the side effect profile of acamprosate is well understood, and this drug is well tolerated at therapeutic doses [36–38]. All of the factors listed above suggest acamprosate may be a viable therapeutic agent in the treatment of epileptic encephalopathies.

## 5.4.2. Phenocopies

A phenocopy is defined as a non-hereditary, environmentally induced phenotype that mimics that of a genotype-induced phenotype in another individual . Phenocopies have been invaluable in understanding the biology of living organisms, with technologies such as RNAi, and small molecule inhibitors having allowed for the study of diseases. They can also be used to assist in lead-optimisation, and therefore are worthy of consideration when identifying compounds of interest [39].

#### 5.4.2.1.1 Metformin

Our largest hit with respect to shared gene interaction is the drug Metformin and the GEL diseases 'Mitochondrial disorders' and 'Undiagnosed metabolic disorders'. Both of these share 19 genes with PanelApp genes with approved drugs for other indications. Metformin is a widely used type 2 diabetes drug prescribed as a first-line drug for people not responding to dietary change. It is widely used, with over 120 million people taking the drug world-wide [40]. It reduces the risk of hyperglycemia by reducing hepatic gluconeogenesis, and increasing insulin sensitivity through the increase in peripheral glucose uptake [41]. It functions by accumulating within the mitochondria and inhibiting the Complex 1 of the respiratory chain, a process key in the production of ATP [42], which is in turn, required in large quantities for gluconeogenesis. Other possible mechanisms have also been proposed, however all are similar in their site of action, with effects due to activity within mitochondria [41].

Side effects of Metformin often reflect this MoA, with type A reactions including physical weakness, muscle pain and hypoglycemia [41,43]. With this in mind, it is clear why Metformin shares so many genes with mitochondrial and metabolic disorders. This association is an example of a drug phenocopying the disease of interest. As such, this indicates that such a drug may only serve to exacerbate the conditions in question. However, this also indicates that an agonist may, in this context, serve to ameliorate these disorders. The GEL diseases 'Inherited white matter disorders' and 'Structural basal ganglia disorders' share fewer genes with Metformin, with 8 and 7 total genes respectively. This cluster of disorders also stem from improper mitochondrial function and therefore this also likely reflects an instance of metformin phenocopying a genetic disorder through type A adverse events.

Further examples of phenocopying drugs exist in Rapacuronium bromide and other non-depolarising and polarising neuromuscular blocking drugs, including Suxamethonium, Doxacurium, Mivacurium, Pancuronium, Rocuronium, Atracurium, and Vecuronium which are linked to arthrogryposis multiplex congenita (AMC). This condition, characterised by muscle weakness and fibrosis [44], is a nonprogressive congenital disorder resulting in joint contractures in numerous body areas [45]. The drugs in question act through the competitive antagonism of post-junctional acetylcholine binding (or other related mechanisms), resulting in muscle relaxation [46]. Study of the effect of such drugs on chick embryonic development induces flaccid paralysis and induces deformities in limb development in ways that resemble AMC [47–49]. Despite this, these drugs are considered safe to use, even in neonatal contexts, such as in caesarian section deliveries.

Whilst examples of phenocopying drugs are not useful in the effort to find drugs that may provide benefit to a patient, they are important as a proof of concept. As many of these drugs produce side effects through intended MoA, a lack of overlap between these disorders and these drugs would simply indicate that we are not capable of identifying mechanistically related associations. Whilst in a drug screening exercise, one may wish to filter out such drugs, here we have explored them to illustrate this broader point. Finally, phenocopies provide the opportunity to explore designing drugs to produce the opposite effect, e.g. and agonist in place of an antagonist, in order to treat the disease in question.

### 5.4.3 Future directions

The database will be introduced to the Genomics England research environment, where it will be freely available for researchers. The exact form of the database has not yet been decided, however it will likely be adapted to a graph database to facilitate further inference of previously unobserved connections. It will be further integrated with genomic data through the addition of variant data to the gene data supplied by PanelApp. This means predicted effects of variants could be used to further narrow down avenues of interest. For example, a gain-of-function variant causing a disease phenotype would be better treated with an antagonist, as opposed to a loss-of-function variant which may require a target agonist. There is much that can be added to the database to increase its utility, including adding the outputs of additional, more complex approaches to drug repurposing. This would include ML approaches to link inference, allowing for some ‘filling in of the gaps’, where data are not necessarily currently available. The continuation of this work remains beyond the scope of this work.

## 5.5 Conclusion

Due to the target centric nature of drug discovery over the last 60 years, there exist many gaps in our knowledge about our drugs, and their full complement of targets. This presents us with an opportunity to fill these gaps in, simply by further examining the data we already have. This is particularly valuable in the context of rare diseases, as it presents a cost effective and relatively rapid approach to find new promising targets. In this chapter, we assembled GELDdb, a database focussed on the disease space from the GEL rare disease programme. By enriching these data, specifically around areas of druggability, and existing medications, we have uncovered novel possible indications for approved drugs. Examples such as acamprosate for the treatment of epilepsies may show promise, and should be explored further. Whilst all of the information contained within is for hypothesis formation, inclusion of this dataset within the GEL research environment may point researchers to targets that had not previously been considered for their disease of interest.

# 5.6 References


1. Hancock, J. M. Editorial: biological ontologies and semantic biology. *Frontiers in Genetics* vol. 5 (2014).
2. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
3. Köhler, S. *et al.* Expansion of the Human Phenotype Ontology (HPO) knowledge base and resources. *Nucleic Acids Res.* **47**, D1018–D1027 (2019).
4. Amberger, J., Bocchini, C. A., Scott, A. F. & Hamosh, A. McKusick's Online Mendelian Inheritance in Man (OMIM). *Nucleic Acids Res.* **37**, D793–6 (2009).
5. Vesztrocy, A. W. & Dessimoz, C. A Gene Ontology Tutorial in Python. in *The Gene Ontology Handbook* (eds. Dessimoz, C. & Škunca, N.) 221–229 (Springer New York, 2017).
6. The Most Popular Databases 2019. https://www.explore-group.com/blog/the-most-popular-databases-2019/bp46/ (2019).
7. Broy, M. & Denert, E. *Software Pioneers: Contributions to Software Engineering*. (Springer Science & Business Media, 2012).
8. CS403: Database Normalization. https://learn.saylor.org/mod/page/view.php?id=23139.
9. Martin, A. R. *et al.* PanelApp crowdsources expert knowledge to establish consensus diagnostic gene panels. *Nat. Genet.* **51**, 1560–1565 (2019).
10. Feng, J. *et al.* Binding of amelogenin to MMP-9 and their co-expression in developing mouse teeth. *J. Mol. Histol.* **43**, 473–485 (2012).
11. Millar, A. W. *et al.* Results of single and repeat dose studies of the oral matrix metalloproteinase inhibitor marimastat in healthy male volunteers. *Br. J. Clin. Pharmacol.* **45**, 21–26 (1998).
12. Panayiotopoulos, C. P. *Epileptic Encephalopathies in Infancy and Early Childhood*

*in Which the Epileptiform Abnormalities May Contribute to Progressive Dysfunction*. (Bladon Medical Publishing, 2005).

13. Nariai, H., Duberstein, S. & Shinnar, S. Treatment of Epileptic Encephalopathies: Current State of the Art. *J. Child Neurol.* **33**, 41–54 (2018).
14. Hildebrand, M. S. *et al.* Recent advances in the molecular genetics of epilepsy. *J. Med. Genet.* **50**, 271–279 (2013).
15. Perucca, P., Bahlo, M. & Berkovic, S. F. The Genetics of Epilepsy. *Annu. Rev. Genomics Hum. Genet.* **21**, 205–230 (2020).
16. Berkovic, S. F. Genetics of Epilepsy in Clinical Practice. *Epilepsy Curr.* **15**, 192–196 (2015).
17. Burnashev, N. & Szepetowski, P. NMDA receptor subunit mutations in neurodevelopmental disorders. *Curr. Opin. Pharmacol.* **20**, 73–82 (2015).
18. Yuan, H., Low, C.-M., Moody, O. A., Jenkins, A. & Traynelis, S. F. Ionotropic GABA and Glutamate Receptor Mutations and Human Neurologic Diseases. *Mol. Pharmacol.* **88**, 203–217 (2015).
19. Xu, X.-X. & Luo, J.-H. Mutations of N-Methyl-D-Aspartate Receptor Subunits in Epilepsy. *Neurosci. Bull.* **34**, 549–565 (2018).
20. Lemke, J. R. *et al.* Delineating the GRIN1 phenotypic spectrum: A distinct genetic NMDA receptor encephalopathy. *Neurology* **86**, 2171–2178 (2016).
21. Ohba, C. *et al.* GRIN1 mutations cause encephalopathy with infantile-onset epilepsy, and hyperkinetic and stereotyped movement disorders. *Epilepsia* **56**, 841–848 (2015).
22. Endele, S. *et al.* Mutations in GRIN2A and GRIN2B encoding regulatory subunits of NMDA receptors cause variable neurodevelopmental phenotypes. *Nat. Genet.* **42**, 1021–1026 (2010).
23. Lemke, J. R. *et al.* Mutations in GRIN2A cause idiopathic focal epilepsy with rolandic spikes. *Nat. Genet.* **45**, 1067–1072 (2013).
24. Dimassi, S. *et al.* A subset of genomic alterations detected in rolandic epilepsies

contains candidate or known epilepsy genes including GRIN2A and PRRT2. *Epilepsia* **55**, 370–378 (2014).

25. Boutry-Kryza, N. *et al.* Molecular characterization of a cohort of 73 patients with infantile spasms syndrome. *Eur. J. Med. Genet.* **58**, 51–58 (2015).
26. Lal, D. *et al.* Investigation of GRIN2A in common epilepsy phenotypes. *Epilepsy Res.* **115**, 95–99 (2015).
27. Sibarov, D. A. *et al.* Functional Properties of Human NMDA Receptors Associated with Epilepsy-Related Mutations of GluN2A Subunit. *Front. Cell. Neurosci.* **11**, 155 (2017).
28. Swanger, S. A. *et al.* Mechanistic Insight into NMDA Receptor Dysregulation by Rare Variants in the GluN2A and GluN2B Agonist Binding Domains. *Am. J. Hum. Genet.* **99**, 1261–1280 (2016).
29. Tsai, G. & Coyle, J. T. The role of glutamatergic neurotransmission in the pathophysiology of alcoholism. *Annu. Rev. Med.* **49**, 173–184 (1998).
30. Frye, M. A. *et al.* Anterior Cingulate Glutamate Is Reduced by Acamprosate Treatment in Patients With Alcohol Dependence. *J. Clin. Psychopharmacol.* **36**, 669–674 (2016).
31. Kalk, N. J. & Lingford-Hughes, A. R. The clinical pharmacology of acamprosate. *Br. J. Clin. Pharmacol.* **77**, 315–323 (2014).
32. Hajj, R. *et al.* Combination of acamprosate and baclofen as a promising therapeutic approach for Parkinson's disease. *Sci. Rep.* **5**, 16084 (2015).
33. Hajj, R. Parkinson Disease Therapies and Drugs. in *Pathology, Prevention and Therapeutics of Neurodegenerative Disease* (eds. Singh, S. & Joshi, N.) 151–158 (Springer Singapore, 2019).
34. Chumakov, I. *et al.* Combining two repurposed drugs as a promising approach for Alzheimer's disease therapy. *Sci. Rep.* **5**, 7608 (2015).
35. Part 3: Alcohol-related prescriptions - NHS Digital. *NHS Digital* https://digital.nhs.uk/data-and-information/publications/statistical/statistics-on-alcoh

ol/2018/part-3.

36. Paille, F. M. *et al.* Double-blind randomized multicentre trial of acamprosate in maintaining abstinence from alcohol. *Alcohol Alcohol* **30**, 239–247 (1995).
37. Sass, H., Soyka, M., Mann, K. & Zieglgänsberger, W. Relapse prevention by acamprosate. Results from a placebo-controlled study on alcohol dependence. *Arch. Gen. Psychiatry* **53**, 673–680 (1996).
38. Pelc, I. *et al.* Efficacy and safety of acamprosate in the treatment of detoxified alcohol-dependent patients. A 90-day placebo-controlled dose-finding study. *Br. J. Psychiatry* **171**, 73–77 (1997).
39. Baum, P. *et al.* Phenocopy--a strategy to qualify chemical compounds during hit-to-lead and/or lead optimization. *PLoS One* **5**, e14272 (2010).
40. Viollet, B. *et al.* Cellular and molecular mechanisms of metformin: an overview. *Clin. Sci.* **122**, 253–270 (2012).
41. Rena, G., Hardie, D. G. & Pearson, E. R. The mechanisms of action of metformin. *Diabetologia* **60**, 1577–1585 (2017).
42. Owen, M. R., Doran, E. & Halestrap, A. P. Evidence that metformin exerts its anti-diabetic effects through inhibition of complex 1 of the mitochondrial respiratory chain. *Biochemical Journal* vol. 348 607 (2000).
43. Thomas, I. & Gregg, B. Metformin; a review of its history and future: from lilac to longevity: THOMAS AND GREGG. *Pediatr. Diabetes* **18**, 10–16 (2017).
44. Epstein, J. B. & Wittenberg, G. J. Maxillofacial manifestations and management of arthrogryposis: literature review and case report. *J. Oral Maxillofac. Surg.* **45**, 274–279 (1987).
45. Kimber, E. AMC: amyoplasia and distal arthrogryposis. *J. Child. Orthop.* **9**, 427–432 (2015).
46. Rang, H. P., Dale, M. M. & Ritter, J. M. Local anaesthetics and other drugs that affect excitable membranes. *Pharmacology. Edinburgh: Churchill Livingstone* 665–677 (1995).

47. Lamb, K. J. *et al.* Diverse range of fixed positional deformities and bone growth restraint provoked by flaccid paralysis in embryonic chicks. *Int. J. Exp. Pathol.* **84**, 191–199 (2003).

48. Drachman, D. B. & Coulombre, A. J. Experimental clubfoot and arthrogryposis multiplex congenita. *Lancet* **2**, 523–526 (1962).

49. Osborne, A. C., Lamb, K. J., Lewthwaite, J. C., Dowthwaite, G. P. & Pitsillides, A. A. Short-term rigid and flaccid paralyses diminish growth of embryonic chick limbs and abrogate joint cavity formation but differentially preserve pre-cavitated joints. *J. Musculoskelet. Neuronal Interact.* **2**, 448–456 (2002).

# Chapter 6: Conclusion

In this thesis we have explored the role of human genetics in drug discovery, and ways in which we can use disparate forms of data to infer potential new targets for drug development and repurposing.

First we started with a question of data integrity, how can we trust the variants that we see in sequenced cohorts? The rarity of LoF variation, combined with its generally deleterious nature means that this class of variants is enriched for sequencing artifacts. Tools such as LOFTEE attempt to tackle this problem by applying rules to discern variants that are likely to be real versus those that are not. This tool has been relatively widely adopted, as is shown by its use in the PROMIS, gnomAD and DeCODE studies discussed in this thesis. However, this tool is limited by its rule-base, and does not attempt to capture all of the classes of artifacts one can expect to find. Currently, the gold-standard approach to filtering spurious LoF variants is manual curation. This requires domain knowledge and a lot of time, and is therefore not necessarily the most efficient of approaches. However, beginning with this approach will allow for the generation of positive control datasets, and a greater understanding of the artifacts we can expect to find with these data. Chapter 2 covered the manual curation of homozygous predicted LoF variants that had already been filtered using LOFTEE. This was completed during my placement with the gnomAD team at the Broad Institute in the first of several collaborations in this thesis, and represents a small part of the curation work occurring there. Of the 4,379 variants assessed, 25% were deemed to be spurious (with a further 8% being undetermined); a vast improvement on the 50% that has previously been reported, indicating that LOFTEE is a valuable addition to any curation pipeline. However, the possibility of manually curating all pLoF variants identified is remote, and therefore machine learning approaches to complete this curation must be actively sought. These will build upon such manual curation efforts by using the data gleaned to prioritise specific features for prediction, and providing a gold-standard dataset for testing. The observation from the manual curation that strand bias and last exon do not tend to result in the ruling out of a candidate LoF could be instructive for this. It is also clear that multinucleotide polymorphisms (MNPs) need to be handled as a special case, as pLoF variants that are simply one position in an MNP are almost never truly LoF. Following this work, we continued our investigation of homozygous pLoF variants.

Whilst chapter 2 sought to increase data integrity, chapter 3 aimed to increase data coverage. The number of LoF variants discovered is limited by the number of individuals sequenced. Furthermore, the individuals must be sequenced from diverse backgrounds in order to capture as broad a genetic sample as possible. There are many reasons to want to identify all possible homozygous LoF variants, including the understanding of disease and normal biology. Our interest lies in the hypothesis that homozygous LoF variation is a model of lifelong inhibition of a gene. This is of particular interest in drug development, where we might aim to artificially inhibit a gene-product, and knowledge of the phenotypes associated with such knockouts may be valuable. Additionally, evidence suggests that drug development efforts against targets with homozygous benign LoF are twice as likely to succeed. Therefore in chapter 3, we first catalogued known instances of homozygous LoF in cohorts. As our interest is in drug development, we created a label based on whether the gene was found knocked-out in an apparently healthy individual. These benign knockouts were compared to genes known to be associated with disease and additional features were used to predict which of the remainder could be assigned to this benign group. Features added included data from protein-protein interaction data, and ontology data from the gene ontology. We generated numerous models through the use of a genetic algorithm, finding high degrees of concordance between them (i.e. they selected the same genes as likely benign). The best performing models were an ensemble model and a decision tree model, but we prioritised further investigation in the former model as it had higher recall, a factor we sought to prioritise. Overall, we found a total of 1,744 benign genes in existing literature, and predicted a further 442 pBenign with a relatively high degree of specificity and sensitivity (F-score 0.7). It remains to be seen which of these predictions will hold, and this will no doubt become clear as more sequencing data become available.

With this additional set of potential targets defined, in chapter 4 we aimed to identify whether benign and pBenign genes were good drug targets. We did this in collaboration with Abbvie, an industrial partner, by studying historic drug data, and comparing our labels with data from previous studies detailing the effect of genetic knowledge on drug discovery, and against the LOEUF score, a metric of constraint. We showed that benign LoF labels are indeed predictive of drug development success, and that it is more predictive of success than either of the other metrics when looking at non-oncology indications only. The splitting of non-oncological and oncological targets into separate models showed that the constraint metric LOEUF is strongly predictive of

oncology indication probability of success at all trial phases of development. This is in contrast to benign LoF labels, which provide no additional information in oncological indications. This highlights that these two complementary features, both based on analysis of LoF variation, inform drug discovery in distinct ways. We then studied the properties of manually and predicted genes to show enrichment of olfactory receptors within our pBenign data. This class of genes is historically understudied, despite being part of the highly druggable GPCR family, and growing evidence suggesting they may modulate numerous physiological processes. We highlight two possible olfactory receptors, OR2T10 and OR2T11, which our data suggest should be safe to target, and may modulate disease pathology in both psoriasis and atopic dermatitis. This chapter serves to further confirm that genetic information is valuable in the drug discovery process, and demonstrates the value of LoF data in this process in a clinical trial phase dependent manner. It also highlights a possible rich seam of putatively safe drug targets in the less explored members of the GPCR family.

In Chapter 5, we incorporated all the data from previous chapters and applied it to the rare disease programme of the 100KGP. By building on a primitive disease ontology describing diseases in broad classes, we built a relational database to approximate the disease space in question. We first enriched this ontology with data from the PanelApp website, providing an expertly curated set of associated genes for each of the rare diseases included. We then added data relating to protein-protein interaction, approved medications, drug structures and more. In particular we focussed on data that may be of use for target identification and drug repurposing. We then wrote a series of functions and tools to simplify the querying of this database, so that data germain to any one disease or area of interest could be easily accessed and analysed. We demonstrated the value of assembling such data in this manner with this example of acamprosate. This drug was highlighted due to it sharing targets with multiple diseases of interest, however these other diseases currently have no available treatments. Evidence in the literature supports the case for acamprosate, but investigations outside the scope of this thesis would need to be conducted to confirm this as a possibly viable drug for epilepsies. The inclusion of this database in the GEL research environment will fulfil the aim of this chapter to facilitate the discovery of such drug-disease associations. The form it will take is still in consideration, as database forms such as graph databases allow for more inference to be drawn from the data in question, but at the cost of requiring queries to be conducted in less familiar query languages. We will also focus on ensuring that this database is machine-learning ready, so that more

advanced tools for association inference can be used. This will be achieved by expanding data linkage to other datasets, and ensuring completeness of data as much as is possible. We believe that this dataset is already a long way towards being machine-learning ready, but we will add alternate representations of ontology derived data, such as exhibited in the embedding of the Gene Ontology in chapter 3, to ensure that data are of a suitable type. The continuation of this work should result in more repurposing hypotheses to be formed, in addition to those already suggested. This could bring a fast and pragmatic way to help address the significant unmet need that is found in the search for new drug treatments for rare disease.

Manuscripts contributed to or authored during this PhD:

Karczewski et al. 2020, Nature - The mutational constraint spectrum quantified from variation in 141,456 humans

Minikel et al. 2020, Nature - Evaluating drug targets through human loss-of-function genetic variation

Rhodes et al. - Leveraging loss-of-function genetic variant data from population cohorts for drug target prioritization (manuscript in preparation)

# Appendix

## 3.2.4.3 Cleaned feature set

The following 316 features were provided for the machine learning algorithms. All data were either integer or numeric classes.

| | | | | | |
|---|---|---|---|---|---|
| tdl.pharos_Tbio | GO_embed_45 | GO_embed_150 | MouseEssential | GO_embed_97 | oe_syn |
| tdl.pharos_nan | GO_embed_46 | GO_embed_151 | gwasCatalog | GO_embed_98 | Adipose - Subcutaneous |
| tdl.pharos_Tdark | GO_embed_47 | GO_embed_152 | NonEssentialCulture | GO_embed_99 | Adipose - Visceral (Omentum) |
| tdl.pharos_Tchem | GO_embed_48 | GO_embed_153 | insider | GO_embed_100 | Adrenal Gland |
| tdl.pharos_Tclin | GO_embed_49 | GO_embed_154 | centrality_betweenness | GO_embed_101 | Artery - Aorta |
| chromosome_6 | GO_embed_50 | GO_embed_155 | centrality_degree | GO_embed_102 | Artery - Coronary |
| chromosome_7 | GO_embed_51 | GO_embed_156 | centrality_eigen | GO_embed_103 | Artery - Tibial |
| chromosome_13 | GO_embed_52 | GO_embed_157 | centrality_hub | GO_embed_104 | Bladder |
| chromosome_12 | GO_embed_53 | GO_embed_158 | centrality_integration | GO_embed_105 | Brain - Amygdala |
| chromosome_1 | GO_embed_54 | GO_embed_159 | GO_embed_1 | GO_embed_106 | Brain - Anterior cingulate cortex (BA24) |
| chromosome_19 | GO_embed_55 | GO_embed_160 | GO_embed_2 | GO_embed_107 | Brain - Caudate (basal ganglia) |
| chromosome_8 | GO_embed_56 | GO_embed_161 | GO_embed_3 | GO_embed_108 | Brain - Cerebellar Hemisphere |
| chromosome_20 | GO_embed_57 | GO_embed_162 | GO_embed_4 | GO_embed_109 | Brain - Cerebellum |
| chromosome_11 | GO_embed_58 | GO_embed_163 | GO_embed_5 | GO_embed_110 | Brain - Cortex |
| chromosome_4 | GO_embed_59 | GO_embed_164 | GO_embed_6 | GO_embed_111 | Brain - Frontal Cortex (BA9) |
| chromosome_21 | GO_embed_60 | GO_embed_165 | GO_embed_7 | GO_embed_112 | Brain - Hippocampus |
| chromosome_16 | GO_embed_61 | GO_embed_166 | GO_embed_8 | GO_embed_113 | Brain - Hypothalamus |
| chromosome_9 | GO_embed_62 | GO_embed_167 | GO_embed_9 | GO_embed_114 | Brain - Nucleus accumbens (basal ganglia) |
| chromosome_17 | GO_embed_63 | GO_embed_168 | GO_embed_10 | GO_embed_115 | Brain - Putamen (basal ganglia) |
| chromosome_3 | GO_embed_64 | GO_embed_169 | GO_embed_11 | GO_embed_116 | Brain - Spinal cord (cervical c-1) |
| chromosome_10 | GO_embed_65 | GO_embed_170 | GO_embed_12 | GO_embed_117 | Brain - Substantia nigra |
| chromosome_22 | GO_embed_66 | GO_embed_171 | GO_embed_13 | GO_embed_118 | Breast - Mammary Tissue |
| chromosome_2 | GO_embed_67 | GO_embed_172 | GO_embed_14 | GO_embed_119 | Cells - EBV-transformed |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | lymphocytes |
| chromosome_15 | GO_embed_68 | GO_embed_173 | GO_embed_15 | GO_embed_120 | Cells - Transformed fibroblasts |
| chromosome_14 | GO_embed_69 | GO_embed_174 | GO_embed_16 | GO_embed_121 | Cervix - Ectocervix |
| chromosome_5 | GO_embed_70 | GO_embed_175 | GO_embed_17 | GO_embed_122 | Cervix - Endocervix |
| chromosome_18 | GO_embed_71 | GO_embed_176 | GO_embed_18 | GO_embed_123 | Colon - Sigmoid |
| chromosome_X | GO_embed_72 | GO_embed_177 | GO_embed_19 | GO_embed_124 | Colon - Transverse |
| chromosome_Y | GO_embed_73 | GO_embed_178 | GO_embed_20 | GO_embed_125 | Esophagus - Gastroesophageal Junction |
| oe_lof_upper | GO_embed_74 | GO_embed_179 | GO_embed_21 | GO_embed_126 | Esophagus - Mucosa |
| classic_caf | GO_embed_75 | GO_embed_180 | GO_embed_22 | GO_embed_127 | Esophagus - Muscularis |
| classic_caf_afr | GO_embed_76 | GO_embed_181 | GO_embed_23 | GO_embed_128 | Fallopian Tube |
| classic_caf_amr | GO_embed_77 | GO_embed_182 | GO_embed_24 | GO_embed_129 | Heart - Atrial Appendage |
| classic_caf_asj | GO_embed_78 | GO_embed_183 | GO_embed_25 | GO_embed_130 | Heart - Left Ventricle |
| classic_caf_eas | GO_embed_79 | GO_embed_184 | GO_embed_26 | GO_embed_131 | Kidney - Cortex |
| classic_caf_fin | GO_embed_80 | GO_embed_185 | GO_embed_27 | GO_embed_132 | Liver |
| classic_caf_nfe | GO_embed_81 | GO_embed_186 | GO_embed_28 | GO_embed_133 | Lung |
| classic_caf_oth | GO_embed_82 | GO_embed_187 | GO_embed_29 | GO_embed_134 | Minor Salivary Gland |
| classic_caf_sas | GO_embed_83 | GO_embed_188 | GO_embed_30 | GO_embed_135 | Muscle - Skeletal |
| p_afr | GO_embed_84 | GO_embed_189 | GO_embed_31 | GO_embed_136 | Nerve - Tibial |
| p_amr | GO_embed_85 | GO_embed_190 | GO_embed_32 | GO_embed_137 | Ovary |
| p_asj | GO_embed_86 | GO_embed_191 | GO_embed_33 | GO_embed_138 | Pancreas |
| p_eas | GO_embed_87 | GO_embed_192 | GO_embed_34 | GO_embed_139 | Pituitary |
| p_fin | GO_embed_88 | GO_embed_193 | GO_embed_35 | GO_embed_140 | Prostate |
| p_nfe | GO_embed_89 | GO_embed_194 | GO_embed_36 | GO_embed_141 | Skin - Not Sun Exposed (Suprapubic) |
| p_oth | GO_embed_90 | GO_embed_195 | GO_embed_37 | GO_embed_142 | Skin - Sun Exposed (Lower leg) |
| p_sas | GO_embed_91 | GO_embed_196 | GO_embed_38 | GO_embed_143 | Small Intestine - Terminal Ileum |
| cds_length | GO_embed_92 | GO_embed_197 | GO_embed_39 | GO_embed_144 | Spleen |
| gene_length | GO_embed_93 | GO_embed_198 | GO_embed_40 | GO_embed_145 | Stomach |
| obs_hom_lof | GO_embed_94 | GO_embed_199 | GO_embed_41 | GO_embed_146 | Testis |
| obs_het_lof | GO_embed_95 | GO_embed_200 | GO_embed_42 | GO_embed_147 | Thyroid |
| least1Hom | GO_embed_96 | oe_mis | GO_embed_43 | GO_embed_148 | Uterus |
| | | | GO_embed_44 | GO_embed_149 | Vagina |
| | | | | | Whole Blood |

## 3.2.1 Dataset compilation script

```
#NOTE
# There is some redundancy in this script (e.g. multiple calls to
biomart),
# this is a choice to.
#1) reduce memory burden.
#2) keep things a little clearer in workflow
# Setup
-----------------------------------------------------------------
-
pkgs <- c('dplyr',
          'tidyr',
          'data.table',
          'magrittr',
          'stringr',
          'splitstackshape',
          'RMySQL',
          'HGNChelper',
          'STRINGdb',
          'gdata')
if (!require("pacman")) install.packages('pacman')
pacman::p_load(pkgs, character.only = T)
rm(pkgs)


setwd('/mnt/volume/GEL_DB/LOF_DB/Data/Dataset_compilation')
# Funcs
-----------------------------------------------------------------
-
## INSIDER functions ##
insider_parse <- function(x){
 #Further parsing of insider data
 tmp <- as.data.frame(str_split_fixed(x$V1, "_ppi_", 2)) %>%
```

```
   mutate(Source = gsub('SOURCE: ', '', x$V2))
 names(tmp) <- c('Protein1','Protein2','Source')
 tmp[c('Protein1','Protein2')] <-
lapply(tmp[,c('Protein1','Protein2')],
                                        as.character)
 return(tmp)
}
###




##########
# START
##########
dat <-
fread('/mnt/volume/GEL_DB/LOF_DB/Data/gnomAD/Constraint/full_lof_m
etrics_by_transcript_an_adj_by_gene.txt',
           stringsAsFactors = F, header = T) %>%
 as_tibble() %>%
 group_by(gene) %>%
 arrange(desc(oe_lof)) %>%
 distinct(gene,.keep_all = T) %>%
 ungroup()


# Let's get rid of unwanted cols
dat %<>% dplyr::select(gene, transcript, oe_lof,
               oe_lof_upper, oe_lof_upper_bin,
               chromosome, starts_with("classic"),
starts_with("p_"),
               cds_length, gene_length, obs_hom_lof, obs_het_lof)
%>%
 mutate(pheno = 'ND',
        least1Hom = ifelse(obs_hom_lof > 0, 1, 0),
        Source = 'gnomAD') %>%
 select(gene, transcript, pheno, chromosome, everything())
```

```
#Lets also change those as NAs to 0 for least1Hom instead of na
dat %<>% mutate(least1Hom = ifelse(is.na(least1Hom), 0,
least1Hom))

#Lets update the names of the genes. We know that gnomAD uses some
old symbols
hgnc <- dat$gene %>% unique()

#Using HGNChelper - this seems like the most sensible route.
appGen <- checkGeneSymbols(x = hgnc) %>%
 filter(Approved != 'TRUE') %>%
 filter(!is.na(Suggested.Symbol))
#Some return more than one gene symbol. I think for now just go
through and
#check these manually as there are not that many.
appMult <- appGen[grep('/', appGen$Suggested.Symbol),]
appMult$Suggested.Symbol[grep('CXXC11', appMult$x)] <- 'RTP5'
appMult$Suggested.Symbol[grep('CEA', appMult$x)] <- 'CEACAM5'
appMult$Suggested.Symbol[grep('AGPAT9', appMult$x)] <- 'GPAT3'
appMult$Suggested.Symbol[grep('C11orf48', appMult$x)] <- 'LBHD1'
appMult$Suggested.Symbol[grep('B3GNT1', appMult$x)] <- 'B4GAT1'
appMult$Suggested.Symbol[grep('STRA13', appMult$x)] <- 'CENPX'
appMult$Suggested.Symbol[grep('ATP6C', appMult$x)] <- 'ATP6V1C1'
appMult$Suggested.Symbol[grep('CSRP2BP', appMult$x)] <- 'KAT14'

#Filter out then add back into other sets
appGen %<>% filter(!(x %in% appMult$x)) %>%
 bind_rows(appMult) %>%
 select(-Approved)

rm(appMult)

#Update gnomAD symbols
```

```
dat %<>%
 left_join(appGen, by=c('gene' = 'x')) %>%
 mutate(gene = ifelse(!is.na(Suggested.Symbol), Suggested.Symbol,
gene)) %>%
 select(-Suggested.Symbol)
#Now gather all the other genes seen in the other datasets - add
data if they
#are new. There shouldn't be any missing from the gnomAD set.
However, the other
#datasets will how us if the vars are found in a homozygous state.




# ELGH
----------------------------------------------------------------
--
#Only genes found in homozygous state included in this set. ELGH
currently not
#included within gnomAD.
#NOTE - ELGH did use LOFTEE
elgh <-
read.table('ELGH/all_LoFs.gatk_PASS.FS_30.DP_0.GQ_20.AB_0.01.LoFs.
missingness_lt_0.genotype_counts.present_in_ELGH.n_transcripts_cor
rected.all_transcripts_printed.annotation_not_in_last_exon_and_pre
sent_in_all_transcripts.txt',
                  stringsAsFactors = F, sep = '\t',
                  comment.char = '', header = T) %>%
 as_tibble()




#Add revised phenotype column to data
elgh %<>% mutate(pheno = 'ND',
                 Source = 'ELGH')



#Let's check out the ones that don't overlap - these will likely
be due to
#synonyms. gnomAD uses some old gene symbols
```

```
hgnc <- elgh$SYMBOL[which(!elgh$SYMBOL %in% dat$gene)] %>%
 unique()


appGen <- checkGeneSymbols(x = hgnc) %>%
 filter(Approved != 'TRUE') %>%
 filter(!is.na(Suggested.Symbol)) %>%
 select(-Approved)


#Thankully none have multiple suggested.
#Update ELGH to match
elgh %<>%
 left_join(appGen, by=c('SYMBOL' = 'x')) %>%
 mutate(SYMBOL =
          ifelse(!is.na(Suggested.Symbol), Suggested.Symbol,
SYMBOL)) %>%
 select(-Suggested.Symbol)


#Now check again for lack of overlap
hgnc <- elgh$SYMBOL[which(!elgh$SYMBOL %in% dat$gene)] %>%
 unique()
#Going to leave these for now. Potentially try and cover these
later. I imagine
#some will be pseudogenes.


#All ELGH seen in a homozygous state - update main data to show
this, keep track
#of where the hom came from
dat %<>% mutate(least1Hom = ifelse(least1Hom != 1 &
                                     gene %in% elgh$SYMBOL, 2,
least1Hom))
dat %<>% mutate(Source = ifelse(least1Hom == 2, 'ELGH',Source),
 least1Hom = ifelse(least1Hom > 0, 1, 0))


rm(elgh, appGen, hgnc)
```

```
#Now we have recorded homs as found in ELGH in the main dataset



# Benign
-----------------------------------------------------------------
#Add those included in the Narasimhan 2016 paper. These are all
benign LoF
#as they are found in healthy adults.
#These data are included in the ELGH set - but we KNOW that the
ones from this
#subset are healthy hence why we are treating them separately
nar <- read.xls('aac8624_Data_S1.xlsx') %>%
 as_tibble() %>%
 mutate(Gene.Name = as.character(Gene.Name))


hgnc <- nar$Gene.Name[which(!nar$Gene.Name %in% dat$gene)] %>%
 unique()


appGen <- checkGeneSymbols(x = hgnc) %>%
 filter(Approved != 'TRUE') %>%
 filter(!is.na(Suggested.Symbol)) %>%
 select(-Approved)


#We have one with multiple options
appGen$Suggested.Symbol[grep('AGPAT9', appGen$x)] <- 'GPAT3'


nar %<>%
 left_join(appGen, by=c('Gene.Name' = 'x')) %>%
 mutate(Gene.Name =
          ifelse(!is.na(Suggested.Symbol), Suggested.Symbol,
Gene.Name)) %>%
 select(-Suggested.Symbol)
```

```
hgnc <- nar$Gene.Name[which(!nar$Gene.Name %in% dat$gene)] %>%
 unique() #only loss of three


#Update to include narasimhan - slight change in how it is done
here, we want
#to overwrite the previous set if found here.
dat %<>% mutate(least1Hom = ifelse(gene %in% nar$Gene.Name, 2,
least1Hom))
dat %<>% mutate(Source = ifelse(least1Hom == 2, 'BIB', Source),
                least1Hom = ifelse(least1Hom > 0, 1, 0),
                pheno = ifelse(Source == 'BIB', 'BENIGN', pheno))


rm(nar, appGen, hgnc)



#From Identification of a large set of rare complete human
knockouts, Sulem 2015
sulem <- read.csv('Sulem_supp_data.csv', header = T,
stringsAsFactors = F) %>%
 as_tibble()



#Subset to those where earliest death is 50
sulem %<>% filter(Earliest.death.among.homozygotes..years. > 50)


hgnc <- sulem$Gene[which(!sulem$Gene %in% dat$gene)] %>%
 unique()



appGen <- checkGeneSymbols(x = hgnc) %>%
 filter(Approved != 'TRUE') %>%
```

```
 filter(!is.na(Suggested.Symbol)) %>%
 select(-Approved)


#Drop AGPAT9 - this is covered by BIB
appGen %<>% filter(x == 'AGPAT9')


sulem %<>%
 left_join(appGen, by=c('Gene' = 'x')) %>%
 mutate(Gene.Name = ifelse(!is.na(Suggested.Symbol),
                           Suggested.Symbol, Gene)) %>%
 select(-Suggested.Symbol)


#Change genes in our data that are found in this set to benign
dat %<>% mutate(least1Hom = ifelse((gene %in% sulem$Gene & Source
== 'gnomAD'),
                                   2, least1Hom))
dat %<>% mutate(Source = ifelse(least1Hom == 2, 'Sulem', Source),
               pheno = ifelse(Source == 'Sulem', 'BENIGN', pheno),
               least1Hom = ifelse(least1Hom > 0, 1, 0))


rm(sulem)


# PROMIS cohort https://doi.org/10.1038/nature22034
# For this set we are accepting those genes that have don't have
an association
#to the 250 odd diseases tested, and where there are two or more
homs -
#using 2 instead of 1 (as in the nar) because we don't have links
to health
#records.
prom <- read.xls('nature22034-s2.xlsx') %>% as_tibble()


prom$Gene[prom$Gene==''] <- NA
```

```
prom$GWAS..Traits[prom$GWAS..Traits==''] <- NA
prom %<>% fill(Gene.Number, Gene, Gene..Homozygous.pLoF.Count,
               GWAS..Number.of.Traits, GWAS..Traits,
               .direction = 'down')
prom %<>%
 filter(Confident.pLoF. == 'Yes') %>%
 filter(GWAS..Number.of.Traits == 0) %>%
 filter(Gene..Homozygous.pLoF.Count > 1) %>%
 distinct(Gene, .keep_all = T)

#This gives us a total of 237 unique genes
#Check the symbols
hgnc <- prom$Gene[which(!prom$Gene %in% dat$gene)] %>%
 unique()



appGen <- checkGeneSymbols(x = hgnc) %>%
 filter(Approved != 'TRUE') %>%
 filter(!is.na(Suggested.Symbol)) %>%
 select(-Approved)
#Dropping AGPAT9 + CXXC11 as already covered, otherwise all fine
prom %<>% filter(!(Gene %in% c('AGPAT9','CXXC11')))


prom %<>%
 left_join(appGen, by=c('Gene' = 'x')) %>%
 mutate(Gene.Name = ifelse(!is.na(Suggested.Symbol),
                           Suggested.Symbol, Gene)) %>%
 select(-Suggested.Symbol)


dat %<>% mutate(least1Hom = ifelse((gene %in% prom$Gene & Source
== 'gnomAD'),
                                   2, least1Hom))
```

```
dat %<>% mutate(Source = ifelse(least1Hom == 2, 'PROMIS', Source),
                pheno = ifelse(Source == 'PROMIS', 'BENIGN',
pheno),
                least1Hom = ifelse(least1Hom > 0, 1, 0))
rm(appGen, hgnc)


#Next have a deeper look at the Lim paper - see what to do wiht
this




# Gene lists
-----------------------------------------------------------------
#Switching to using gene lists from Macarthur lab
# wget -L
https://github.com/macarthur-lab/gene_lists/raw/master/lists/mgi_e
ssential.tsv
# wget -L
https://github.com/macarthur-lab/gene_lists/raw/master/lists/gwasc
atalog.tsv
# wget -L
https://github.com/macarthur-lab/gene_lists/raw/master/lists/NEGv1
_subset_universe.tsv
# wget -L
https://github.com/macarthur-lab/gene_lists/raw/master/lists/CEGv2
_subset_universe.tsv
# wget -L
https://github.com/macarthur-lab/gene_lists/raw/master/lists/cling
en_level3_genes_2018_09_13.tsv
#last downloaded 13/03/2018
#Mouse essential genes
meg <- fread('Macarthur_genelists/mgi_essential.tsv', header = F)
%>%
 as_tibble()
appGen <- checkGeneSymbols(x = meg$V1) %>%
 filter(Approved != 'TRUE') %>%
 filter(!is.na(Suggested.Symbol)) %>%
 select(-Approved)
appGen$Suggested.Symbol[grep('CSRP2BP', appGen$x)] <- 'KAT14'
```

```
meg %<>%
 left_join(appGen, by=c('V1' = 'x')) %>%
 mutate(V1 = ifelse(!is.na(Suggested.Symbol),
                            Suggested.Symbol, V1)) %>%
 select(-Suggested.Symbol)

dat %<>% mutate(MouseEssential = ifelse(gene %in% meg$V1, 1, 0))

rm(appGen, meg)

#Gwas catalog - genes near GWAS peaks
gwas <- fread('Macarthur_genelists/gwascatalog.tsv', header = F)
%>%
 as_tibble()

appGen <- checkGeneSymbols(x = gwas$V1) %>%
 filter(Approved != 'TRUE') %>%
 filter(!is.na(Suggested.Symbol)) %>%
 select(-Approved)

rm(appGen, gwas)
dat %<>% mutate(gwasCatalog = ifelse(dat$gene %in% gwas$V1, 1, 0))

#Non-essential genes in culture (CRISPR/CAS9)
neg <- fread('Macarthur_genelists/NEGv1_subset_universe.tsv',
header = F) %>%
 as_tibble()

appGen <- checkGeneSymbols(x = neg$V1) %>%
 filter(Approved != 'TRUE') %>%
 filter(!is.na(Suggested.Symbol)) %>%
```

```
 select(-Approved)


dat %<>% mutate(NonEssentialCulture = ifelse(dat$gene %in% neg$V1,
1, 0))


rm(appGen, neg)


#Essential in culture (CRISPR/CAS9)
ceg <- fread('Macarthur_genelists/CEGv2_subset_universe.tsv',
header = F) %>%
 as_tibble()


appGen <- checkGeneSymbols(x = ceg$V1) %>%
 filter(Approved != 'TRUE') %>%
 filter(!is.na(Suggested.Symbol)) %>%
 select(-Approved)


dat %<>% mutate(EssentialCulture = ifelse(dat$gene %in% ceg$V1, 1,
0))


rm(ceg, appGen)


#Clingen Haploinsufficient genes
clin <-
fread('Macarthur_genelists/clingen_level3_genes_2018_09_13.tsv',
             header = F) %>%
 as_tibble()
appGen <- checkGeneSymbols(x = clin$V1) %>%
 filter(Approved != 'TRUE') %>%
 filter(!is.na(Suggested.Symbol)) %>%
 select(-Approved)


dat %<>% mutate(clingenHI = ifelse(gene %in% clin$V1, 1, 0))
```

```
rm(appGen, clin)


#Now change the phenotypes for the essential culture to
deleterious and
#clingen HI if not already listed as benign. Doing separately for
clarity.
dat %<>%
 mutate(pheno =
          ifelse((pheno != 'BENIGN' & clingenHI == 1),
'DELETERIOUS', pheno),
        pheno =
          ifelse((pheno != 'BENIGN' & EssentialCulture == 1),
                 'DELETERIOUS', pheno))




# Biomart
----------------------------------------------------------------------
library('biomaRt')
hgnc <- dat$gene %>% unique()
ensembl <- useMart("ensembl",dataset="hsapiens_gene_ensembl")
res.go <- getBM(attributes = c('hgnc_symbol' ,'go_id','name_1006',
                               'namespace_1003' ),
                filters = 'hgnc_symbol',
                values = hgnc,
                mart = ensembl) %>%
 as_tibble()


#Switching up how I store these. Will now just have all GO terms
stored
res.mf <- res.go[res.go$namespace_1003 == 'molecular_function',]
%>%
 group_by(hgnc_symbol) %>%
 summarize(MF_GO_ID = paste(go_id, collapse = '|'),
           MF_GO = paste(name_1006, collapse = '|'))
```

```
res.cc <- res.go[res.go$namespace_1003 == 'cellular_component',]
%>%
 group_by(hgnc_symbol) %>%
 summarize(CC_GO_ID = paste(go_id, collapse = '|'),
           CC_GO = paste(name_1006, collapse = '|'))


res.bp <- res.go[res.go$namespace_1003 == 'biological_process',]
%>%
 group_by(hgnc_symbol) %>%
 summarize(BP_GO_ID = paste(go_id, collapse = '|'),
           BP_GO = paste(name_1006, collapse = '|'))


dat %<>% left_join(res.mf, by=c('gene' = 'hgnc_symbol')) %>%
 left_join(res.bp, by=c('gene' = 'hgnc_symbol')) %>%
 left_join(res.cc, by=c('gene' = 'hgnc_symbol'))


res.mf %<>%
 group_by(hgnc_symbol) %>%
 summarize(MF_GO_ID = paste(go_id, collapse = '|'),
           MF_GO = paste(name_1006, collapse = '|'))


rm(list=ls(pattern = 'res.*'))


#OMIM genes
res.mim <- getBM(attributes = c('hgnc_symbol',
                                "mim_gene_accession"),
                 filters = 'hgnc_symbol',
                 values = hgnc,
                 mart = ensembl) %>%
 as_tibble()
```

```
res.mim %<>% mutate(OMIM = ifelse(!is.na(mim_gene_accession),
1,0)) %>%
 arrange(desc(OMIM)) %>%
 distinct(hgnc_symbol, .keep_all = T)


dat %<>% left_join(res.mim, by = c('gene'='hgnc_symbol'))




protein <- getBM(attributes=c("hgnc_symbol", "uniprotswissprot"),
                 filters='hgnc_symbol', mart = ensembl, values =
hgnc) %>%
 as_tibble()

protein %<>% arrange(desc(uniprotswissprot)) %>%
                 distinct(hgnc_symbol, .keep_all = T)


dat %<>% left_join(protein, by=c('gene'='hgnc_symbol'))


rm(protein, res.mim, ensembl, hgnc)


detach('package:biomaRt', unload = T)


#Any that are omim and not benign = DELETERIOUS
#This is introducing NAs - these should really just become NDs too
dat %<>%
 mutate(pheno = ifelse((pheno != 'BENIGN' & OMIM == 1),
'DELETERIOUS',pheno))


#I'm not sure I'm happy about this step. It basically makes a
tonne of them
#deleterious that I'm not sure SHOULD be
```

```
# INSIDER
------------------------------------------------------------------
ins <- fread('Insider/07-02-19_interactions.txt', sep = ',',
header = F) %>%
 insider_parse() %>%
 as_tibble()



#Reset index
rownames(ins) <- NULL
#If there is more than one source, keep the best source
# CoCystal > Homology > Predicted
ins %<>%
 mutate(Source =  factor(Source, levels = c('CoCrystal Structure',
                                            'Homology Model',
                                            'Predicted
Interface'))) %>%
 arrange(Source) %>%
 distinct(Protein1,Protein2,.keep_all = T)


#This data can probably only be added to the protein model. Think
about this.
dat %<>% mutate(insider =
                ifelse(uniprotswissprot %in%
                         unique(c(ins$Protein1, ins$Protein2)),
1, 0))


#Keep ins for the stringDB stuff later.


# Add Pharos
------------------------------------------------------------------


#Remote connection to pharosDB, then pull relevant features
pharos.ids <- dat$uniprotswissprot %>% unique
```

```
pharos.ids <- paste0("'", pharos.ids, "'")
pharos.db = dbConnect(dbDriver("MySQL"), user='tcrd',
dbname='tcrd520',
                      host='tcrd.kmc.io')
query <-
 paste0('SELECT id, uniprot, family, dtoid FROM protein WHERE
uniprot IN (',
               paste0(pharos.ids, collapse = ', '), ');')
rm(pharos.ids)


ids <- dbGetQuery(pharos.db, query)
ids$dtoid <- paste0("'", ids$dtoid, "'")
query2 <- paste0('SELECT id, name, ttype, tdl, fam FROM target
WHERE id IN (',
                paste0(ids$id, collapse = ', '), ');')
query3 <- paste0('SELECT * FROM dto WHERE id IN (',
                paste0(ids$dtoid[!is.na(ids$dtoid)], collapse = ',
'), ');')
targs <- dbGetQuery(pharos.db, query2)
dto <- dbGetQuery(pharos.db, query3)


names(targs)[2] <- 'protein_name'
names(dto)[2] <- 'dto_name'


#Need to drop "'" from data
ids$dtoid <- gsub("\'",'',ids$dtoid)
pharos.dat <- left_join(ids, targs, by='id') %>%
 left_join(dto, by=c('dtoid'='id')) %>%
 dplyr::select(-id)
names(pharos.dat) <- paste0(names(pharos.dat), '.pharos')


#Add to main data/cleanup
dbDisconnect(pharos.db)
dat %<>%
```

```
 left_join(pharos.dat, by=c('uniprotswissprot'='uniprot.pharos'))


rm(pharos.dat, query, query2, query3, ids, dto, pharos.db, targs)


# Consistency check
------------------------------------------------------
#Group and arrange, then replace with the first one (this works
because alphabetical
# order gives me the correct order of phenotypes)
dat %<>%
 group_by(gene) %>%
 arrange(pheno) %>%
 ungroup() %>%
 distinct(gene, .keep_all = T)


#This occurs because a small number of genes are represented twice
in the
#constraint doc




# Write to file
------------------------------------------------------------
filename <- paste0('Clean/', Sys.Date(),'_LOFdb.txt')
write.table(dat, filename, sep='\t', row.names = F)
filename <- paste0('Clean/', Sys.Date(),'_insider_edges.txt')
write.table(ins, filename, sep = '\t', row.names = F)


# StringDB links
-----------------------------------------------------------
#Reworking to just use the stringDB package - this is now good
string_db <- STRINGdb$new( version="10",
                           species=9606,
                           score_threshold=0,
```

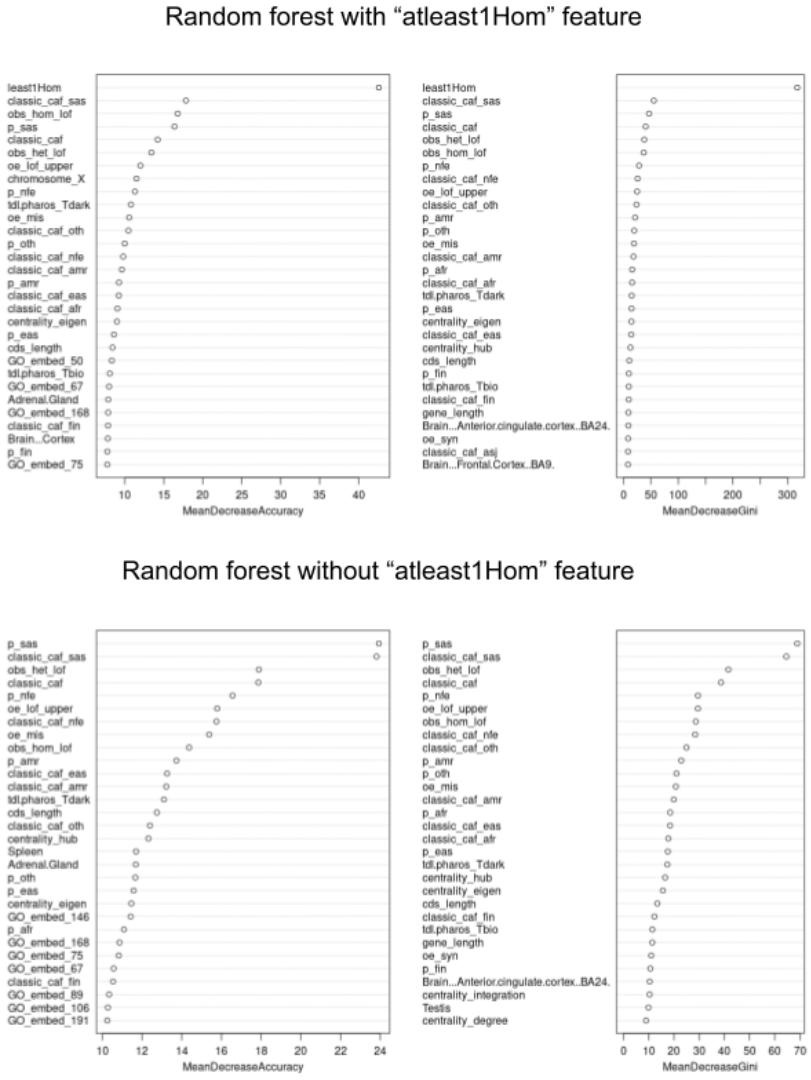
```
input_directory="/mnt/volume/GEL_DB/LOF_DB/Data/Dataset_compilatio
n/StringDB" )



#This just adds a STRING_id column

dat %<>% as.data.frame %>% #function doesn't work with tibbles

 string_db$map('gene', removeUnmappedRows = F)


#Now just add gene names

ints <- string_db$get_interactions(dat$STRING_id)

dim(ints)

ints %<>%

 as_tibble %>%

 left_join(dplyr::select(dat, gene, STRING_id),
by=c('from'='STRING_id')) %>%
 rename('gene1'='gene') %>%

 left_join(dplyr::select(dat, gene, STRING_id),
by=c('to'='STRING_id')) %>%
 rename('gene2'='gene')


#Write these to file

filename <- paste0('../Network/',
Sys.Date(),'_StringDB_links.txt')
write.table(ints, filename, sep = '\t', row.names = F)

##########

# END

##########
```

# 3.3.9 Table 1

| | Benign | Deleterious | Class.error | With “atleast1Hom” | Out of bag error (%) |
|---|---|---|---|---|---|
| Benign | 454 | 850 | 0.65 | Y | 9.37 |
| Deleterious | 157 | 9284 | 0.017 | Y | |
| Benign | 352 | 952 | 0.73 | N | 10.23 |
| Deleterious | 147 | 9249 | 0.016 | N | |

Appendix Table 1 - A confusion matrix showing the class assignments based on two random forest models, one with and one without the feature “atleast1Hom”. The class error shows the proportion of genes mislabelled from the training data. The out of bag error provides an estimate of the prediction error of the model. By this measure, the model omitting the “atleast1Hom” feature performs worse than the model including it. Removal of the “atleast1Hom” feature results in reduced performance of the model, with positive benign labels being more likely to be classified as deleterious.

# 3.3.9 Feature importance estimation




Appendix Figure 3.3.9 - Dotcharts showing the feature importance as measured by a Random Forest model. The left hand plots show the mean decrease in the number of observations that are correctly classified (accuracy) upon removal of a feature, the right hand plot show the mean decrease in Gini importance index  A) Data for the random forest model including the “atleast1Hom feature”. B) Data for the random forest model omitting the “atleast1Hom” feature.

## 5.2.1 Database build scripts

### Db_update.sh

```
#!/bin/bash
#title          :db_update.sh
#description    :Call R scripts to run updates on the GEL DB
#author         :Dan Rhodes
#date           :20180321
#version        :v0.1
#usage          :./db_update.sh
#notes          :
#bash_version   :4.3.11(1)-release
#==============================================================================
=========

read -p "Enter Username: " usr
read -s -p "Enter Password: " pwrd


RESULT=`mysqlshow --user=$usr --password=$pwrd GEL| grep -v Wildcard
| grep -o GEL` #2>&1 >/dev/null



[ "$RESULT" == "GEL" ] && echo "Database Found" || exit "GEL
database not found"

read -p "Enter path for python env (if applicable), e.g.
./envs/bin/activate: " pyth
read -p "Enter path with filename to clinical phenotypes: `echo
$'\n> '`" PhenPath

[ -z "$PhenPath" ] && { echo "Need to set filepath"; exit 1; }
[ ! -e "$PhenPath" ] && { echo "File doesn't exist"; exit 1; }

#Order of scripts
s1='Clinical_phenotypes_add.R'
```

```
s2='PanelApp_prep.R'
s3='dis_pheno_mapping.R'
s4='DGIdb_api.R'
#s5='SMILE.R' Replaced with new set
s5='drugbank_parse.R'
s6='drug_pheno_disease_mappings.R'

echo "Suspending GEL database foreign keys"

mysql --user=$usr --password=$pwrd GEL -Bse "SET foreign_key_checks
= 0;"

echo "Cutting rows containing 'Genomic medicine service
indications'"
sed -i '/Genomic medicine service indications/d' $PhenPath

echo "#######################################################"
echo "Adding GEL disease data"
# Args to script - filepath, db password
echo "Running $s1"
Rscript $s1 $PhenPath $pwrd

echo "#######################################################"
echo "Adding panelapp gene data"
echo "Running $s2"
Rscript $s2 $PhenPath $pwrd

echo "#######################################################"
echo "Creating disease to phenotype mapping tables"
echo "Running $s3"
Rscript $s3 $PhenPath $pwrd

echo "#######################################################"
[[ ! -z "${pyth// }" ]] && { echo "Starting pyenv"; source $pyth; }

echo "Grabbing gene-drug data from DGIdb"
echo "Running $s4"
Rscript $s4 $PhenPath $pwrd

echo "#######################################################"
echo "Adding SMILE drug data"
echo "Running $s5"
Rscript $s5 $PhenPath $pwrd

echo "#######################################################"
echo "Adding mapping tables data"
```

```
echo "Running $s6"
Rscript $s6 $PhenPath $pwrd

echo "#######################################################"
echo "Re-engaging foreign key constraints on GEL database"
mysql --user=$usr --password=$pwrd GEL -Bse "SET foreign_key_checks
= 1;"


echo "Complete"
```

## Clinical_phenotypes_add.R

```
#!/usr/bin/env r

#########################################################################
#
## Project: GEL_DB
## Script purpose: Add or update data from clinical phenotypes list
## Date: 2018-03-22
## Author: Dan Rhodes
#########################################################################
#

# Libs
-------------------------------------------------------------------
pkgs <- c('dplyr',
          'RMySQL',
          'magrittr',
          'tidyr',
          'data.table')

if (!require("pacman")) install.packages('pacman')
pacman::p_load(pkgs, character.only = T)

rm(pkgs)

# Functions
-----------------------------------------------------------------
cleanNA <- function(x){
  #Function to remove all blanks and NAs
  x[x==''] <- NA
  x <- na.omit(x)
  return(x)
}

check_data <- function(gel.tables){
  #Check existing data and only keep new data
  query.list <- list()
  existing.list <- list()
  for(x in names(gel.tables)){
    query.list[[x]] <- paste0('select * from ', x)
    existing.list[[x]] <- dbGetQuery(con1, query.list[[x]])
    if(x == 'disease'){
      existing.list[[x]] %<>%
select(names(gel.tables[[x]])[names(gel.tables[[x]]) !=
'last_updated'])
```

```
    } else {
      existing.list[[x]] %<>% select(names(gel.tables[[x]]))
    }
    #Convert all data types to char for joining
    existing.list[[x]] %<>% mutate_all(as.character)
    gelchar <- gel.tables[[x]] %>% mutate_all(as.character)
    tmp <- anti_join(gelchar, existing.list[[x]])
    gel.tables[[x]] <- tmp
  }
  return(gel.tables)
}

# Data import
------------------------------------------------------------
BashArgs = commandArgs(trailingOnly=TRUE)


if(Sys.info()["nodename"] == 'dan-XPS-13-9350'){

source('/home/dan/Documents/Data/Visible/QMUL/GeL/Disease_ontology/G
EL_DB/DB_setup/Scripts/db_connect.R')
} else {
  source('/mnt/volume/GEL_DB/DB_setup/Scripts/db_connect.R')
}


inter <- ifelse(length(BashArgs) > 0, 0, 1)
if(inter == 1){
  if(Sys.info()["nodename"] == 'dan-XPS-13-9350'){
    gel.dat <-
read.csv('/home/dan/Documents/Data/Visible/QMUL/GeL/Disease_ontology
/GEL_DB/DB_setup/Clinical_phenotypes/v1.9.0/v1.9.0.csv',
                         header = T, stringsAsFactors = F, row.names
= NULL)
  } else {
    gel.dat <-
read.csv('/mnt/volume/GEL_DB/DB_setup/Clinical_phenotypes/v1.9.0/v1.
9.0.csv',
                         header = T, stringsAsFactors = F, row.names
= NULL)
  }
} else{
  gel.dat <- read.csv(BashArgs[1], header = T, stringsAsFactors = F,
row.names = NULL)
}
```

```
con1 <- start_con(BashArgs)

db.tabs <- dbListTables(con1)

# Clean
------------------------------------------------------------------------
gelnames <- names(gel.dat)
gelnames <- gelnames[2:length(gelnames)]
gel.dat$Test.ID <- NULL
names(gel.dat) <- gelnames
rm(gelnames)


# Create input tables
------------------------------------------------------
gel.tables <- list()

##Disease group
tmp <- dbListFields(con1, 'disease_group')
gel.tables[['disease_group']] <-
unique(gel.dat[c('id','Level.2.Disease.Group')])
names(gel.tables[['disease_group']]) <- tmp

##Disease subgroup
tmp <- dbListFields(con1, 'disease_subgroup')
gel.tables[['disease_subgroup']] <-
unique(gel.dat[c('id.1','id','Level.3.Disease.Subgroup')])
names(gel.tables[['disease_subgroup']]) <- tmp

##Specific disease
tmp <- dbListFields(con1, 'disease')
gel.tables[['disease']] <-
unique(gel.dat[c('id.2','id.1','Level.4.Specific.Disorder')])
names(gel.tables[['disease']]) <- tmp[!tmp %in%
c('panelapp_v','last_updated')] # panelapp data added in later
script
gel.tables$disease$disease <-
Hmisc::capitalize(tolower(gel.tables$disease$disease))


##Phenotype
tmp <- dbListFields(con1, 'phenotype')
gel.tables[['phenotype']] <-
unique(gel.dat[c('Phenotype.ID','Phenotype')])
names(gel.tables[['phenotype']]) <- tmp
```

```
##Test
tmp <- dbListFields(con1, 'test')
gel.tables[['test']] <- unique(gel.dat[c('Test.ID','Test')])
names(gel.tables[['test']]) <- tmp

gel.tables <- lapply(gel.tables, cleanNA)
gel.tables <- lapply(gel.tables,`rownames<-` , NULL ) #reset index

# Add to DB
------------------------------------------------------------------
to_upload <- check_data(gel.tables)

#Upload
for(n in names(to_upload)){
  dbWriteTable(conn = con1, name = n, value = to_upload[[n]],
row.names=FALSE, append=T, overwrite = F)
  cat(dim(to_upload[[n]])[1], 'records updated in table ', n, '\n')
}

rm(to_upload, gel.tables)

# Read data back
-----------------------------------------------------------
db.list <- list()
db.list[['gene']] <- dbGetQuery(con1, 'select * from gene')
db.list[['disease']] <- dbGetQuery(con1, 'select * from disease')
db.list[['pheno']] <- dbGetQuery(con1, 'select * from phenotype')
db.list[['gene_dis']] <- dbGetQuery(con1, 'select * from
gene_to_disease')
db.list[['test']] <- dbGetQuery(con1, 'select * from test')
db.list[['dis_to_pheno']] <-dbGetQuery(con1, 'select * from
dis_to_pheno')

# Test to disease mapping
-------------------------------------------------
td <- gel.dat[,c('Level.4.Specific.Disorder','Test', 'Test.ID')] %>%
unique
td$Test[td$Test== ''] <- NA
td %<>% drop_na(-Test.ID)
td %<>% left_join(db.list$disease,
by=c("Level.4.Specific.Disorder"="disease"))
td[,c('id_dis_sub','Test','Level.4.Specific.Disorder')] <- NULL
td$id_test_to_disease <- NA
td$id_test_to_disease <- paste(td$id_disease,td$Test.ID, sep='_')
names(td)[grep('test.id',names(td),ignore.case = T)] <- 'id_test'
td <- td[,c(dbListFields(con1,'test_to_disease'))]
```

```
td$id_test_to_disease <- paste(td$id_disease, td$id_test, sep ='_')

# Disease to phenotype mapping
-----------------------------------------
dp <- gel.dat[,c('id.2','Phenotype.ID')] %>% unique
dp$id.2[dp$id.2 == ''] <- NA
dp$Phenotype.ID[dp$Phenotype.ID== ''] <- NA
dp %<>% drop_na()
names(dp) <- c('id_disease','id_phenotype')
dp$id_dis_pheno <- paste(dp$id_phenotype, dp$id_disease, sep='.')
dp <- dp[,c(dbListFields(con1,'dis_to_pheno'))]
maps <- list(test_to_disease = td, dis_to_pheno = dp)

to_upload <- check_data(maps)

#Upload
for(n in names(to_upload)){
  dbWriteTable(conn = con1, name = n, value = to_upload[[n]],
row.names=FALSE, append=T, overwrite = F)
  cat(dim(to_upload[[n]])[1], 'records updated in table ', n, '\n')
}

#Disconnect from DB
dbDisconnect(con1)

quit()

### END ###
```

## PanelApp_prep.R

```
##############################################
# Take Panelapp data harmonise with GEL DB  #
##############################################

# Take panelapp data and add to mysql database, adds to tables
disease, gene and gene_to_disease

# Libs
----------------------------------------------------------------------
pkgs <- c('RMySQL',
          'httr',
          'tidyjson',
          'jsonlite',
          'dplyr',
          'tidyr',
          'stringr',
          'Hmisc',
          'magrittr',
          'magrittr'
          #'biomaRt', this is loaded later due to conflicts in
commands with tidyr
          )

if (!require("pacman")) install.packages('pacman')
pacman::p_load(pkgs, character.only = T)

rm(pkgs)

# Functions
----------------------------------------------------------------

clean_json <- function(x){
  #Prep disease data for entry to main database
  #Take a query response, parse JSON and extract relevant info,
returns df
  pan <- httr::content(x, 'text', encoding = 'UTF-8') %>%
as.tbl_json
  #Check genes are present
  if(length(attr(pan,'JSON')[[1]][[1]]$Genes) < 1){
    disease <- pan %>%
      enter_object("result") %>%
      spread_values(jstring("SpecificDiseaseName") ) %>%
      as.data.frame
    print(paste(disease[,2] , "has no genes in panel", sep= ' '))
```

```
  } else{
    #Temporary names for genes
    gene_tmp <-
paste("Gene",c(seq(length(attr(pan,'JSON')[[1]][[1]]$Genes))),sep=''
)
    names(attr(pan,"JSON")[[1]][[1]]$Genes) <- gene_tmp
    genes <- pan %>%
      enter_object("result") %>%
      enter_object("Genes")
    genekeys <- gather_keys(genes)$key
    #Pull gene info from each object
    genedf <- genepull_json(genekeys = genekeys, genes = genes)
    #Pull all other info
    otherdf <- pan %>%
      enter_object("result") %>%
      spread_values(
        Dis = jstring("SpecificDiseaseName"),
        Dis_sub = jstring("DiseaseSubGroup"),
        NV = jstring("version"),
        Dis_group = jstring("DiseaseGroup") ) %>%
      mutate( NV = as.numeric(NV)) %>%
      tbl_df
    total <- cbind(otherdf,genedf)
    return(total)
  }
}

genepull_json <- function(genekeys, genes){
  #Take keys from json and genes object, pull data from each gene
object and return as df
  outdf <- data.frame(LevelOfConfidence = NA,
                      Penetrance = NA,
                      MoI = NA,
                      Gene = NA,
                      Ensembl = NA,
                      Pheno = NA,
                      MoP = NA)
  for(x in genekeys){
    tmp <- genes %>% enter_object(x) %>%
      spread_values(
        Gene = jstring("GeneSymbol"),
        Ensembl = jstring("EnsembleGeneIds"),
        MoI = jstring("ModeOfInheritance"),
        MoP = jstring("ModeOfPathogenicity"),
        Penetrance = jstring("Penetrance"),
        Pheno = jstring("Phenotypes"),
```

```
        LevelOfConfidence = jstring("LevelOfConfidence") ) %>%
      tbl_df
    outdf <- bind_rows(outdf,tmp) %>% tbl_df
  }
  #Remove NA rows
  idx <- apply(outdf, 1, function(x) all(is.na(x)))
  outdf <- outdf[ !idx, ]
  return(outdf)
}


# MySQL db
------------------------------------------------------------------
BashArgs = commandArgs(trailingOnly=TRUE)

if(Sys.info()["nodename"] == 'dan-XPS-13-9350'){

source('/home/dan/Documents/Data/Visible/QMUL/GeL/Disease_ontology/G
EL_DB/DB_setup/Scripts/db_connect.R')
} else {
  source('/mnt/volume/GEL_DB/DB_setup/Scripts/db_connect.R')
}

con1 <- start_con(BashArgs)

db.tabs <- dbListTables(con1)

# Pull disease and gene information from mysql db
pull_data <- function(tablist, con){
  db.list <- list()
  for(tab in tablist){
    query <- paste0('SELECT * FROM ', tab, ';')
    db.list[[tab]] <- dbGetQuery(con, query)
  }
  return(db.list)
}

db.list <- pull_data(db.tabs, con1)

# Panelapp info
-------------------------------------------------------------
#Pull version numbers for disease from panelapp
# All panel app info
panelapp <-
GET('https://panelapp.genomicsengland.co.uk/WebServices/list_panels/
')
```

```
if(panelapp$status_code != 200){
  errm <- paste0("Panelapp API response code: ",
panelapp$status_code)
  stop(errm)
}

# Parse and subset json, output as df
pa.json <- httr::content(panelapp,'text')
pa <- pa.json %>% as.tbl_json
pa.sub <- pa %>%
  enter_object("result") %>%
  gather_array %>%
  spread_values(
    panel_id = jstring("Panel_Id"),
    Disease = jstring("Name"),
    CV = jstring("CurrentVersion"),
    Dis_sub = jstring("DiseaseSubGroup"),
    Dis_group = jstring("DiseaseGroup") ) %>%
  mutate( CV = as.numeric(CV) ) %>%
 tbl_df()


#Create table showing those which don't need updating
#Init output table
update.df <- as.data.frame(matrix(ncol = ncol(pa.sub))) %>%
  as_tibble()
names(update.df) <- names(pa.sub)

#Find those where version in db matches version in panelapp
for(i in 1:nrow(pa.sub)){
  x <- db.list$disease[i, c('disease','panelapp_v')]
  pa.x <- pa.sub[grep(tolower(x$disease), tolower(pa.sub$Disease),
fixed = T),]
  if(nrow(pa.x) > 0){
    if((x$panelapp_v == pa.x$CV)|(is.na(x$panelapp_v))){
      update.df[i,] <- pa.x
      pa.x <- NULL
      x <- NULL
    }
  }
}
#Drop these
update.df <- update.df[rowSums(is.na(update.df)) != ncol(update.df),
]
pa.sub %<>% filter(!panel_id %in% update.df$panel_id)
```

```
#Cleanup
rm(i,x,pa.x)

# Panelapp data pull
-------------------------------------------------------
#Loop through diseases, GET json, only taking high confidence genes
panels.ls <- c()
for(x in pa.sub$panel_id){
  y <- URLencode(x)
  com <-
paste('https://panelapp.genomicsengland.co.uk/WebServices/get_panel/
',y,'/?LevelOfConfidence=HighEvidence', sep = '')
  panels.ls[[x]] <- GET(com)
}

# Clean json
-----------------------------------------------------------
tidy_panels.ls <- lapply(panels.ls, clean_json)

#Drop panels with no genes
tonull <- c()
for(i in 1:length(tidy_panels.ls)){
  if(grepl('has no genes in panel$', tidy_panels.ls[[i]][1]) ==
TRUE){
    tonull <- c(tonull,i)
  }
}
tidy_panels.ls[tonull] <- NULL

#Bind into one df
panels.df <- as.data.frame(matrix(ncol = 13)) %>% tbl_df
names(panels.df) <- names(tidy_panels.ls[[1]])
for(x in tidy_panels.ls){
  if(typeof(x)=='list'){
  panels.df <- bind_rows(panels.df, x)
  }
}
idx <- apply(panels.df, 1, function(x) all(is.na(x)))
panels.df <- panels.df[!idx, ]

#Remove any punctuation from Gene names
panels.df$Gene <- sub('[[:punct:]]','',panels.df$Gene)

#Capitalise each word for diseases
panels.df$Dis <- Hmisc::capitalize(tolower(panels.df$Dis))
```

```
#Some don't have any group or subgroup disease information. This
will break
#our data hierarchy, so drop them
panels.df %<>% filter(Dis_group != '' & Dis_sub != '')


# Integrate data
-------------------------------------------------------------

## GENES ##
#Now need to separate all cases containing multiple phenotypes for
phenotype mapping
## Ensembl IDs
#Panelapp uses Ensembl IDs so sticking to these, ignore entrez for
now for simplicity
ensembl.df <- panels.df[,c('Dis','Gene','Ensembl')] %>% as_tibble()
ensembl.df[,'Ensembl'] <-
  apply(ensembl.df[,'Ensembl'], 1, function(x)
str_extract(x,'ENSG[[:digit:]]*'))

#Limit to new genes for gene table, limit to new disease/gene
combination for disease/gene mapping
tmp_gene <- ensembl.df[,c('Ensembl','Gene')]
tmp_gene <- tmp_gene[!(toupper(tmp_gene$Gene) %in%
toupper(db.list$gene$hgnc_symbol)), ]
names(tmp_gene) <- names(db.list$gene)[2:3]
if(nrow(tmp_gene) > 0){
  tmp_gene$id_gene <- NA
  tmp_gene <- tmp_gene[,names(db.list$gene)]
  tmp_gene <- unique(tmp_gene)
  #Add unique key
  if(nrow(db.list$gene) > 0){
    start.idx <- max(db.list$gene$id_gene)
  } else {
    start.idx <- 0
  }
  tmp_gene$id_gene <- seq(start.idx + 1, (start.idx + 1) +
nrow(tmp_gene)-1)

  #Add to gene.db
  dbWriteTable(conn = con1, name = "gene", value = tmp_gene,
row.names=FALSE, append=TRUE, overwrite = F)
} else {
  print('No new genes')
}
```

```
rm(x, tmp_gene, update.df, tidy_panels.ls, panels.ls, i, idx,
db.tabs, com,
   y, tonull, ensembl.df, pa, panelapp)
## GENES END ##

## DISEASE ##
#Add any new diseases not in clin phen
toadd <- panels.df %>%
  filter(!tolower(Dis) %in% tolower(db.list$disease$disease)) %>%
  select(Dis, Dis_sub, Dis_group, NV) %>%
  distinct() %>%
  mutate(id_disease = NA,
         id_dis_group = NA,
         id_dis_sub= NA)

#Prep the tables
#Disease tab
distab <- toadd %>%
  select('disease'='Dis') %>%
  filter(!disease == '') %>%
  mutate(id_disease = NA)

#Disease_Subgroup tab
subtab <- toadd %>%
  select('disease_subgroup' = 'Dis_sub') %>%
  distinct() %>%
  filter(!tolower(disease_subgroup)
         %in% tolower(db.list$disease_subgroup$disease_subgroup))
%>%
  filter(!disease_subgroup == '') %>%
  mutate(id_dis_sub = NA)

#We know there is a special case with haematological disorders
#edit this manually
toadd %<>% mutate(Dis_group =
                    ifelse(Dis_group == 'Haematological disorders',
                           'Haematological and immunological
disorders',
                           Dis_group))

# Group tab
grptab <- toadd %>%
  select('disease_group'='Dis_group') %>%
  distinct() %>%
  filter(!disease_group %in% db.list$disease_group$disease_group)
```

```
%>%
  filter(!disease_group == '') %>%
  mutate(id_dis_group = NA) %>%
  select(id_dis_group, disease_group)



#Now have to deal with the lack of appropriate IDs for the new sets
#Add a prefix and start new numbering convention
idx_start <- function(x){
  #Given column in specific table, which is most recent identifier
  #Just testing to see if previous PA vs added, NOT GENERAL USE
  if(length(grep('^PA',x))> 0){
    existgrp <- grep('^PA',x, value = T)
    existgrp <- gsub('PA','',existgrp)
    existgrp <- as.numeric(existgrp[which.max(existgrp)])
    existgrp <- sprintf("99%03d", existgrp + 1)
  } else {
    existgrp <- 1
    existgrp <- sprintf("99%03d", existgrp)
  }
  return(existgrp)
}

add_ids <- function(table, start_id){
  #Add IDs to whichever column is necessary
  n <- as.numeric(gsub('^99','',start_id))
  newids <- sprintf("99%03d",
                    seq(from = n,
                        to = nrow(table)))
  return(newids)
}

#Now we need to fill in the missing data that is already found in
the DB
#So first we need the mapping for group to subgroup.
grpadd <- toadd %>%
  filter(is.na(id_dis_group)) %>%
  left_join(db.list$disease_group, by =
c('Dis_group'='disease_group')) %>%
  mutate(id_dis_group = coalesce(as.character(id_dis_group.x),
                                 as.character(id_dis_group.y))) %>%
  select(-id_dis_group.x,-id_dis_group.y)

sbgrpadd <- toadd %>%
  filter(is.na(id_dis_sub)) %>%
```

```
  left_join(select(db.list$disease_subgroup,
                   disease_subgroup, id_dis_sub),
            by=c('Dis_sub'='disease_subgroup')) %>%
  mutate(id_dis_sub = coalesce(as.character(id_dis_sub.x),
                               as.character(id_dis_sub.y))) %>%
  select(-id_dis_sub.x,-id_dis_sub.y)

#None of the diseases will need adding

#Add these to the toadd tab
sbgrpadd %<>%
  select(-id_dis_group) %>%
  left_join(select(grpadd,
                   Dis_group,
                   id_dis_group), by ='Dis_group') %>%
  distinct()

rm(grpadd)

toadd %<>%
  select(-id_dis_group, -id_dis_sub) %>%
  left_join(select(sbgrpadd,
                   Dis,
                   id_dis_sub,
                   id_dis_group), by =c('Dis'))
rm(sbgrpadd)


#now actually replace those of the NAs

toadd$id_dis_sub[is.na(toadd$id_dis_sub)] <-
  idx_start(db.list$disease_group$id_dis_sub) %>%
  add_ids(table = filter(toadd, is.na(id_dis_sub)))

toadd$id_disease[is.na(toadd$id_disease)] <-
  idx_start(db.list$disease$id_disease) %>%
  add_ids(table = filter(toadd, is.na(id_disease)))

#Now redefine the tables that need to be added with their IDs
#Disease tab
distab <- toadd %>%
  select('disease'='Dis',
         'panelapp_v'='NV',
         id_disease,
         id_dis_sub) %>%
  mutate(last_updated = NA) %>%
```

```
  filter(!disease == '') %>%
  select(id_disease,id_dis_sub, disease, last_updated, panelapp_v)

#Disease_Subgroup tab
subtab <- toadd %>%
  select('disease_subgroup' = 'Dis_sub',
         id_dis_sub,
         id_dis_group) %>%
  distinct() %>%
  filter(!disease_subgroup %in%
db.list$disease_subgroup$disease_subgroup) %>%
  filter(!disease_subgroup == '') %>%
  select(id_dis_sub, id_dis_group, disease_subgroup)


# Group tab
grptab <- toadd %>%
  select('disease_group'='Dis_group') %>%
  distinct() %>%
  filter(!disease_group %in% db.list$disease_group$disease_group)
%>%
  filter(!disease_group == '') %>%
  mutate(id_dis_group = NA) %>%
  select(id_dis_group, disease_group)

#Write to db
dbWriteTable(conn = con1,
             name = "disease_subgroup",
             value = subtab,
             row.names=FALSE,
             append=TRUE,
             overwrite = F)
dbWriteTable(conn = con1,
             name = "disease",
             value = distab,
             row.names=FALSE,
             append=TRUE,
             overwrite = F)

#Just need to update the panelapp v
tmp_disease <-
unique(panels.df[,c('Dis','Dis_sub','Dis_group','NV')])
#For now removing any new diseases from panelapp api but not
included in the master list received from Damian
tmp_disease <- tmp_disease[tolower(tmp_disease$Dis) %in%
tolower(db.list$disease$disease),]
```

```
#Create index for looping through only those changed
if(dim(tmp_disease)[1] > 0){
  updated.idx <- c()
  for(i in 1:nrow(tmp_disease)){
    d.idx <- grep(tolower(tmp_disease$Dis[i]),
tolower(db.list$disease$disease), fixed = T)
    if((db.list$disease$panelapp_v[d.idx] !=
tmp_disease$NV[i])|(is.na(db.list$disease$panelapp_v[d.idx]))){
      db.list$disease$panelapp_v[d.idx] <- tmp_disease$NV[i]
      updated.idx <- c(updated.idx,d.idx)
    }
  }

  #Overwrite panelapp_v
  #Can't do this in one dbWriteTable, just loop as won't be updating
that many in any one go
  for(x in updated.idx){
    id <- db.list$disease$id_disease[x]
    v <- db.list$disease$panelapp_v[x]
    s.sql <- paste("UPDATE disease SET panelapp_v = ", v, " WHERE
id_disease = ", id, ";", sep='')
    dbSendQuery(con1,statement= s.sql)
  }
}
## DISEASE END ##

## GENE TO DISEASE
#Get updated version of the tables
db.list2 <- list()
db.list2[['gene']] <- dbGetQuery(con1, 'select * from gene')
db.list2[['disease']] <- dbGetQuery(con1, 'select * from disease')

#Get IDs of any genes and disease combos
g_d <- panels.df[,c('Dis','Gene')]

g_d %<>% left_join(db.list2$disease, by=c('Dis'='disease')) %>%
  left_join(db.list2$gene, by=c('Gene'='hgnc_symbol')) %>%
  select('Dis','Gene','panelapp_v','id_gene','id_disease')

g_d %<>% drop_na(id_gene,id_disease)

g_d %<>% mutate(Check = ifelse(is.na(match(paste0(g_d$id_disease,
g_d$id_gene),
                                paste0(db.list$gene_dis$id_disease,
```

```
 db.list$gene_dis$id_gene))),"No", "Yes")) %>%
   filter(Check == 'No') %>%
   select('id_disease','id_gene')

 g_d$id_gene_to_disease <- NA
 g_d <- g_d[,c('id_gene_to_disease','id_disease','id_gene')]

 #Unique ID, composite of disease ID and gene ID
 g_d$id_gene_to_disease <- paste(g_d$id_disease,g_d$id_gene,sep='.')

 #Add to the disease/gene mapping tablerm(l)
 dbWriteTable(conn = con1, name = "gene_to_disease", value = g_d,
 row.names=FALSE, append=TRUE, overwrite = F)

 ## GENES TO DISEASE END ##

 #Disconnect from DB
 dbDisconnect(con1)
 quit()
### END ###
```

## dis_to_pheno_mapping.R

```
#!/usr/bin/env r

######################################################################
#
## Project: GEL_DB
## Script purpose: Create phenotype to gene mapping tables
## Date: 2018-03-22
## Author: Dan Rhodes
######################################################################
#

# Setup
----------------------------------------------------------------------
pkgs <- c('RMySQL',
          'dplyr',
          'biomaRt')

if (!require("pacman")) install.packages('pacman')
pacman::p_load(pkgs, character.only = T)

rm(pkgs)

# Data load
----------------------------------------------------------------------
BashArgs = commandArgs(trailingOnly=TRUE)

source('/mnt/volume/GEL_DB/DB_setup/Scripts/db_connect.R')

con1 <- start_con(BashArgs)

db.list <- list()
db.list[['pheno']] <- dbGetQuery(con1, 'select * from phenotype')
db.list[['gene']] <- dbGetQuery(con1,'select * from gene')
db.list[['pheno_gene']] <- dbGetQuery(con1,'select * from
pheno_to_gene')

#HPO data
hpo.dat <-
read.table('/mnt/volume/GEL_DB/DB_setup/HPO_files/archive/annotation
/ALL_SOURCES_ALL_FREQUENCIES_phenotype_to_genes.txt', sep='\t')
names(hpo.dat) <- c('HPO.id','Phenotype','Entrez','Gene.symbol')

# Subset
----------------------------------------------------------------------
```

```
hpo.sub <- hpo.dat[hpo.dat$HPO.id %in% db.list$pheno$id_phenotype,]
%>%
  tbl_df %>%
  mutate_at(.,vars(-Entrez), as.character)

rm(hpo.dat)

#Check if new genes
new.genes <- as.character(unique(hpo.sub$Gene.symbol))
new.genes <- new.genes[!(toupper(new.genes) %in%
toupper(db.list$gene$hgnc_symbol))]

#Get ensembl IDs
mart <- useMart(biomart = "ensembl", dataset =
"hsapiens_gene_ensembl")
results <- getBM(attributes = c("hgnc_symbol","ensembl_gene_id"),
filters = "hgnc_symbol",
                 values = new.genes, mart = mart, uniqueRows = T)

#If multiple ensembl IDs exist, drop all ensembl IDs for this gene
dup.idx <- which(duplicated(results$hgnc_symbol) |
duplicated(results$hgnc_symbol, fromLast = TRUE))
if(length(dup.idx) > 0){
  results[dup.idx,'ensembl_gene_id'] <- NA
}
results <- unique(results)

#Add new genes to db
if(dim(results)[1] > 0){
  results$id_gene <- NA
  results <- results[,c('id_gene','ensembl_gene_id','hgnc_symbol')]
  names(results) <- names(db.list$gene)
}

if(nrow(results) > 0){
  #Add unique key
  if(nrow(db.list$gene) > 0){
    start.idx <- max(db.list$gene$id_gene)
  } else {
    start.idx <- 0
  }
  results$id_gene <- seq(start.idx + 1, (start.idx + 1) +
nrow(results)-1)

  #Add to gene.db
  dbWriteTable(conn = con1, name = "gene", value = results,
```

```
row.names=FALSE, append=TRUE, overwrite = F)
}

#Get updated v.
gene.db2 <- dbGetQuery(con1,'select * from gene') %>%
  as_tibble()

gene.db2 %<>% left_join(hpo.sub, by =
c('hgnc_symbol'='Gene.symbol')) %>%
  unique

### Mapping table
#Check if combo exists
gene.db2$Check <- ifelse(is.na(match(paste0(gene.db2$HPO.id,
gene.db2$id_gene),

paste0(db.list$pheno_gene$id_phenotype,
db.list$pheno_geneb$id_gene))),"No", "Yes")


ptg.map <- gene.db2[gene.db2$Check =='No',c('HPO.id','id_gene')]
ptg.map$id_pheno_to_gene <- paste(ptg.map$HPO.id, ptg.map$id_gene,
sep='.')
ptg.map <- ptg.map[,c('id_pheno_to_gene','HPO.id','id_gene')]
names(ptg.map) <- names(db.list$pheno_gene)

#Write to db
dbWriteTable(conn = con1, name = "pheno_to_gene", value = ptg.map,
row.names=FALSE, append=TRUE, overwrite = F)
quit()
### END ###
```

## DGIdb_api.R

```
#!/usr/bin/env r

#########################################################################
#
## Project: GEL_DB
## Script purpose: Get data from DGIdb API data haromise with GEL DB
## Date: 2018-03-22
## Author: Dan Rhodes
#########################################################################
#

#Takes genes from db, calls DGiDB API and gets drug-gene interaction
info
#Adds new drugs to drug db and new drug/gene combinations to mapping
table


# Setup
-----------------------------------------------------------------
# Specific steps required for rPython install
# If install fails run "sudo apt-get install python-dev" in terminal
# or "sudo apt-get install pip3-python | pip3 install python-dev" if
using pip3
# install.packages("rPython", configure.vars=
"RPYTHON_PYTHON_VERSION=3") # in R
pkgs <- c('data.table',
          'dplyr',
          'Hmisc',
          'rDGIdb',
          'RMySQL',
          'magrittr',
          'rPython',
          'tidyr')

if (!require("pacman")) install.packages('pacman')
pacman::p_load(pkgs, character.only = T)

rm(pkgs)



# Functions
-------------------------------------------------------------
```

```
### Data cleaning funcs ###
drug_names <- function(x){
  #Get rid of drug differences caused by punctuation
  uni <- unique(x$drug_name)
  uni <- sapply(uni, function(x){gsub(pattern =
'[[:punct:][:blank:]]',
                                      replacement = "", x)})
  dups <- uni[uni %in% uni[duplicated(uni)]] %>% sort
  #Keep the one with more information

  while(length(dups) > 0){
    d <- 1
    #print(names(dups)[d])
    idx <- grep(dups[d], dups)
    tmp <- x[grep(escapeRegex(paste0(names(dups)[idx], collapse =
'|')), x$drug_name),]

    if(any(is.na(tmp$interaction_type))){
      #Cases where one interaction type is na
      todrop <- tmp$drug_name[which(is.na(tmp$interaction_type))]
    } else if(length(table(tmp$interaction_type)) == 1){
      #Cases where they are of the same type, keep with punct
      todrop <- grep('[[:punct:]]', tmp$drug_name)
      if(length(todrop) > 1){
        todrop <- todrop[-1]
      } else if(length(todrop) == 1){
        todrop <- tmp$drug_name[-todrop]
      }
      x <- x[-grep(todrop, x$drug_name),]
    }
    #print(paste('Done',names(dups)[d]))
    dups <- dups[-idx]
  }
  return(x)
}

interaction_types <- function(x){
  # Lets cut down the interaction types to two main types if known
  #adding a look around grep to match agonist but ignore antagonist
for ags, requires perl = t in grep
  ags <- c('^(?!.*ant).*agonist','activator', 'stimulator',
'positive allosteric modulator',
           'inducer','cofactor', 'allosteric modulator',
'potentiator')
  ants <- c('antagonist','blocker','channel blocker', 'inhibitor',
'negative modulator')
```

```
  x$clean_interaction_type <- NA
  x$clean_interaction_type[grep(paste0(ags,collapse = '|'),
x$interaction_type, ignore.case = T, perl = T) ] <- 'agonist'
  x$clean_interaction_type[grep(paste0(ants,collapse = '|'),
x$interaction_type, ignore.case = T) ] <- 'inhibitor'
  x$clean_interaction_type[is.na(x$interaction_type)] <- 'unknown'
  x$clean_interaction_type[is.na(x$clean_interaction_type)] <-
'other'
  x %<>% distinct(gene_name, drug_name, gene_categories,
clean_interaction_type, .keep_all = T)
  #Deal with NAs
  dups <- x[duplicated(x[,c('drug_name','gene_name')]) |
duplicated(x[,c('drug_name','gene_name')], fromLast=TRUE),]
  #Get rid of unknowns in this group first
  x <- anti_join(x, dups[dups$clean_interaction_type == 'unknown',])
  dups <- x[duplicated(x[,c('drug_name','gene_name')]) |
duplicated(x[,c('drug_name','gene_name')], fromLast=TRUE),]
  #Sort conflicts if dups still remain
  if(nrow(dups) > 0){
    n <- 1
    todrop <- list()
    while(nrow(dups) > 0){
      d <- 1
      idx <- grep(paste0(dups$gene_name[d],dups$drug_name[d]),
paste0(dups$gene_name,dups$drug_name))
      tmp <- dups[idx,]
      if('inhibitor' %in% names(table(tmp$clean_interaction_type))){
        todrop[[n]] <-
tmp[-grep('inhibitor',tmp$clean_interaction_type),]
      }
      if('agonist' %in% names(table(tmp$clean_interaction_type))){
        todrop[[n]] <-
tmp[-grep('agonist',tmp$clean_interaction_type),]
      }
      if('other' %in% names(table(tmp$clean_interaction_type))){
        todrop[[n]] <-
tmp[-grep('other',tmp$clean_interaction_type),]
      }
      dups <- dups[-idx,]
      n <- n + 1
    }
    todrop <- do.call(rbind, todrop)
    x <- anti_join(x, todrop)
  }
  return(x)
}
```

```
drug_clean <- function(x){
  cat('Dimensions before cleaning:', dim(x))
  x$drug_name <- toupper(x$drug_name)
  #Apply cleaning funcs
  x <- drug_names(x)
  x <- interaction_types(x)
  cat('Dimensions after cleaning:', dim(x))
  return(x)
}
###

# Data import
----------------------------------------------------------------
BashArgs = commandArgs(trailingOnly=TRUE)

source('/mnt/volume/GEL_DB/DB_setup/Scripts/db_connect.R')
source('/mnt/volume/GEL_DB/DB_setup/Scripts/gene_cats.R')

con1 <- start_con(BashArgs)

# Get DB data
----------------------------------------------------------------
db.tabs <- dbListTables(con1)
db.list <- pull_data(db.tabs, con = con1)

gene.query <- db.list$gene$hgnc_symbol %>% unique


# DGIdb API
----------------------------------------------------------------
#Using python script
py <- 'python3'
script <- '/mnt/volume/GEL_DB/DB_setup/Scripts/DGIdb_API.py'


py.args <- paste0(gene.query, collapse = ',') %>%
paste0("--genes=","'", ., "'")
outter <- " > /mnt/volume/GEL_DB/DB_setup/ResTmp.txt"
outter <- paste(py.args, outter )
# Add path to script as first arg
allArgs = c(script, outter)

system2(py, args=allArgs)
```

```
# Clean data
------------------------------------------------------------
tmp <- readLines('/mnt/volume/GEL_DB/DB_setup/ResTmp.txt')
if(length(grep('^Possible|^Unmatched', tmp)) > 0){
  tmp <- tmp[-grep('^Possible|^Unmatched', tmp)]
}
res <- read.table(textConnection(tmp), stringsAsFactors = F, header
= T, sep = '\t') %>% as_tibble
system('rm ~/GEL_DB/DB_setup/ResTmp.txt')
rm(tmp)

res$source <- NULL
res %<>% unique()

res$interaction_type[res$interaction_type == ''] <- NA
res <- drug_clean(res)


# Cross-check against DB
------------------------------------------------

###Add new drugs to drug.db
new.drugs <- res[!(res$drug_name %in%
db.list$drug$drug_name),]$drug_name %>%
  tbl_df %>%
  unique
new.drugs$id_drug <- NA
new.drugs <- new.drugs[,c('id_drug','value')]
names(new.drugs) <- c('id_drug','drug_name')


if(nrow(db.list$drug) > 0){
  start.idx <- max(db.list$drug$id_drug)
} else {
  start.idx <- 0
}
new.drugs$id_drug <- seq(start.idx + 1, (start.idx + 1) +
nrow(new.drugs)-1)

#Add to drug.db
dbWriteTable(conn = con1, name = "drug", value = new.drugs,
row.names=FALSE, append=TRUE, overwrite = F)

#Read in updated db
drugs.db2 <- dbGetQuery(con1,"SELECT * FROM drug")
###
```

```
## Mapping table ##
#Check that specific drug to gene link is new
dg.df <-
res[,c('gene_name','drug_name','interaction_type','clean_interaction
_type')]
dg.df$id_drug <- NA
dg.df$id_gene <- NA

#Get gene and drug IDs
for(i in 1:nrow(dg.df)){
  tryCatch({
    #escapeRegex() deals with parentheses therefore allowing str to
be treated as regex and not literal
    dg.df[i,'id_drug'] <- drugs.db2[grep(paste('^',
escapeRegex(dg.df$drug_name[i]), '$', sep=''), drugs.db2$drug_name,
ignore.case = T),'id_drug']
  },error = function(e){cat('ERROR:', conditionMessage(e), 'For
Drug', dg.df$drug_name[i],'\n')})
  tryCatch({
    dg.df[i,'id_gene'] <-  db.list$gene[grep(paste('^',
dg.df$gene_name[i], '$', sep=''), db.list$gene$hgnc_symbol,
ignore.case = T),'id_gene']
  },error = function(e){cat('ERROR:', conditionMessage(e), 'For
Gene', as.character(dg.df$gene_name)[i],'\n')})
}

#These are all pre-existing genes, so just need to list the new
drugs into the db

dg.df$Check <- ifelse(is.na(match(paste0(dg.df$drug_name,
dg.df$gene_name),
                                  paste0(db.list$drug_gene$id_drug,
db.list$drug_gene$id_gene))),"No", "Yes")


dg_upload <- dg.df[dg.df$Check ==
'No',c('id_drug','id_gene','interaction_type',
'clean_interaction_type')]
dg_upload$id_drug_to_gene <-
paste(dg_upload$id_drug,dg_upload$id_gene,sep='.')
dg_upload <-
dg_upload[,c('id_drug_to_gene','id_drug','id_gene','interaction_type
', 'clean_interaction_type')]
```

```
dbWriteTable(conn = con1, name = "drug_to_gene", value = dg_upload,
row.names=FALSE, append=TRUE, overwrite = F)



### Add DGIdb gene cats to gene data

res.sub <- res %>%
  select(gene_name, gene_categories) %>%
  distinct()

res.sub$gene_categories[res.sub$gene_categories == ''] <- NA
res.sub %<>% na.omit

#Add new cats if there are any
new_cats(res.sub$gene_categories, source = 'DGIdb')
#Add data
map_cats(res, db.list)


closecon(x)
quit()
### End ###
```

## SMILE.R

```
#!/usr/bin/env r

#########################################################################
#
## Project: GEL_DB
## Script purpose: Get SMILE data
## Date: 2018-03-22
## Author: Dan Rhodes
#########################################################################
#


# Setup
-------------------------------------------------------------------
pkgs <- c('dplyr',
          'Hmisc',
          'webchem',
          'RMySQL',
          'httr')

if (!require("pacman")) install.packages('pacman')
pacman::p_load(pkgs, character.only = T)

rm(pkgs)

# Connection to db
-----------------------------------------------------------
BashArgs = commandArgs(trailingOnly=TRUE)

source('/mnt/volume/GEL_DB/DB_setup/Scripts/db_connect.R')

con1 <- start_con(BashArgs)
db.tabs <- dbListTables(con1)

drugs.db <- dbGetQuery(con1,"SELECT * FROM drug")
drugs.db.orig <- drugs.db

# Query SMILES
-------------------------------------------------------------

# Data from
(http://cactus.nci.nih.gov/chemical/structure_documentation)
#Only looking up those with NAs
```

```
to_update <- which(is.na(drugs.db$smile))
updated.idx <- c()
for(i in to_update){
  i.query <- drugs.db$drug_name[i] %>% URLencode(reserved = T)
  url.query <-
paste0('https://cactus.nci.nih.gov/chemical/structure/',i.query,'/sm
iles')
  res <- GET(url.query)
  if(http_status(res)$category == "Success"){
  drugs.db[i,'smile'] <- content(res)
  updated.idx <- c(updated.idx,i)
  } else {
    print(paste(drugs.db$drug_name[i],'not found'))
    }
}

#Update those that have smiles added
for(i in updated.idx){
  id <- drugs.db$id_drug[i]
  smile <- drugs.db$smile[i]
  s.sql <- paste("UPDATE drug SET smile = ",
AnnotationDbi::toSQLStringSet(smile), " WHERE id_drug = ", id, ";",
sep='')
  dbSendQuery(con1,statement= s.sql)
}

#Disconnect from DB
dbDisconnect(con1)
quit()
```

## drugbank_parse.R

```
#!/usr/bin/env r

##################################################
## Project: GEL_DB
## Script purpose: Parse drugbank data
## Date: 2018-07-10
## Author: Dan Rhodes
##################################################


#Take drugbank data downloaded from website, take what's needed
# e.g. data obtain -
# curl -Lfv -o drug_structure.zip -u username:password
https://www.drugbank.ca/releases/5-1-1/downloads/all-structures


# Setup
----------------------------------------------------------------------
pkgs <- c('dplyr',
          'RMySQL',
          'Biostrings',
          'magrittr')
if (!require("pacman")) install.packages('pacman')
pacman::p_load(pkgs, character.only = T)
rm(pkgs)

# Data import
------------------------------------------------------------
BashArgs = commandArgs(trailingOnly=TRUE)

if(Sys.info()["nodename"] == 'dan-XPS-13-9350'){

source('/home/dan/Documents/Data/Visible/QMUL/GeL/Disease_ontology/G
EL_DB/DB_setup/Scripts/db_connect.R')
} else {
  source('/mnt/volume/GEL_DB/DB_setup/Scripts/db_connect.R')
}

con1 <- start_con(BashArgs)
db.tabs <- dbListTables(con1)

drugs.db <- dbGetQuery(con1,"SELECT * FROM drug")
drugs.db.orig <- drugs.db
```

```
# Import
-----------------------------------------------------------------
#Links to other data sources
DatSource <-
read.csv('/mnt/volume/GEL_DB/DB_setup/Drugbank/drug_links.csv',
header = T)
#SDF data for drugs
#datSDF <-
read.csv('/mnt/volume/GEL_DB/DB_setup/Drugbank/structures.sdf',
header = T)
#SMILE data
datSM <-
read.csv('/mnt/volume/GEL_DB/DB_setup/Drugbank/structure_links.csv',
header = T)
#FASTA with biologic sequence data
datFA <-
readAAStringSet('/mnt/volume/GEL_DB/DB_setup/Drugbank/drug_sequences
.fasta')

# Wrangle
-----------------------------------------------------------------
#Data on drug IDs is not available from DGIdb - so try to harmonise
with the drugbank data
harm <- drugs.db[which(is.na(drugs.db$id_drugbank)),]
harm %<>% mutate(d_name = tolower(drug_name))
datSM %<>% mutate(d_name = tolower(Name)) %>% select(Name,
Drug.Groups, SMILES, d_name)
harm %<>% left_join(datSM, by = 'd_name') %>% select(id_drug,
drug_name, d_name, Drug.Groups, smile=SMILES)

#Lets add data from DatSource and match cols with drugs.db
DatSource %<>% mutate(d_name = tolower(Name))
harm %<>% left_join(DatSource, by = 'd_name') %>% select(id_drug,
                                                         drug_name,
                                                         smile,
drug_type=Drug.Type,
drug_groups=Drug.Groups,
id_drugbank=DrugBank.ID,
cas_number=CAS.Number,
id_pubchem_compound=PubChem.Compound.ID,
id_chebi=ChEBI.ID,
id_chemspider=ChemSpider.ID)


#Get rid of cases that haven't actually been changed (no drugbank
data)
```

```
harm %<>% mutate(na.count = rowSums(is.na(.))) %>%
  filter(na.count != 8) %>%
  select(-na.count)
harm$smile <- harm$smile %>% as.character()

# Add to DB
----------------------------------------------------------------
#Will have to update rather than overwrite
for(i in 1:nrow(harm)){
  smile <- harm$smile[i]
  s.sql <- paste("UPDATE drug SET smile = ",
AnnotationDbi::toSQLStringSet(smile),
                 ", drug_type = ", "'", harm$drug_type[i], "'",
                 ", drug_groups = ", "'", harm$drug_groups[i], "'",
                 ", id_drugbank = ", "'", harm$id_drugbank[i], "'",
                 ", cas_number = ", "'", harm$cas_number[i], "'",
                 ", id_pubchem_compound = ", "'",
harm$id_pubchem_compound[i], "'",
                 ", id_chebi = ", "'", harm$id_chebi[i], "'",
                 ", id_chemspider = ", "'", harm$id_chemspider[i],
"'",
                 " WHERE id_drug = ", "'", harm$id_drug[i], "'",
";", sep='')
  s.sql <- gsub("'NA'","NULL", s.sql)
  dbSendQuery(con1,statement= s.sql)
}

closecon(X)
```

## drug_pheno_disease_mappings.R

```
#!/usr/bin/env r

################################################################################
#
## Project: GEL_DB
## Script purpose: Disease and phenotype to drug mapping tables
## Date: 2018-03-22
## Author: Dan Rhodes
################################################################################
#

# Create mapping tables based on drug information.

# Libs
--------------------------------------------------------------------
pkgs <- c('dplyr',
          'RMySQL',
          'magrittr',
          'tidyr')

if (!require("pacman")) install.packages('pacman')
pacman::p_load(pkgs, character.only = T)

rm(pkgs)

# Data import
-------------------------------------------------------------
BashArgs = commandArgs(trailingOnly=TRUE)

source('/mnt/volume/GEL_DB/DB_setup/Scripts/db_connect.R')

con1 <- start_con(BashArgs)

db.tabs <- dbListTables(con1)
db.list <- list()
db.list[['disease']] <- dbGetQuery(con1, 'select * from disease')
db.list[['pheno']] <- dbGetQuery(con1, 'select * from phenotype')
db.list[['disease']] <- dbGetQuery(con1, 'select * from disease')
db.list[['drug']] <- dbGetQuery(con1, 'select * from drug')
db.list[['drug_to_gene']] <- dbGetQuery(con1, 'select * from
drug_to_gene')
```

```
db.list[['gene']] <- dbGetQuery(con1, 'select * from gene')
db.list[['gene_to_disease']] <- dbGetQuery(con1, 'select * from
gene_to_disease')
db.list[['pheno_to_gene']] <- dbGetQuery(con1, 'select * from
pheno_to_gene')
db.list[['disease_to_drug']] <- dbGetQuery(con1, 'select * from
disease_to_drug')
db.list[['pheno_to_drug']] <- dbGetQuery(con1, 'select * from
pheno_to_drug')

# Disease to drug mapping
-----------------------------------------------

#Need to link the drug to the gene, then that gene to its associated
diseases disease
gd <- db.list$drug_to_gene %>% left_join(db.list$gene_to_disease, by
= c("id_gene"="id_gene")) %>%
  select(id_drug, id_disease) %>%
  unique

gd$id_disease_to_drug <- paste(gd$id_disease, gd$id_drug, sep = '_')
gd <- gd[,c(dbListFields(con1,'disease_to_drug'))]
gd <- gd[!(gd$id_disease_to_drug %in%
db.list$disease_to_drug$id_disease_to_drug),]

# Phenotype to drug mapping
-----------------------------------------------
pd <- db.list$drug_to_gene %>% left_join(db.list$pheno_to_gene, by =
c("id_gene"="id_gene")) %>%
  select(id_drug, id_phenotype) %>%
  unique

pd$id_pheno_to_drug <- paste(pd$id_pheno, pd$id_drug, sep = '_')
pd <- pd[,c(dbListFields(con1,'pheno_to_drug'))]
pd <- pd[!(pd$id_pheno_to_drug %in%
db.list$pheno_to_drug$id_pheno_to_drug),]

# Add to db
-----------------------------------------------------------------
to_upload <- list()
to_upload[['disease_to_drug']] <- gd
to_upload[['pheno_to_drug']] <- pd

#Upload
for(n in names(to_upload)){
  dbWriteTable(conn = con1, name = n, value = to_upload[[n]],
```

```
row.names=FALSE, append=T, overwrite = F)
  cat(dim(to_upload[[n]])[1], 'records updated in table ', n, '\n')
}


### END ###
```

## MySQL GEL database schema

```
-- MySQL dump 10.13  Distrib 5.7.22, for Linux (x86_64)
--
-- Host: localhost    Database: GEL
-- ------------------------------------------------------
-- Server version 5.7.22-0ubuntu0.16.04.1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `dis_to_pheno`
--

DROP TABLE IF EXISTS `dis_to_pheno`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `dis_to_pheno` (
  `id_dis_pheno` varchar(16) NOT NULL,
  `id_phenotype` varchar(10) NOT NULL,
  `id_disease` int(5) NOT NULL,
  PRIMARY KEY (`id_dis_pheno`),
  KEY `fk_dis_to_pheno_1_idx` (`id_disease`),
  KEY `fk_dis_to_pheno_2_idx` (`id_phenotype`),
  CONSTRAINT `fk_dis_to_pheno_1` FOREIGN KEY (`id_disease`)
REFERENCES `disease` (`id_disease`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `fk_dis_to_pheno_2` FOREIGN KEY (`id_phenotype`)
REFERENCES `phenotype` (`id_phenotype`) ON DELETE CASCADE ON UPDATE
```

```
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Mapping table for many
to many relationship between diseases and phenotypes';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `disease`
--

DROP TABLE IF EXISTS `disease`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `disease` (
  `id_disease` int(5) NOT NULL,
  `id_dis_sub` int(5) NOT NULL,
  `disease` varchar(100) DEFAULT NULL,
  `last_updated` date DEFAULT NULL,
  `panelapp_v` varchar(45) DEFAULT '0.0',
  PRIMARY KEY (`id_disease`),
  KEY `dis_subgroup_idx` (`id_dis_sub`),
  CONSTRAINT `dis_subgroup` FOREIGN KEY (`id_dis_sub`) REFERENCES
`disease_subgroup` (`id_dis_sub`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `disease_group`
--

DROP TABLE IF EXISTS `disease_group`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `disease_group` (
  `id_dis_group` int(5) NOT NULL,
  `disease_group` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_dis_group`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Disease groups as
defined by GEL';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `disease_subgroup`
--

DROP TABLE IF EXISTS `disease_subgroup`;
```

```
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `disease_subgroup` (
  `id_dis_sub` int(5) NOT NULL,
  `id_dis_group` int(5) NOT NULL,
  `disease_subgroup` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_dis_sub`),
  KEY `fk_disease_subgroup_1_idx` (`id_dis_group`),
  CONSTRAINT `fk_disease_subgroup_1` FOREIGN KEY (`id_dis_group`)
REFERENCES `disease_group` (`id_dis_group`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `disease_to_drug`
--

DROP TABLE IF EXISTS `disease_to_drug`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `disease_to_drug` (
  `id_disease_to_drug` varchar(11) NOT NULL,
  `id_disease` int(5) DEFAULT NULL,
  `id_drug` int(5) DEFAULT NULL,
  PRIMARY KEY (`id_disease_to_drug`),
  KEY `fk_disease_to_drug_1_idx` (`id_disease`),
  KEY `fk_disease_to_drug_2_idx` (`id_drug`),
  CONSTRAINT `fk_disease_to_drug_1` FOREIGN KEY (`id_disease`)
REFERENCES `disease` (`id_disease`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `fk_disease_to_drug_2` FOREIGN KEY (`id_drug`)
REFERENCES `drug` (`id_drug`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `drug`
--

DROP TABLE IF EXISTS `drug`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `drug` (
  `id_drug` int(11) NOT NULL,
  `drug_name` varchar(200) DEFAULT NULL,
```

```
  `smile` varchar(10000) DEFAULT NULL,
  `drug_type` varchar(100) DEFAULT NULL,
  `drug_groups` varchar(200) DEFAULT NULL,
  `id_drugbank` varchar(7) DEFAULT NULL,
  `cas_number` varchar(30) DEFAULT NULL,
  `id_pubchem_compound` int(20) DEFAULT NULL,
  `id_chebi` int(20) DEFAULT NULL,
  `id_chemspider` int(20) DEFAULT NULL,
  PRIMARY KEY (`id_drug`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `drug_to_gene`
--

DROP TABLE IF EXISTS `drug_to_gene`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `drug_to_gene` (
  `id_drug_to_gene` varchar(11) NOT NULL,
  `id_drug` int(5) DEFAULT NULL,
  `id_gene` int(5) DEFAULT NULL,
  `interaction_type` varchar(100) DEFAULT NULL,
  `clean_interaction_type` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_drug_to_gene`),
  KEY `fk_drug_to_gene_1_idx` (`id_drug`),
  KEY `fk_drug_to_gene_2_idx` (`id_gene`),
  CONSTRAINT `fk_drug_to_gene_1` FOREIGN KEY (`id_drug`) REFERENCES
`drug` (`id_drug`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_drug_to_gene_2` FOREIGN KEY (`id_gene`) REFERENCES
`gene` (`id_gene`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `gene`
--

DROP TABLE IF EXISTS `gene`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `gene` (
  `id_gene` int(5) NOT NULL,
  `ensembl_id` varchar(45) DEFAULT NULL,
  `hgnc_symbol` varchar(45) DEFAULT NULL,
```

```
  PRIMARY KEY (`id_gene`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `gene_cats`
--

DROP TABLE IF EXISTS `gene_cats`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `gene_cats` (
  `id_cat` int(11) NOT NULL,
  `category` varchar(45) DEFAULT NULL,
  `source` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_cat`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `gene_to_disease`
--

DROP TABLE IF EXISTS `gene_to_disease`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `gene_to_disease` (
  `id_gene_to_disease` varchar(11) NOT NULL,
  `id_disease` int(5) DEFAULT NULL,
  `id_gene` int(5) DEFAULT NULL,
  PRIMARY KEY (`id_gene_to_disease`),
  KEY `fk_gene_to_disease_1_idx` (`id_disease`),
  KEY `fk_gene_to_disease_2_idx` (`id_gene`),
  CONSTRAINT `fk_gene_to_disease_1` FOREIGN KEY (`id_disease`)
REFERENCES `disease` (`id_disease`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `fk_gene_to_disease_2` FOREIGN KEY (`id_gene`)
REFERENCES `gene` (`id_gene`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `gene_to_genecat`
--

DROP TABLE IF EXISTS `gene_to_genecat`;
```

```
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `gene_to_genecat` (
  `id_gene_to_genecat` varchar(11) NOT NULL,
  `id_cat` int(5) DEFAULT NULL,
  `id_gene` int(5) DEFAULT NULL,
  PRIMARY KEY (`id_gene_to_genecat`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `pheno_to_drug`
--

DROP TABLE IF EXISTS `pheno_to_drug`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `pheno_to_drug` (
  `id_pheno_to_drug` varchar(16) NOT NULL,
  `id_phenotype` varchar(10) DEFAULT NULL,
  `id_drug` int(5) DEFAULT NULL,
  PRIMARY KEY (`id_pheno_to_drug`),
  KEY `fk_pheno_to_drug_1_idx` (`id_phenotype`),
  KEY `fk_pheno_to_drug_2_idx` (`id_drug`),
  CONSTRAINT `fk_pheno_to_drug_1` FOREIGN KEY (`id_phenotype`)
REFERENCES `phenotype` (`id_phenotype`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `fk_pheno_to_drug_2` FOREIGN KEY (`id_drug`) REFERENCES
`drug` (`id_drug`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `pheno_to_gene`
--

DROP TABLE IF EXISTS `pheno_to_gene`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `pheno_to_gene` (
  `id_pheno_to_gene` varchar(16) NOT NULL,
  `id_phenotype` varchar(10) DEFAULT NULL,
  `id_gene` int(5) DEFAULT NULL,
  PRIMARY KEY (`id_pheno_to_gene`),
  KEY `fk_pheno_to_gene_1_idx` (`id_phenotype`),
  KEY `fk_pheno_to_gene_2_idx` (`id_gene`),
```

```
  CONSTRAINT `fk_pheno_to_gene_1` FOREIGN KEY (`id_phenotype`)
REFERENCES `phenotype` (`id_phenotype`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `fk_pheno_to_gene_2` FOREIGN KEY (`id_gene`) REFERENCES
`gene` (`id_gene`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `phenotype`
--

DROP TABLE IF EXISTS `phenotype`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `phenotype` (
  `id_phenotype` varchar(10) NOT NULL,
  `phenotype` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_phenotype`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `test`
--

DROP TABLE IF EXISTS `test`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `test` (
  `id_test` float NOT NULL,
  `test` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_test`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `test_to_disease`
--

DROP TABLE IF EXISTS `test_to_disease`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `test_to_disease` (
  `id_test_to_disease` varchar(13) NOT NULL,
  `id_test` float DEFAULT NULL,
```

```
  `id_disease` int(5) DEFAULT NULL,
  PRIMARY KEY (`id_test_to_disease`),
  KEY `fk_test_to_disease_1_idx` (`id_test`),
  KEY `fk_test_to_disease_2_idx` (`id_disease`),
  CONSTRAINT `fk_test_to_disease_2` FOREIGN KEY (`id_disease`)
REFERENCES `disease` (`id_disease`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```