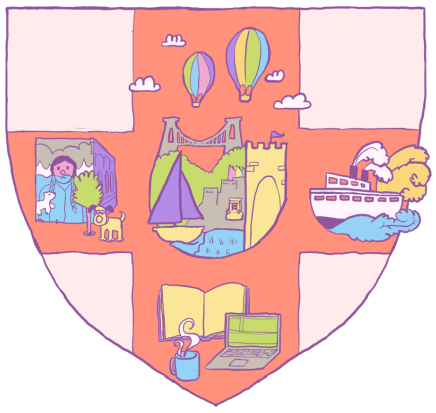
Phenotype and Function from Genotype: Combining Data Sources to Create Explanatory Predictions

Natalie Zelenka
Department of Computer Science
University of Bristol



A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy (PhD) in the Faculty of Engineering.

February 14, 2024 - Version 2.0

Word count: 57,873

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Signed: Natalie Zelenka

Date 16/02/2024

Abstract

The title of this thesis describes the ambitious scientific aim running through it: explaining the link between genotype and phenotype through molecular biology data. In our cells, proteins are constantly being created and are degrading. These cells are always accumulating and interacting, which leads to the emergence of the measurable human traits we call phenotypes such as height, levels of enzymes in blood, and diseases. There are hidden molecular explanations for these phenotypes, and for our proteins functions. The types of proteins that it is *possible* for an organism to produce are determined primarily by its protein-coding DNA. Meanwhile which of these possible proteins are *actively created* in each cell are determined by the environment of each cell at each time. The data about these molecules and their activity is our trail of breadcrumbs in the search for a molecular explanation for phenotype, and these data exist in computational biology's collection of large databases of community-sourced experimental and computational results.

This thesis explores two main approaches for making and improving explanatory predictions of phenotype and protein function from genotype. Both predictors seek to leverage the power of the researchers around the world which contribute their results to community databases, and combines these where possible to get a fuller picture of the complex system of interacting molecules.

The first part of this thesis contains all of the necessary background, and contains three chapters. Chapter 1 briefly introduces the philosophy of this thesis. The biology background chapter (chapter 2) then presents a detailed overview of the scientific model that links genotype and phenotype. It tells the story of how phenotype arises from genotype, and introduces the different biological molecules that are involved. It begins at the very basics: what are DNA, RNA, proteins, and phenotypes; how are they related; how do we categorise them? This background is intended to make this thesis readable to someone without a background in biology, and to explain the overall aims and context of the research in this thesis. It does not contain any of my own research. The computational biology background chapter (chapter 3) follows on from the previous chapter by discussing popular resources in computational biology, their provenance, and the impact of this on the field. In this chapter, I also present my contributions to collaborative projects: the Proteome Quality Index paper[2], and the 2014 SUPERFAMILY update paper[3]

In the second part, I present the Snowflake phenotype predictor, which uses variants conservation scores, prevalence in the population, and protein domain architectures as input to an unsupervised learning method. This predictor, the development of which resulted in a patent[4], finds unusual combinations of variants associated with phenotypes, and is designed to create explanatory predictions of complex traits. The algorithm itself, and the results of experiments in validating Snowflake are presented in chapter 4.

In investigating Snowflake's predictions, it became clear that it was possible for it to include protein-coding SNPs in predictions about phenotypes that exist in tissues in which the protein is never expressed, which brings us to the third and final part of this thesis. The Filip protein function prediction filter is discussed in chapter 5, which uses gene expression data to filter out predictions of proteins which are not expressed in the tissue relating to a given phenotype. I discuss attempts to validate Filip's predictions, including it's performance in the CAFA3 protein function prediction competition[5]. In addition, this part presents tools and datasets that were developed through creating and developing Filip: Ontology a Python package for querying OBO files in chapter 6, and a combined data set of gene expression data in chapter 7.

Acknowledgements

I would like to thank the Engineering and Physical Sciences Research Council for funding my PhD, and the Bristol Centre for Complexity Sciences (BCCS) for being the incubator for it. The wide breadth of different topics that I was taught, and the companionship of my peer cohort made the masters year a particularly enjoyable experience. Having the opportunity to move from a background of maths and physics to the world of computational biology is something I don't take for granted.

A big thank you to my supervisor Professor Julian Gough. You have been kind, patient and encouraging, even when I was feeling defeated by this PhD. I'm also grateful to all the members of the Gough group who I have been lucky enough to get to know and learn from. You're also to blame/thank for my hipster coffee habit, which continues to spiral out of control.

I would also like to thank Dr Oliver Ray for supervising me for the latter part of my PhD. Your guidance and interesting conversations have been much appreciated.

Huge thanks also to my current colleagues the Jean Golding Institute for Data Science. I'm so grateful for the friendly atmosphere of our team, and the things I've been able to learn through being part of it. Particular thanks to Professor Kate Robson-Brown and Patricia Holley for their support and understanding while I finished this PhD. I really couldn't have done it without you! Even before I was part of the team, the Jean Golding Institute nurtured my interest in reproducible research and data science, by providing opportunities for continued learning and growing my confidence through the opportunities they offer. I'm now so pleased to be a part of the team and help other PhD students in the same way.

Outside of work, I am very lucky to have the most fantastic group of friends who have cooked me meals, listened to my woes, bought me a beer when I was skint, and just provided great company. Thank you SO much! Although she'll never read this, I'd also like to thank a different type of friend - my dog Biscuit - for making me excited to get up every morning to see her sweet face, and for taking me for a walk every day.

To my family, especially my parents: I would never have thought I could finish this PhD if I didn't have a lifetime of your encouragement behind me.

Finally, the biggest possible thanks to my husband Tom for being here for me throughout this long, long, LONG PhD. You helped me in so many ways to get through these years. And you're my favourite!

Table of Contents

Background

Phenotype prediction

Tissue-specific gene expression

1. Thesis style and philosophy

The abstract and background chapters provide the scientific introduction to this thesis, describing the research aims and context. This short chapter, on the other hand, explains the choices of style and research philosophy.

1.1. Unusual stylistic choices in this thesis

There are a few unusual things about the format of this thesis: some easter eggs that made it easier for me to push through and finish it, and which also represent some of the things I like most about research: making it transparent, inclusive, and accessible.

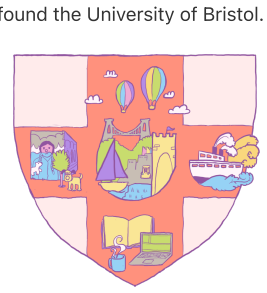
First, in the spirit of trying to make my work as reproducible and Open as possible, this thesis is available online as a Jupyter Book[1] here. I would recommend reading it online unless you really like your PDF viewer, since it includes some interactive features which don't translate to PDF. This book was written entirely in markdown documents and Jupyter Notebooks - which means that most of the graphs within are created directly from these notebooks.

The second unusual thing about this thesis is that you will see asides mentioning researchers who have been involved in eugenics and/or racism. This is the unfortunate reality of much of the history of the field, and I didn't want to highlight the scientific achievements of these individuals, without also acknowledging their legacy of scientific racism, particularly in the light of the Black Lives Matter movement.

For the same reason, I also drew an alternative University of Bristol crest, which you can see on the title page and in this margin comment. This is also an example of the third weird thing about this thesis, which is that I drew some illustrations for it (using Krita[6]), particularly in the background chapters. My aim in including the majority of these drawings was simply to illustrate concepts, and help the reader (and myself) imagine some of the incredible stuff that is going on in all of our bodies.

Bristol Crest

The official University of Bristol crest (above) has symbols for the Wills, Fry, and Colston families. These families made their wealth in industries built on slavery and used some of that wealth to found the University of Bristol.



I drew an alternative crest which has symbols for three of my favourite Bristol festivals instead: Upfest, the Balloon Fiesta, and St Paul's Carnival.

Illustrations

The illustrations are CC-BY licensed (use freely, with attribution) in case they are useful to anyone.

1.2. Research philosophy

Here I explain a little about my approach to the work in this thesis. I'm including this to add clarity about the lenses through which I did this work, as well as what I consider to be a scientific contribution.

1.2.1. Complexity science, systems biology and multi-omics

This PhD was completed as part of the Bristol Centre for Complexity Science. Complexity science is the study of systems of interacting parts, i.e. the study of parts of the world where reductionism breaks down. Typical applications of complexity science are predator-prey models, epidemiological modelling (e.g. of pandemics), protein-protein interaction networks, or models of neurons. When applied to biology, it often falls under the banner of Systems Biology.

Everything in this thesis looks at biology at the level of whole genomes, whole organisms, whole species, or even across the tree of life, and takes the view that this is necessary if we want to understand emergent properties of these interactions. Where I do "zoom in" to a particular case study or the details of some data, I am usually doing so either to understand how the complex systems approach is working, or in order to feed back and improve the resources that make the systems approach possible.

Emergence

Emergent properties of systems are properties that are found only when the consistuent entities of those systems interact, for example traffic jams emerge when vehicles interact on a network of roads[7], or cheetah's spots emerge when chemicals diffuse across cells[8].

The work in this thesis could also be considered multi-omics. I integrate, combine and harmonise some of the large, collaborative "omics" (e.g. genomics, proteomics) data projects that are available in this field, to create new resources and make new predictions.

A complex systems approach does not mean taking into account all parts of a system. In modelling our infinitely complicated reality, we have to simplify to some extent, whether this means not taking properties such as location or speed of reactions into account, not taking certain entities or classes of processes into account. Discovering what must be included and what can be left out is one outcome of this type of research.

1.2.2. Team science

Computational biology is a field which I see as characterised by excellent examples of team science, from the Human Genome Project to Biomedical Ontologies. I think that the best progress can be made when we all work together to create robust resources and build on each other's work and are fairly credited for that. I recognise that not everyone shares this view, computational biologists who use the results of other scientists experiments have been referred to by some as "research parasites", or seen as a branch of IT services to whom "real" scientists can export technical work. In this thesis, however, I take the view that contributing to existing scientific resources (Open Source or curated information), software engineering, writing, coming up with hypotheses all fall under the banner of scientific contribution. This is in line with policies like the Contributor Roles Taxonomy, and the recent additions of data sets and research software to the Research Excellent Framework (REF) research outputs.

1.2.3. Open and reproducible science

During my PhD, one aspect of research came as quite a surprise to me, which was that I couldn't trust the results of peer-reviewed papers to the extent that I originally assumed. As the reproducibility crisis unfolded, it became clear that the "untrustworthiness" of research was also an issue for many other researchers. Since then, it has been an important part of the way that I do research, and I have made as much of my work as reproducible and Open as I could, taking new items from the Buffet of Open Research[9] as I continued. I describe the ways in which I do this as they come up. Some earlier parts of my work, particularly chapter 4 do remain closed-source.

We might like to think that scientific research is the "view from nowhere", that it is objective, and simply measuring reality. The reality, however, is that there are many, often equally valid, decisions to make when doing research and these decisions will impact the results of that research (as I will explain later). If these decisions are not documented, the validity of the research can be obscured, and in addition, one aspect of scientific work that makes it difficult for other people to build on is that materials or details of analyses are not freely shared. For these reasons, in this thesis, I tried to include enough detail so that the work could be reproduced (repeated to get the same answer), and the decisions made understood.

2. How phenotype arises from genotype

I have three aims in this background Chapter:

1. To anchor the work that I've done within the context of the big questions in, and history of, genetics and computational biology.
2. To discuss the current model for how biological molecules impact phenotype, in order to aid in discussions about the types of data we have (which I discuss in Chapter 3), and how to use it (which relates to the rest of this thesis).
3. To provide a basic run-down of key terms/concepts in molecular biology (in particular, how dna, proteins, and phenotype are labelled and classified) in order to allow someone without a biology background to understand the rest of this thesis.

The biological background presented in this section begins at the very basics of molecular biology. This first details what biological molecules (e.g. DNA, RNA and proteins) are, then discusses the different levels that they can be viewed at, and how current research suggests that they effect the body. These details reveal the complexity of the entities and concepts that computational biologists are interested in. The additional complexities that arise from how we store data about these entities and use it in downstream analyses, is discussed in the next Chapter.

At the end of this chapter, I provide a short summary of the scientific model for how genotype and phenotype are linked.

2.1. Big questions: What is genetically determined, and how?

As humans, we are curious and want to understand ourselves. We want to know the answers to questions like: "Why are people the way we are?", "Which aspects of ourselves have we inherited?", and "What is fixed and what can be changed by the way we live our lives?" Looking at what we are made from - more specifically the DNA that can be found in each of our cells - has promised answers to some of these questions.

We ask so many questions, not only out of curiosity, but also in order to improve and control our lives and environment. This drive for control hasn't always been a good thing: in recent history, genetics has been used to justify extremely harmful and unethical racist eugenics policies. While eugenics is thankfully no longer in vogue, there is no guarantee that scientific knowledge will be used ethically. Modern-day genetics still raises concerns about which of our traits should be medicalised or pathologised: should we be looking for cures for autism if autistic people don't want them?

However, knowing more about ourselves clearly also has the capacity to be used for the good of all. By understanding how our bodies work, researchers seek to develop new, more effective, and kinder treatments for diseases. Alongside curiosity, these were my aims in seeking to explore the molecular link between genotype and phenotype.

2.1.1. History of inheritable traits

Humanity has been trying to answer the big questions long before we discovered DNA. The ancient theory of soft inheritance[11] said that people can pass on traits they gained during their lives, while 16th-century alchemists theorised that sperm contains tiny fully formed humans[12] (i.e. women didn't pass down anything).

Charles Darwin and racism

Darwin used his theory of natural selection to argue that women and non-white races were inferior to white men[10]. The full title of *On the Origin of Species* was *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*.

In 1859, Charles Darwin published his book *On the Origin of Species*[13], explaining his theory of natural selection: organisms compete for resources and not all can survive to reproduce, some organisms will have traits that increase their chances of this, and those that do pass on their traits to the next generation. The theory predicts and explains the gradual change of heritable characteristics over time: evolution. Darwin presented homologous anatomical structures, like the similarity between a bat's wing and a human hand, as evidence for evolution and shared ancestry.

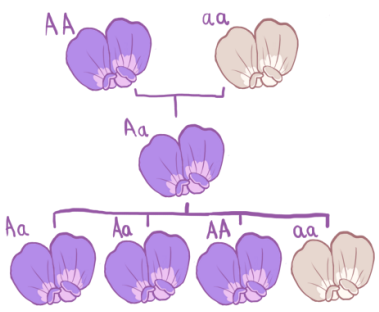


Fig. 2.1 An example of Mendel's experimental results, for white and purple pea flowers. He began with "pure line" pea plants (which always produced self-identical plants when self-pollinated). Crossing the "pure line" plants resulted in first generation offspring which always had purple flowers. When self-pollinated, these first generation plants created plants with purple and white flowers in a 3:1 ratio.

Homologous, orthologous, and analagous features.

Homology is similarity due to a shared evolutionary ancestry. This similarity can be between anatomy, or biological molecules and sequences (like DNA and proteins).

Homologous sequences are *orthologous* if they exist in different species, for example mice and humans have some orthologous genes, which are different versions of the same genes that perform the same function in the organism. This is in contrast to homology within species (paralogy) which occurs due to gene duplication, for example, humans have different versions of the histone gene.

Similar features and sequences which do not have shared evolutionary ancestry (i.e. which evolved independently), are *analagous*.

Inspired by Darwin[14], his contemporary Gregor Mendel's famous pea experiments[15] provided the earliest scientific basis for genetics through his experiments with independently inherited traits of peas (e.g. purple or white flowers, tall or shot plants, wrinkled or round seeds...). Importantly, he discovered rules of inheritance that indicated that offspring have combinations of discrete genetic material (rather than a blend), i.e. we don't see pink flowers when we cross purple and white flowered peas. He also showed that single traits (e.g. purple flowers) can actually be caused by different underlying genetics (see Fig. 2.1).

This concept was expanded by Wilhelm Johannsen, who coined the term "gene" as the name for the hidden material that caused the traits. Johnannsen's work also distinguished between "genotype" and "phenotype": *genotype* being the hidden (genetic) material that organisms have, and *phenotype* being the measurable trait[16]. His research showed that some phenotypes (e.g. seed size) could vary considerably even with genetically identical plants, due to their environment.

Ronald Fischer, racism and eugenics

Fischer has a legacy of scientific racism. For example, actively campaigning for sterilisation of a tenth of the population in the name of eugenics [17].

Phenotypes are not only strictly Mendelian, with a fixed number of "types", or continuous and without a genetic basis. Ronald Fischer showed that variation in continuous traits (such as height in humans) can be consistent with Mendelian inheritance if multiple genes contributed additively to the trait. Many traits are *complex* in this way, meaning that they are influenced by many different genetic factors, as well as the environment.

When Watson and Crick discovered the now familiar structure of DNA in 1953[18], we took a huge step towards being able to answer our big questions. We finally understood the molecular structure that underlies inherent traits. Then in 2003, with the completion of the Human Genome Project[19], it was possible to read the human version of this "code of life". Once researchers had access to the whole genetic code

for a person, they could set about trying to decode it, with the world hoping that this landmark would aid the search for treatments for diseases like cancers and Alzheimer's[20].

We now have thousands of human genomes to investigate. As I will explain, this enables the computational predictive methods that enable any characterisation of protein function for the majority of proteins, since the cost of alternative investigation of our complex network of proteins remains prohibitive.

James Watson and racism

Watson's has publicly asserted that he believes differences in average measured IQ between blacks and whites are due to genetic differences[21].

The promises of gene therapies and personalised medicine are now beginning to become reality, due in part to computational biology and bioinformatics breakthroughs. At the time of writing, there are eleven cell and gene therapies approved by the European Medicines Agency[22], which treat a variety of cancers, as well as Crohn's disease, and eye and cartilage problems. In addition, the first personalised genomic medicine chemotherapy treatment is now available on the NHS, for cancer patients with the allele of the DYPD gene that cause slower breakdown of chemotherapy toxins[23].

2.2. Biological molecules: DNA, RNA, Proteins and the central dogma of molecular biology.

Here I introduce the classes of biological molecules that are vital in our understanding of genetics: the nucleic acids (DNA and RNA), and their product: proteins.

2.2.1. DNA

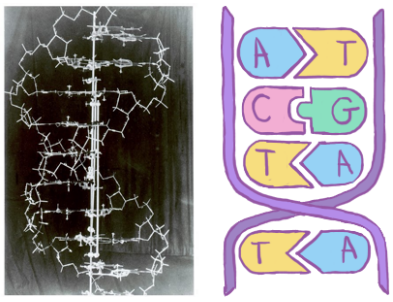


Fig. 2.2 Left: A photo of the original six-foot tall metal model of DNA made by Watson and Crick in 1953, alongside their discovery[18]. Image from the Cold Spring Harbor Archives[24]. Right: A cartoon representation of DNA, showing the concept of the complementary strand.

Most people recognise the double helix of deoxyribonucleic acid (DNA) shown in Fig. 2.2, it's a twisted ladder consisting of four nucleotides; adenine, cytosine, thymine, and guanine (A, C, T, G). Its the "code of life" that contains the instructions for making (almost) all of the things which make up our bodies and therefore, an obvious starting point for understanding how they work. A given nucleotide on one strand is always linked to its partner on the other strand - A with T, and G with C - which creates redundancy and a convenient copying mechanism. Lengths of DNA are measured in these base pairs (bp).

Human DNA is organised into chromosomes, we have two copies of each of our 23 nuclear chromosomes within (almost) every cell, as well as a varying number of copies of our mitochondrial chromosome in the cells which have mitochondria.

2.2.1.1. How DNA affects us: the central dogma of molecular biology

The way in which DNA affects the body can be understood through the *central dogma of molecular biology*, and in doing so we will also become acquainted with two more important biological molecules: RNA and proteins.

The central dogma of molecular biology can be paraphrased as "DNA makes RNA makes proteins". The idea is that as the "code of life", DNA contains the instructions for making RNA, which contains the instructions for making proteins, and proteins are the molecules which constitute and make up almost everything in our bodies.

The central dogma is a description of the process of *gene expression*. Gene expression has two parts: transcription ("DNA makes RNA") and translation ("RNA makes proteins"). By looking at these mechanisms we can gain an appreciation for the role of DNA in gene expression, and in phenotype.

***Almost* everything**

Proteins make up a lot of molecular machinery in the body as well as structures, however there are other actors, too, for example, functional RNA molecules (some of which are explained further down this page). Fats (which may or may not be synthesised using proteins) can also make up structural parts of the body such as cell membranes.

Gene expression = DNA + cellular environment

The environment, of the cell, and of the individual, and even of modifications to the DNA molecules, can affect the rate of gene expression. This is obvious when we think about the fact that the same DNA is present in their liver cells, skin cells, and neurons and realise that the difference in how these cells/tissues look and function is due to differences in which proteins are actually being made at any given time.

Superkingdoms: eukaryotes, archea, and bacteria

Living things are classified into taxonomies such that similar organisms are in the same group. At the highest level of these classifications there are three groups: eukaryotes (containing humans and all other multicellular organisms), bacteria, and archaea.

All three superkingdoms contain DNA, RNA and proteins, and are therefore a source of data and information about genetics.

2.2.1.2. "DNA makes RNA", a.k.a, transcription

RNA (or ribonucleic acid) was originally discovered alongside DNA as a nucleic acid, an acidic substance found in the nucleus of cells, hence it's similar name. It was later discovered that they are also found in bacterial and archeal cells (which don't have nuclei). In contrast to DNA, RNA is a single-stranded molecule, with the bases A, C, G and U (i.e. uracil instead of thymine), and with a different backbone (containing ribose, rather than dioxyribose). There are different forms of RNA which perform different functions. It is messenger RNA (mRNA) that is the intermediate product between DNA and Proteins.

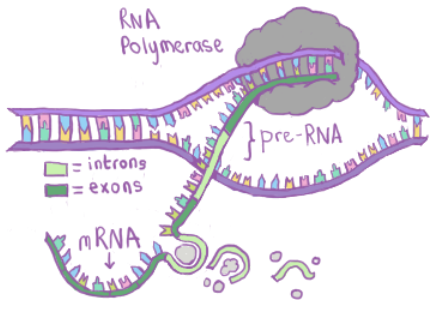


Fig. 2.3 An illustration of the transcription and splicing processes, showing the role of RNA polymerase in building RNA.

The process by which "DNA makes RNA" is known as transcription. The process happens to certain lengths of DNA, and these lengths of DNA are which are called *genes*. In humans, genes vary in length from hundreds to millions of base pairs.

The action of transcription is largely carried out by an enzyme called *RNA polymerase*, which binds to a promoter region of the DNA, close to but outside the gene. This region is only accessible in certain cellular conditions. A rough model is

Definining "genes"

that in conditions in which it is unfavourable for the gene to be transcribed, other molecules will block the promoter.

Genes are not a straightforward unit, as I describe in a bit more detail later.
Some sources use "gene" only to describe genes which encode for proteins, but I include genes which encode for RNA.

Fig. 2.3 illustrates the next part of this process. The RNA polymerase splits the DNA and adds complementary RNA nucleotides to the DNA, after which the RNA sugar backbone is formed. The RNA-DNA helix is then split, at which point we have what is known as precursor RNA or pre-RNA.

After transcription, the pre-RNA then undergoes *post-transcriptional modifications* such as *splicing*, where parts of the RNA (*introns*) are removed, leaving only *exons*. This can also bee seen in Fig. 2.3. Splicing is part of the final processing step to create the finished product: a mature RNA transcript.

During this step a gene could be transcribed into one of multiple transcripts, via a process known as *alternative splicing*. This can happen, for example, by skipping some of the exons during splicing. So, a more accurate statement is "DNA makes RNAs": there's not a one-to-one relationship between genes and transcripts, and therefore the same is true between genes and proteins. These different but related versions of proteins which come from the same gene are known as *protein isoforms*.

Other post-transcriptional modifications can also affect transcription and therefore human disease. Two key such modifications are RNA interference (RNAi) and RNA editing. RNAi relates to the degradation of mRNA before translation - this mechanism has been used to successfully create several drugs such as Givosiran which is used to treat rare metabolic diseases. Meanwhile RNA editing relates to all types of RNAs being edited after transcription - these modifications have also been linked to disease[25].

What is transcribed and how quickly is affected by many different kinds of proteins, as well as other molecules, through epigenetic modifications. Transcription factors are of particular note. These are proteins that bind to DNA close to or in a nearby *promoter* region, and either *activate* the gene (increasing it's rate of transcription), by for example recruiting RNA polymerase, or *repress* it (decrease it's rate of transcription). These transcription factors are in turn regulated by other transcription factors, which creates a network of gene regulation; a gene regulatory network (GRN).

Epigenetic modifications

Epigenetic modifications are persistent and heritable changes to DNA that do not affect the nucleotide, but can cause a difference in gene expression, such as histone modifications, chromatin remodelling, and DNA methylation. Epigenetic modifications can be responsible for phenotypes through altering gene expression.

2.2.1.3. "RNA makes Proteins", a.k.a. Translation

The second part of the central dogma is "RNA makes proteins" a.k.a. translation.

Proteins were discovered independently from DNA and RNA. They were named by Dutch chemist Gerardus Mulder in his 1839 paper[26], where he found that all proteins from animals and plants have more or less the same elemental makeup - approximately $C_{400}H_{620}N_{100}O_{120}$. This intriguing result bolstered research in this area, eventually resulting in our current understanding of proteins as biological macromolecules composed of amino acids.

The fact that the sequence of amino acids was a code which precisely determined the three-dimensional structure of proteins was discovered by Christian Anfinsen for which he was awarded the 1972 Nobel Prize. This was achieved through a series of experiments, which for example, showed that a protein could be reduced to a string of amino acids, and then in the right conditions could refold to the exact original protein structure[27,28].

Translation describes the process in which a string of amino acids is created based on the RNA sequence. Proteins are made of these amino acid strings (called *polypeptides*), and after translation, they will fold (potentially with the assistence of *chaperone* proteins) into the proteins usual globular three dimensional conformation.

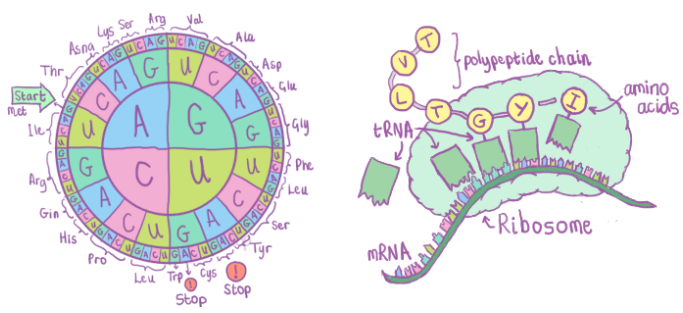


Fig. 2.4 Left: An amino acid wheel chart showing the mapping between nucleotide codons of RNA and amino acids. The chart is read from the inside out, for example UGA is a stop codon and UUG encodes for leucine "Leu"). Right: An illustration of the translation process, showing tRNAs delivering amino acids to the ribosome in order to build the polypeptide chain which makes up proteins.

Translation is mostly carried out by a large and complex piece of molecular machinery called the ribosome, which is made up of proteins and ribosomal RNA (rRNA). The ribosome reads and processes RNA in sets of three nucleotides at a time - these are called codons. Each codon is either a flag to the ribosome (e.g "stop here", "start here") or corresponds to an amino acid. Transfer RNA (tRNA) transports the amino acids to the ribosome where the polypeptide chain of amino acids is created.

Although we would expect $4^3 = 64$ permutations of nucleotides, there are only 21 different amino acids which can be incorporated into proteins in humans, so there is redundancy: multiple codons can encode for the same amino acids. We can see this clearly in the left part of Fig. 2.4, for example, UAA, UGA, and UAG are all read as stop codons. Different codons are not necessarily entirely equivalent, however, they can cause different patterns of expression due to being translated at different speeds.

Amino acid strings then fold reliably into 3-dimensional protein structures, sometimes with the help of other "chaperone" proteins. Despite all science has long known about the chemistry governing the folding of amino acid chains, we could not predict this very well until very recently.

After translation, and either before or after folding[30], proteins can also be subject to post translational modifications. These changes consist of chemicals bonding to the protein, which can for example change its function or structure, assist in folding, or target them for degradation.

The protein folding problem

Amino acid strings reliably fold into globular protein structures, but accurately predicting what form a sequence of amino acids will take as a protein is known as the protein folding problem and has been one of the grandest challenges in biology for over half a century. Recently, this challenge has been conquered by machine-learning techniques by DeepMind's AlphaFold algorithm[29].

The process of translating mRNAs can be repressed by very short RNAs called microRNAs (miRNAs) that can bind to more than half of mammalian mRNAs[31].

2.2.1.4. "... and proteins do everything."

The unwritten addendum implied by "DNA makes RNA makes proteins" is "...and proteins do (almost) everything". If DNA is the blueprint for life, then proteins are what make up life. They are the material building blocks of our bodies, and they also have a vast number of other functions: they can be enzymes catalysing reactions, hormones controlling metabolism, transporters for other proteins, signalling proteins, they might be transcription factors (controlling the expression of genes), or have many other functions.

In turn, these processes influence our phenotypes. A phenotype can be something like the level of a certain hormone in the bloodstream, so in a very simple case, a different amino acid in a hormone protein could cause the protein to be expressed or function differently. A phenotype could alternatively be something like height, which could have a number of genetic (and non-genetic) influences.

2.3. A closer look at DNA: Genomes, Genes, and Genetic Variation

Now that we have a basic overview of how DNA can influence phenotype, we can discuss the way that DNA is organised and categorised in a little more detail. We'll look from big (genomes) to small (single nucleotides).

2.3.1. Genomes

The genome is the full amount of DNA belonging to an organism. We can talk about the genome of an individual, or about the genome of an organism (e.g. the human genome). When we talk about an organism's genome, we are actually talking about an example genome for that organism: the organisms' *reference* genome. The reference genome does not belong to any individual organism, but instead is supposed to have the most common nucleotide at each DNA location.

Reference genomes allow us to make general statements about an organism (e.g. "the human genome is 3 billion base pairs long"), and also to make comparisons between organisms (e.g. "humans share 1% of their DNA with a banana"). We also discuss individuals' genomes in relation to the difference between the individual and the reference genome.

2.3.2. The exome and the proteome

Still thinking big, we have the *exome* and the *proteome*. Both of these refer to locations across the whole genome, but missing stretches in between. The exome describes the set of all exons (protein-coding nucleotides) across the genome. The proteome is generally used to mean the set of all proteins in an organism (which can be much larger than the set of genes due to alternate splicing), but it can also be used to describe the part of the genome relating to the set of protein sequences.

Humans and bananas

Humans share 50% of their *protein-coding* dna with bananas, but only 1% of their genome.

2.3.3. Genes

DNA is often considered at the level of the gene. Genes have been so central to the historical study of DNA (hence the name *gene*tics), and the gene-centric view of molecular biology continues to this day. For example, many researchers have favourite genes, which they primarily study, and understand the mechanisms of in detail. And for this reason, diseases and phenotypes are often attributed at the level of the gene, rather than at a more fine-grained level of the specific mutation.

Omics

As well as genomics (the science of genomes) and proteomics (proteins), there is also transcriptomics (transcripts), metabolomics (metabolites, e.g. sugars, lipids, etc). Together these research areas are known as *omics* and research which combines these fields is known as multi-omics.

As previously mentioned, in this thesis, I use genes to mean a stretches of DNA which can be transcribed into RNA (i.e. I include "RNA genes" in my definition). However, the seemingly simple definition hides a lot of complexities: due to their long history, the word "gene" has had different uses and meanings.

2.3.3.1. "A gene for X"

The word gene is often used as shorthand for "DNA that causes phenotypic differences" (for example in Richard Dawkin's best-seller "The Selfish Gene", and in news articles with titles of the form "Scientists have discovered a gene for..."). However, there are multiple reasons why this is an incomplete and in some ways outdated understanding. Single gene diseases do exist, however most of the time the same gene can make multiple different final proteins (isoforms)) which may not all cause phenotypic differences, the same protein can be involved in multiple different pathways and have multiple functions, and multiple proteins can contribute to one function. Genes are also not guaranteed to cause phenotypic differences, and are not the only sections/types of DNA which can influence phenotype. Another complication is that genes can overlap, meaning that a single nucleotide mutation could impact on multiple genes. And finally, sometimes entirely different genes can create identical proteins after translation.

The interaction between DNA, RNA and proteins, and the environment is also important to consider. Although DNA makes RNA makes proteins and proteins do pretty much everything in our bodies, which proteins are made and how they behave is highly dependent on the environment. The function of a gene might not be evident in some environments because the protein is never transcribed, or it may behave differently. Many traits may be mostly environmental.

2.3.3.2. Units of heritability

Genes are also often touted as a "unit of heritability/heredity", but this is similarly not always the case. DNA is more likely to be inherited together if it is close together on the chromosome, so generally we inherit whole copies of genes (and the regions around them) together - in fact usually we often inherit stretches of multiple genes together. Despite this, it is also possible that genes are not inherited "in one piece" with one whole copy from each parent.

2.3.4. Things that are not genes

There are many related concepts that contain the word gene simply because they are stretches of DNA, but that do not fit our definition. For example "jumping genes" and pseudogenes are both important parts of the human genome, which may effect phenotype, but not via proteins.

In addition, there are also stretches of DNA that are of interest in relation to genes: enhancers, silencers, insulators, and promoters. These are stretches of DNA that control the regulation of specific gene's transcription. Mutations in these stretches of DNA are often understood in relation to the genes that they regulate.

Transposable Elements

Transposable elements a.k.a. transposons or "jumping genes" are sections of DNA that move from one section of the genome to another. Their similarity with genes only extends as far as the fact that they are stretches of DNA. Despite making up 50% of the human genome, they only rarely overlap with protein-coding genes[32]. The vast majority of human transposons are *silent*, i.e. are incapable or prevented from moving.

Pseudogenes

Pseduogenes are segments of DNA that look like genes (have high homology to known genes), but are missing some functionality such that they do not encode proteins. Some pseudogenes cannot be transcribed (e.g. due to missing regulatory regions), and some are transcribed, but not translated (e.g. due to a premature stop codon). Although they were originally characterised as non-

2.3.5. Indels and Copy Number Variations

Smaller lengths of DNA are mostly of interest as *variants* where there are differences at those locations between individuals. These differences include insertions and deletions (*indels*) where sections of DNA, ranging from 1 to 10,000 base pairs, are inserted or deleted from the genome, and *copy number variations* where (generally larger) stretches of the genome are repeated. However, the most well-studied and common type of genetic variation between humans are single nucleotide polymorphisms (SNPs, pronounced "snips").

2.3.6. Single Nucleotide Polymorphisms

functional, they have been found to have biological roles, for example through being transcribed into functional RNAs[33].

A SNP is a location on an organism's genome where there are differences of a single nucleotide (A, C, T, G) between individuals. In some fields, these variations are only considered to be Single Nucleotide *Polymorphisms* if they are relatively commonly occurring in the population (at least 1%), while Single Nucleotide Variants (SNVs) can include both rare and common variants.

Variation at a location does not imply a disease-causing effect, many SNPs appear to be neutral. Much of the time, the aim of studying such variants is to determine which are which. This is often done through looking at their rarity, either in a specific human population (e.g. people with diabetes), the entire human population, or across the tree of life.

SNPs are defined by their location on a human reference genome, for example "chromosome 5, position 7870860" (often written `5:7870860`). An individual allele for a given SNP is defined as "wild" type if it matches the reference genome and "mutant" if it does not. The reference genome does not always have the most common allele at each location, although this is it's aim, so "wild" and "mutant" do not necessarily imply anything about rarity.

Alleles

The different forms that a variant can take in the population are called alleles. Alleles can be as big as different forms of a whole gene, or as small in length as an individual nucleotide.

If there are only two nucleotide possibilities for a SNP (e.g. it could be A or C at a given position), then it is called bi-allelic; the vast majority of SNPs are of this type. Multi-allelic SNPs such as tri-allelic SNPs (three choices, e.g. it could be A, T or C) are much rarer.

Since humans mostly have two copies of each chromosome (except for X/Y chromosomes in genetically male people, and people with chromosomal anomalies), an individual will usually have two alleles for each SNP. These may match (which we call homozygous) or not (heterozygous). Sometimes a disease-causing allele can cause problems even for heterozygotes, while in other cases a person needs two copies of the disease-causing allele in order for it to have an effect.

SNPs can occur either in coding or non-coding regions of the genome. In non-coding regions, SNPs can still affect gene expression, for example by altering a regulatory site. SNPs in coding regions have two types: synonymous or non-synonymous, based on whether they alter the amino acid sequence.

2.3.6.1. Non-synonymous SNVs

If a SNP alters the amino acid makeup of a protein, it is known as non-synonymous. Non-synonymous SNVs can cause either nonsense or missense mutations.

Nonsense mutations occur where the SNP substitution results in a stop codon (e.g. TAG) in an unusual position, which signals for a ribosome to stop translating RNA into a protein. This results in an incomplete and usually nonfunctional protein. The effect of a nonsense mutation would be more or less severe depending on the location of the new stop codon. For example, if it was close to the end of the protein, the protein may still be functional. Sufficiently incomplete proteins are usually destroyed by the cell.

On the other hand, missense mutations occur where the SNP substitution results in an amino acid substitution in the protein. Some amino acids can be substituted without causing any difference to the function of the protein, while others can severely impede the protein.

2.3.6.2. Synonymous SNVs

Synonymous SNVs occur where substituting the usual nucleotide with another results in the same amino acid. The resulting protein will have the exact same functionality. However, synonymous SNVs could still have an effect on high-level traits. One reason for this is that different nucleotides are translated at different speeds. This difference in translation speed has been shown to impact on both folding and abundance of proteins[34]. Furthermore, this mechanism has been shown to affect multiple phenotypes[35], for example lead to multidrug resistance of cancer cells[36].

2.4. Looking more closely at proteins: function, structure and classification

Just as DNA has been classified at different levels (SNP, gene, genome), so too have proteins. In contrast to DNA, what's interesting (and useful!) to know about proteins is their structure.

As mentioned earlier, translated strings of amino acids fold automatically (or sometimes with the help of other proteins) into the 3D structure of proteins. This structure defines what molecules (large and small) they can bind to, and this has huge consequences for their functionality in the body. Moreover, proteins have recognisable features of their structure which appear again and again. These features occur at different levels and sizes, and they allow us to learn about the evolution and functional similarity of proteins.

2.4.1. Protein structure: Primary, Secondary, Tertiary, and Quaternary

Proteins are described and classified in terms of the their primary, secondary, tertiary, and quaternary structure.

The primary structure is simply the amino acid makeup of the protein, which describes the protein's chemical makeup, but not it's three-dimensional structure. These amino acid strings tend to form into a small number of familiar (secondary) three-dimensional structures, for example beta strands, beta sheets, and alpha helices. At the next level, the tertiary structure describes combinations of secondary structures, for example a TIM barrels (a torroidal structure made up of alpha helices and beta strands). At this level, similar structures do not imply an evolutionary or functional similarity.

2.4.1.1. Quaternary structures: protein domains

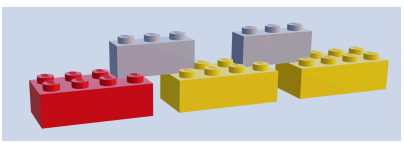


Fig. 2.5 An illustration of the lego analogy for protein domains. Coloured bricks represent protein domains - colour represents a specific protein domain type, while thin grey bricks represent polypeptide linkers which link domains. Image created using mecabricks[37]

The quaternary structures of proteins - protein domains - have proved particularly interesting for research. A simple and oft-used metaphor is to think of protein domains as lego building blocks (Fig. 2.5) which can be linked by polypeptide chains to make up a protein. These polypeptide chains (known as linkers) are often inflexible, in order to allow only one conformation of the protein. Small and simple proteins often consist of just one domain, while bigger proteins can contain many domains. An individual domain can be found in many different proteins, and multiple times in the same protein.

Protein domains are interesting because they are highly conserved in evolution, and are thought of as units of function, evolution, and/or structure. The functions of proteins, at both low-level (e.g. "calcium signalling protein") and high-level (e.g.

Conserved sequences

involved in “liver disease”), are costly and difficult to discern, so there are many proteins about which little is known. For this reason, often proteins are classified according to their similarity to proteins about which functions are known, for example those containing the same protein domains.

> Sequences are highly *conserved* if identical or homologous sequences are frequently found in other species. The underlying assumption is that the sequence must be important for organisms survival if they are present in all species.

2.4.1.2. Disorder

While protein domains always exist in one conformation, this is not the case for proteins as a whole. One reason for this is that not all linkers are rigid. Flexible regions of proteins which allow for various conformations are referred to as *disordered*.

And disordered regions are not only relegated to linkers between domains. Proteins can be constituted entirely of disordered regions, or may have large disordered regions.

Such *intrinsically disordered* proteins can exist in a number of conformations, rather than one fixed structure. On some occasions, the disordered regions are known to be functional, while on others, proteins may be non-functional until they bind with another macromolecule which forces them into a fixed conformation.

2.4.1.3. Classifying proteins by domain: families and superfamilies

Proteins with known domain structure can be grouped together based on their structural similarities, based on the consideration of the protein’s constituent domains into *families*, *superfamilies* and *folds*. Proteins are classified into *families* representing the most similar proteins, which share a clear evolutionary relationship, while *superfamilies* represent less close evolutionary relationships, and *folds* represent the same secondary structure. This protein classification task, while aided by automation, was carried out largely by manual visual inspection[38].

2.5. Phenotype

2.5.1. What is phenotype?

Phenotypes are observable traits, which can range from neutral (like height, skin colour, or eye colour) to disabling (e.g. chronic fatigue syndrome) or life-threatening (e.g. cancers), to very specific measurements (e.g. level of calcium in blood). Since phenotypes can have various levels of specificity, they can also be hierarchical, an individual could display “abnormal muscle morphology”, or more specifically “facial muscle atrophy”, which means we have to decide at what level to record phenotypes. Human phenotype information is private information, and some phenotypes are not easy to measure, so information about human phenotypes is not always easy to access.

2.5.2. How do proteins influence phenotype?

The easiest phenotypes to understand genetically are Mendelian. In Mendelian phenotypes, a single mutation is responsible for a phenotype, and we can assume that the mutation changes, reduces, or stops entirely the functionality of the protein, and that this protein is the main actor involved in the trait. An example of this in humans is the OPN1MW gene which encodes for green-light absorbing pigment necessary to create green light absorbing cones in the eye: the allele that causes a non-functional OPN1MW gene therefore causes red-green colourblindness.

The way in which Mendelian genetics affect a phenotype can vary. In humans, for a SNP with two alleles, there are three possible *calls*: homozygous wild type (two copies of the most common allele), heterozygous (one copy of the most common allele and one copy of the rarer allele), and homozygous mutant (two copies of the rarer allele). Sometimes having one copy of the rarer allele is enough to cause a phenotype, but sometimes two copies are required. Not all SNPs are disease-causing at all, i.e. have any disease-causing combinations of alleles.

As well as mutations, phenotypes can be caused by chromosomal abnormalities (extra or missing sections of chromosomes). In this case, the mechanism is the increased or decreased gene expression of the affected section of the chromosome which is influencing phenotypic differences.

Proteins can affect the same phenotype indirectly, through protein-protein interaction networks, through interaction with the metabolism (the body’s creation of small chemicals, like sugars, fatty acids, and vitamins), and through interaction with the environment of the cell. The environment of the cell is of course in turn influenced from the human-scale environment: what we eat, whether we smoke, the air we breathe, and our body’s response to outside stimuli.

2.5.2.1. Limits

For many disease phenotypes (e.g. Breast Cancer, Asbestosis), a genetic mutation might predict an increased probability of having the phenotype, given similar environmental conditions. And there are phenotypes which may not be linked to genetic variation at all, but may be entirely influenced by the environment: for example medical conditions that are the result of poisoning. In these cases, we might imagine that there is a mutation that humans could have that would prevent or reduce the poison reaction, but since no one has it, we can’t study this by looking at human mutations.

> **Social Constructs**
>
> A social construct is an idea or concept that exists because it has been created and agreed upon by society.

To get a little philosophical (metaphysical) for just a paragraph, some phenotypes may not even exist. That is, they might not be *natural* categories such that there is a straight-forward and physical thing that decides membership to the category[39]. As an example, consider an imaginary poorly-understood syndrome, it might be diagnosed if you have some of a list of symptoms, but the syndrome might actually be four separate diseases with four totally separate causes and the treatments might only work for one of these diseases. Some phenotypes might even be *social constructs*; there is a long-running debate among psychologists about whether some mental health conditions and other psychological and behavioural concepts are socially constructed[40,41]. If phenotypes are not based in the physical, then we will likely have difficulty accurately predicting them from genetics.

2.5.2.2. Ethical considerations

Aside from the fact that predicting non-physical concepts is difficult, there are also ethical considerations in trying to predict socially constructed phenotypes. If we try to predict sexual orientation from genetics [42], then we might turn out to be measuring something else which indirectly influences sexual orientation, for example a protein that influences how open people are to new experiences, or something that in turn influences that. And in trying to predict intelligence from genetics, for example, we are likely finding associations between variables like how much you have practiced IQ tests or whether you are in the same cultural group as those that created them[43], reinforcing racist ideas[44].

Even if all phenotypes were natural concepts, predicting the genetic basis of some phenotypes could be harmful[45], for example finding a gay gene could be motivated by, or lead to a search for “treatments” to “cure” homosexuality even if it did have a physical basis.

Physical measurements can also be problematic for similar reasons. Take measurements of facial features for example: this brings to mind the image of nazis measuring skulls. Where physical measurements are proxies for measuring the social construct of race, these kinds of phenotypes can be similarly worrying. Facial recognition technology[46] is often criticised on this basis[47].

It is for these reasons, that the majority of modern concepts of phenotype are based in medical concepts, where looking for a link between genotype and phenotype can have a potential life-saving or life-improving benefit. This scenario still comes with serious ethical considerations, however. Many disabled people do not want cures for their disabilities[48], and people are also worried that the development of genetic screenings for disabilities will effectively result in a genocide of disabled people[49].

Another concern is that people may accidentally find out about phenotypes that they are predisposed to that they do not wish to know about. This is particularly worrying if there are not any existing preventative/proactive measures to avoid a future diagnosis, and if they are not be able to access genetic counselling. For example, 23andMe have a system whereby you must opt-in to viewing reports about your health for some illnesses.

2.5.3. The future computational biologists want

The eventual destination of this field is a full understanding of how our individual genomes and their interaction with the environment affects us. With this understanding, we would anticipate a much wider application of both personalised medicine and gene therapies. These therapies are not yet a common occurrence: the eleven approved cell and gene therapies come from a pool of such 500 clinical trials[22].

Perhaps it makes sense that we are not finding drug targets quickly, as we still don't know the functionality of approximately 20% of human genes[50]. And genes are only a small part (1-2%) of our DNA[51], and the part we understand best. Beyond our DNA, there are many other aspects of our cellular and social environments that will have an effect on which parts of our genes are being actively used, and how much. This section provides an overview of our current scientific model for how DNA affects phenotype, so that we can identify the sources of information that we do have and can make use of.

Despite what we don't know, this is also the moment when we have a unique hope to unravel some of these mysteries. We have openly available, expertly curated, databases containing the great collective knowledge of many experiments about our DNA, how it is being used, and what traits it affects. These databases are being filled at an alarming speed by researchers around the world with the advent of new technologies. Perhaps it is now possible to begin to synthesise some of this collective knowledge into a fuller understanding of complex traits.

2.6. Summary: how genotype and phenotype are linked

The purpose of this introductory chapter was to provide an overview of how we think phenotype arises from genotype. It's also to explain why it's a hard problem!

As we've seen in this chapter, there are many kinds of genetic variation which can influence phenotype. I will summarise the link between genotype and phenotype for the simplest and smallest kind of genetic variation: the SNP.

SNPs can exist anywhere on the genome: in the exome (protein-coding region), or outside of it. If the SNP is in a coding region, it may encode for multiple different protein isoforms. For each protein or isoform, a mutation could change the structure of the protein at one location, cut it short at that point, or have no effect on the structure. If the SNP is non-synonymous for the protein (affects protein structure), then it may fall in a disordered region of a protein (leaving us without structural - and therefore often functional information), and we may not know in what circumstances and cells that protein is transcribed. In addition, the SNP may affect phenotype differently with homozygous or heterozygous calls, and the protein may affect phenotype by influencing a network of other proteins, or the protein may exist as a redundant part of a pathway which will only affect phenotype if three other SNPs have specific calls. Even after all this, the presentation of many phenotypes can depend heavily on the environment, the age of the individual, or artefacts of measurement. The mechanisms will be different for each phenotype, and we can expect some phenotypes to be impossible to predict from genotype.

Given all this complexity, it may seem no wonder that phenotype prediction remains a challenge[5]. However, it is a challenge which we have to meet if we want to understand the genetic mechanism behind diseases, in order to create more easier and more accurate genetic diagnoses, and to create new treatments. In the next chapter, I describe the diverse information about biological entities that can be measured, including gene and protein sequence, protein structure, variant frequencies and functions, and gene expression, and how it is currently used.

3. How the link between genotype and phenotype is researched

We have just introduced the biological mechanisms linking genotype and phenotype. Next, we will discuss the details of how this connection is studied, including how data about DNA and RNA is captured, organised and stored, and how this data is used in computational biology research.

This chapter begins with a short description of popular sequencing technologies, as this is relevant to both DNA and RNA.

Then in the second section, we will retrace the steps we took in the previous chapter, looking again at DNA, RNA, proteins, and phenotypes in turn, but this time considering the data gathered about each of these entities, and the data gathered about the connections between them. Sprinkled throughout the chapter, as they become relevant, I describe some specific examples of resources and tools used in bioinformatics and computational biology that are relevant to this thesis.

Two types of tools and resources, however, have their own sections. The first are biological ontologies, which are efforts to unify some of the information gained in the experiments just described in earlier parts of this chapter. Secondly, predictive computational biology methods and the ecosystem of competitions that are often used to validate them are also described separately in section 3.4. In this section, I also explain my contribution to the update to the SUPERFAMILY resource[3].

I then describe some of the potential sources of bias in the data and tools used throughout this thesis, followed by my contribution to a project designed to counter some of these issues, the Proteome Quality Index (PQI)[2].

Finally, I summarise the data we currently have (and don't have) on the link between genotype and phenotype.

> 💡 **Contributions in this Chapter**
>
> This chapter primarily summarises the work of others, but it also contains my contributions to the following collaborative projects:
>
> - 2014 Superfamily update paper[3]
> - Added some cyanobacteria genomes to the resource
> - Contributed to paper-writing/editing
> - The Proteome Quality Index paper[2]
> - Contributed to development of metrics for measuring proteome quality
> - Contributed to paper-writing/editing

3.1. Sequencing and microarrays

Sequencing and microarrays are how we get measurements of DNA and RNA. We measure DNA so that we can understand what organisms genetic material is capable of doing: and understand what the differences between different species and individuals are. These measures of DNA can tell us (among other things) what proteins it is possible to make. If we think of genes as a collection of blueprints, then one major reason that we measure RNA to tell us how much each blueprint is in production.

3.1.1. Sequencing

Sequencing technologies are used to read strings of DNA or RNA: this can be done *de novo*, i.e. even when we don't know the sequences ahead of time. At one time, we might wish to sequence anything from one gene to the entire genome. No sequencing technology can read whole chromosomes end to end, however, all work by reading shorter lengths of DNA (called *reads*).

In most sequencing technologies (e.g. Sanger, Illumina), in order for the different nucleotides to be detected (by human sight or using a sensor), DNA is first prepared such that different nucleotides bond to different visible markers, e.g. different coloured dyes or fluorescent markers.

From the late 1970's until the mid 2000s, *Sanger sequencing* was the most popular sequencing technology, although it underwent various improvements over this timescale. In Sanger sequencing (and other first-generation methods), reads of around 800bp are sequenced, one at a time, using electrophoresis. The human genome project sequenced the first human genome using this method[19], and it's still used in some circumstances, for example validating next generation sequencing.

Second, or *next generation sequencing* (NGS), also referred to as high-throughput sequencing, is a catch-all term for the faster and cheaper sequencing technologies which replaced the previously used Sanger sequencing. A feature that is common to NGS methods is that many shorter reads (around 100bp, exact numbers depending on the specific technology) are sequenced in parallel. The process is massively parallel: millions to billions of short sequences can be read at a time. This is a huge factor in making NGS much faster (and therefore cheaper) than Sanger sequencing. In turn, this speed and cheapness means that more repeats can be sequenced, increasing the overall accuracy of NGS over Sanger (despite the accuracy of each individual read being generally lower).

> **Whole genome sequencing**
>
> Whole genome sequencing (WGS) is the process of sequencing an individual's entire genome: across all chromosomes (protein-coding and non-coding DNA) and the mitochondria. This can be achieved with different sequencing techologies.

> **Elecrophoresis**
>
> Electrophoresis is a laboratory technique in which molecules are separated based on their size by applying an electric current to molecules. This forces them to travel through a small capillary tube, or through a gel matrix.
>
> In DNA capillary electrophoresis, DNA is read via exciting flurescent markers with lasers and detecting the produced light (this is how automated Sanger sequencing works).
>
> In manual gel electrophoresis, the DNA is prepared in advance so that there are four samples of DNA each containing pieces of DNA of varying length, such that each sample has a different nucleotide at the end of each piece. The length of those pieces of DNA in nucleotides is determined by how far the DNA lengths can move through the gel.

NGS can be used for sequencing either DNA or RNA (known as RNA-seq when applied to the whole transcriptome). While (NGS) DNA-sequencing and RNA-seq can use the same underlying NGS technologies, there exist some differences, e.g. RNA is reverse-transcribed into strands of complementary DNA, before being sequenced, since sequencing DNA is currently easier than sequencing RNA.

There are now also third generation sequencing technologies that allow much longer reads to be sequenced, e.g. nanopore technology.

3.1.1.1. Capped Analysis of Gene Expression

Capped Analysis of Gene Expression (CAGE) is a NGS transcript expression technique which measures very small (27 nucleotide) segments (called *tags*) from the start (5′ end) of mRNA. These tags are mapped to genes based on their distance to the gene in bp. The upside of this approach is that these short tags can be used to identify the transcription start sites (TSS) of RNA transcripts. The downside is that it can only be used to measure mRNA (mature messenger RNA). CAGE is used extensively in the FANTOM research projects, such as FANTOM5 whose data is used in Section 5 and Section 7.

> **Nucleic acid directionality**
>
> DNA and RNA both have a sugar backbone with five carbon atoms, which are numbered from one to five according to chemistry naming convention such that one end of the backbone always terminates with the 5th carbon of the ring (the 5′ pronounced "five prime") end, and the other terminates with the third carbon of the ring (the 3′, pronounced "three prime") end. This leads DNA and RNA to have different chemical properties at each end, and means

3.1.2. Alignment and assembly

> that in the body DNA and RNA can only be produced in one direction: 5′ to 3′. This is also the convention for how we write DNA and RNA sequences.

Whichever technology is used, DNA and RNA is sequenced in small sections. This means that reads must then be *aligned* to an existing sequence (e.g. reference genome, known gene, or transcript), to allows us to know where on the genome (which chromosome and position on that chromosome) the read came from.

If an existing sequence does not yet exist, we say that we are sequencing *de novo*. In this case, reads are aligned with one another, as illustrated in Fig. 3.1 so that they can be *assembled* into a new sequence.

In both cases, alignment requires the reads to overlap, so longer and more numerous reads make these tasks easier.

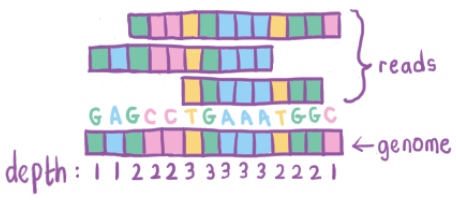


Fig. 3.1 Image illustrating how reads of DNA are aligned with one anther to assemble genomes *de novo*.

The current estimate for raw sequencing inaccuracy of an individual NGS read is around 0.24%[52], meaning that on average one base pair will be incorrect for a 500bp read. Multiple repeats are therefore required to obtain a more accurate measurement of the assembled sequence, which is further necessary since there are many repeated sequences (perhaps over two thirds of the human genome[53]). The depth for a nucleotide is the number of reads that overlap that nucleotide. Similarly, the average depth of a sequence can be calculated.

After assembly, even in the most complete genomes, we are still left with some sequences that could not be placed, and some parts of the genome that we still don't know about.

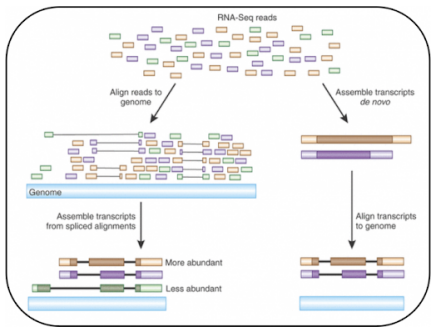


Fig. 3.2 Image showing how RNA-Seq reads are mapped to the genome (image from Advancing RNA-Seq Analysis [54]).

RNA-seq is used much less often for de novo sequencing, and is generally mapped to a reference sequence.

Fig. 3.2 shows how alignment and assembly are used in the context of RNA sequencing.

3.1.3. Microarrays

Through the 1970s into the early 2000s, DNA arrays/microarrays developed alongside sequencing as a way of measuring the presence of previously sequenced DNA in new samples. These arrays contain pre-chosen fragments of DNA (probes) arranged in spots, with each spot containing many copies of the probe, on a solid surface, e.g. a glass, silicon or plastic chip. The probes consist of single strands of DNA, and arrays operate on the principle that the complementary DNA from the sample will bind tightly to it.

These arrays were originally macro-sized, one of the first being 26 × 38 cm and containing 144 probes[55], but are now on small chips, which can contain up to millions of probes. Different chips will contain different probes and therefore measure the presence of different sequences.

Arrays were extremely popular for measuring gene expression, but this technology has largely been superseded by the more accurate and comprehensive RNA-seq. DNA Microarrays are still commonly used by companies like 23andMe for genotyping an individual.

> **Genotyping**
>
> Genotyping is determining DNA alleles at specific locations. This is usually done using DNA microarrays.

3.2. From genotype to phenotype: what is measured

This subsection now delves into the details of the data that is collected, looking in turn at measurements of DNA, RNA, proteins, and phenotypes. This is to illustrate what kinds of data exist within the databases of the bioinformatics landscape, as well as some of the subtle issues that arise when using and linking them

3.2.1. DNA

In the previous Chapter, DNA was described, as well as the working scientific model of it's link to phenotype. In this section, the focus is on the details of how this is measured and stored: and how these details impact computational biology research. Once again, these details are introduced from big to small scale, beginning with whole genomes and moving through to individual SNPs.

3.2.1.1. Whole genomes

Whole genome sequencing (WGS) is the sequencing of all the genetic material of an organism, whether or not it is transcribed into RNA, or translated into protein. In humans, this includes all chromosomal and mitochondrial DNAs. Whole genomes are sequenced and assembled as previously described.

Whole genomes for different species can be compared to one another to give us insight about health and evolution. This work can be comparisons within one species in which individuals of that species are compared, or comparisons across several species. This field is known as comparative genomics. Genomes from different species are stored in databases such as the University of California Santa Cruz (UCSC) Genome Browser database[57], the US National Centre for Biotechnology Information (NCBI) Genome Sequence database[58], or the European Bioinformatics Institute's (EBI) Ensembl Genome database[59].

> **The *whole* genome?**
>
> In practice, almost complete genomes are also referred to as whole genomes, particularly for more complex genomes. Even the human genome still a small outstanding amount of unassembled DNA[56] - satellite DNA which is thought to be part of the structure of chromosomes.

3.2.1.2. The human reference genome

As previously mentioned, reference genomes are designed to represent whole organisms: these genomes aim to have the most common allele at any given nucleotide, and are then annotated at positions where individuals differ.

Builds and patches: As more whole genomes for an organism are sequenced, more information comes to light about the nature of the genome. For example, some locations are revealed to be likely sequencing artifacts. New major versions of genomes are released every few years to fix these changes. These versions are called *builds*. Between builds and patches, sequences may be added, removed, or moved to different locations on chromosomes.

Different versions of the builds are released by the Genome Reference Consortium (GRC) and the University of California Santa Cruz (UCSC) Genomics Institute. Table 3.1 shows information about the most recent human reference builds, taken from the UCSC website[60]. For example, `hg19` (**h**uman **g**enome build **19**), is largely equivalent to `GRCh37` (**G**enome **R**eference **C**onsortium **h**uman build **37**). These are generally used interchangeably by researchers, but there are some differences between them. This includes formatting differences (storing chromosome as integers rather than strings like `chr1`), the inclusion of mitochondrial DNA, as well as small numbers of differences of the locations of some variants on some chromosomes [61].

Release name	UCSC	Release date
GRCh38	hg38	December 2013
GRCh37	hg19	February 2009
NCBI Build 36.1	hg18	March 2006

Table 3.1 Table showing human reference genome builds

3.2.1.3. Genes

Once scientists have an assembled genome, genes are identified within them. This is part of the process of structural annotation.

Like whole genomes, the sequences and positions of genes relative to the reference genome are stored in databases. Again these are part of the UCSC, NCBI and EMBL-EBI ecosystem and these resources are vital to bioinformatics. However, having multiple sources of gene information does cause some ambiguities when there are disagreements between databases. They can sometimes disagree on fundamental details such as locations of genes or the number of genes in an organism[63] since the different databases take different decisions about how to store information.

Each of these databases also have their own identifiers and these names and symbols can change over time. For this reason, it can sometimes be difficult to map between identifiers from different sources.

Due to the history of the gene, and the amount of information that researchers have collected through gene knockouts and gene expression experiments, it is at the level of the gene that a lot of mappings about function take place. This includes, for example, information about a gene's involvement in a gene regulatory network or in a biological pathway, and information about gene function according to observational studies.

Even in the most-studied genomes, there are many genes for which databases contain sequence information, but no functional information. This is due to the low cost in sequencing experiments in comparison to the expense of knock-out or other function-determining experiments, and the inequality of studied proteins/genes. This missing functional information is not likely to appear soon, without some sort of revolution in funding priorities or technology.

Structural/gene annotation

Genome annotation originally focused solely on discovering the locations of protein-coding gene. While gene discovery remains a big focus, structural annotation now also includes the prediction of other features such as non-coding genes and transcription factor binding- or DNA methylation sites. These features are identified by complex predictive algorithms, which can be *ab initio* (often HMM-based), homology-based, or a combination[62].

Intron-exon distances and base and codon distributions, among other data are as predictive features. Observed start (i.e. the amino acid methionine - `ATG`) and stop (`TAA`, `TGA`, or `TAG`) codons are useful in validating structural annotation algorithms.

Persistent identifiers

Persistent identifiers are long-lasting digital reference to entities[64]. Gene names can change; scientists might agree to change them because Excel keeps converting them to dates[65] or because what were thought to be two genes turns out to be one. Gene identifiers should be unique and persistent over time, for example between genome builds and as more is learned about their function, but they can still be merged or retired.

3.2.1.4. Variants

Information about which variants individuals have comes from either genotype or whole genome sequencing (WGS) data.

When WGS is carried out for an organism that has already been sequenced, the sequence data is mapped to the organism's reference genome. When cohorts have their whole genomes sequenced, this allows information from WGS data to be compressed into Variant Call Format (VCF) files, which stores only the allele calls for locations where there is variation in the population. This provides a more detailed and more accurate alternative to genotyping data.

Some variant data is owned by private companies, such as 23andMe. Databases like dbSNP[66], clinVar, and SNPedia contain information about the location of these variants, their possible alleles, their frequency in populations, their functions, and associated phenotypes.

The largest SNP database - NCBI's dbSNP[66] - contains information from ten organisms (including human) and has information on indels and short tandem repeats in addition to SNPs. Anyone can submit their findings about variants to dbSNP, and they must indicate what sort of evidence they have for the association. dbSNP gives SNPs unique identifiers (Reference SNP cluster IDs, a.k.a. RSIDs) of the form `rs###`, which are used by many other resources.

3.2.2. RNA

For RNA there are three main types of data: sequence (and mappings), structure, and gene expression data.

3.2.2.1. RNA Sequence and Structure

The sequences of RNA (including miRNAs, tRNAs, rRNAs, etc), and their locations relative to reference geneomes are stored in databases such as Ensembl. For mRNAs that encode for proteins, this also enables mappings between transcript IDs, gene IDs, and protein IDs, and again these are integrated with previously mentioned gene databases.

Functional RNA has structure with recurring motifs similar to those of proteins. There are also databases of functional RNA structure[67,68] (similar to those for proteins), but those for RNA are at an earlier stage.

3.2.2.2. Gene Expression

As I've already explained RNA abundance in samples can be measured through RNA microarrays and RNA-Seq, and recently, RNA-Seq has been much more popular. Measures of mRNA abundance (i.e. gene expression data) are generally the most popular measures of translation (which it is a proxy for), compared to the more direct measurement of protein abundance for example.

Housekeeping genes

Genes whose core functionality are to perform basic cell maintainance are known as *housekeeping genes*.

The popularity of RNA-Seq therefore means it is the most diverse data currently in databases, which makes it ideal for informing us how DNA's blueprints are being used in different scenarios. Together with mappings, this data is used to understand the function of genes, to identify housekeeping genes, to re-engineer gene regulatory networks, and more - knowledge about DNA function that wouldn't be possible to glean

without measuring RNA. Like other bioinformatics data, gene expression data is also available in databases such as the EBI's Gene Expression Atlas (GxA)[69] and Single Cell Expression Atlas[70] (for bulk and single cell gene expression, respectively).

Differential expression versus baseline experiments: In order to reveal genes that are involved in specific diseases or functions, a popular type of gene expression experiment involves comparing the gene expression between two types of samples, for example between healthy samples and cancerous samples. This is known as a *differential* expression experiment. In contrast, a *baseline* experiment would measure the amount of expression in a range of more regular circumstances, aiming to characterise the range of expected expression in healthy individuals.

Bulk versus Single-cell RNA-Seq

Most RNA-Seq experiments sequence RNA from millions of cells at a time: this is bulk RNA-seq. Bulk RNA-Seq may take place for many cells of the same type or for a collection of different cell types (e.g. a tissue). Single-cell RNA-Seq (scRNA-seq) allows RNA to be sequenced from a single cell and is becoming increasingly widespread[71].

3.2.2.3. RNA-Seq bioinformatics pipeline

RNA-Seq data *counts* the number of times a sequence matching that gene or transcript has been sequenced. The amount of RNA from a particular transcript that is found in a sample in a given experiment is dependent on the sequencing depth and the transcript length. The rate of transcription is dependent on time of day, tissue, location, cell type, etc, so measures of RNA are also dependent on all of these conditions: this can make RNA measurements difficult to compare between experiments. To make matters worse, RNA-Seq and RNA microarray measurements are also sensitive to differences in laboratory conditions and experimental design, creating artefacts in the resulting data known as *batch effects*. Taken together, these things mean that there is a substantial data preparation pipeline for RNA-Seq data.

Quality Control: Before RNA-Seq data undergoes alignment, it undergoes quality control. This involves comparing sequencing parameters to a data set of known accuracy[72] and is usually done as part of the sequencing.

Normalisation - within-sample normalisation: TPM and FPKM: Within-sample normalisation methods are designed to account for sequencing depth and transcript length so that gene expression values from the same sample (e.g. different replicates) can be more easily compared. Longer genes will have more reads mapped to them for an equal level of expression, so RNA-seq will report more counts. Similarly, without normalising, samples with greater sequencing depth will have higher counts for an equal level of expression.

RPKM/FPKM (Reads/Fragments Per Kilobase Million) and TPM (Tags Per Million) are the three major normalisation techniques used for this purpose. In RPKM and FPKM, counts are first normalised for sequencing depth, and then for gene length. This means that they are suitable for comparing within a sample (e.g. between replicates). TPM, however performs the same steps in the opposite order, which has the desirable effect of ensuring that columns corresponding to TPM normalised samples sum to the same number. This means that TPM gives us a measure of relative abundance; the proportion of counts are from each gene can be compared across samples. For this reason, TPM is now generally preferred over RPKM/FPKM[73,74].

Normalisation - between-sample: While TPM gives us a measure of relative abundance, it does not give us a measure of absolute abundance. One outlying gene which is highly expressed will have the effect of making all other genes look relatively less expressed. This might be expected to occur, particularly when samples are under different conditions (e.g. disease/treatment). Between-sample normalisation methods are designed to counter this issue, and enable researchers to compare different samples.

These methods adjust counts to reduce the impact of outlying expression values. Examples include scale normalisation methods like TMM (used in edgeR[75]), the Log Geometric Mean (used in DESeq2[75,76]), and quantile normalisation (giving samples the same distribution of counts).

3.2.3. Proteins

Similar to RNA, proteins also have (amino acid) sequence data, mappings to genes and transcripts, structure data and protein abundance data. While for RNA, abundance (gene expression) data is the most popular type, for proteins, it is structure data, and from this structural information, there is a very detailed system of protein classification.

Proteins are where the history of bioinformatics databases that any researcher can contribute to began. Margaret Dayhoff created the first bioinformatics database in 1969, to store protein structures imaged using X-ray crystallography, related to her publication of Atlas of Protein Sequence and Structure[77]. The Uniprot[78] database of protein sequence and functional information is the heir to this early database, it contains information about protein sequence, domain architecture, and function.

3.2.3.1. Protein Sequence

Just as DNA and RNA can be sequenced in nucleic acids, proteins can be sequenced by their amino acids, although the technology behind doing this is quite different (e.g. using mass spectrometry is the most common way). This is often done for a small part of a protein, to allow it to be matched to the expected amino acid sequence based on gene or transcript sequences. This is how mappings from protein IDs to gene IDs and transcript IDs are available through databases (e.g. Ensembl).

Protein sequencing is also used to characterise protein's post-translational modifications.

3.2.3.2. Protein Abundance

The abundance of proteins in a sample can be measured through various quantitative proteomics techniques. These are carried out using electrophoresis, or mass spectrometry, for example. Similar to gene expression, this technique is often used to compare between two different samples (e.g. disease and control groups). Data from such experiments are also available in databases[79,80].

Mass spectrometry

Mass spectrometry is the process of ionising a sample and accelerating it through an electric or magnetic field to deduce it's mass-to-charge ratio.

3.2.3.3. Protein Structure

The Protein DataBank (PDB)[84] was established not long after Dayhoff's database, it contains three dimensional protein structures, typically obtained using X-ray Crystallography or NMR spectroscopy. The PDB continues to be well-used and updated, at the time of writing holding structures of 148,827 biological molecules. These structures are used for protein classification, and for Molecular Dynamics simulations (simulating the physical interactions of molecules).

Gene Expression and Protein Abundance data

It's interesting to note that gene expression levels (from RNA-Seq and microarray data) are not necessarily strongly correlated with protein abundance; this has been found in mice[81], yeast[82], and human[83].

In human, Spearman correlations between protein abundance and gene expression levels vary between 0.36 and 0.50, depending on tissue, meaning that they are only weakly or moderately correlated[83].

3.2.4. Phenotypes

As described in Section 2.5.1, most phenotypes that are studied today are based in medicine: this can range from the results of a blood test, to the presence of a disease diagnosis. Neutral traits like height, eye colour, baldness, etc, are also sometimes measured.

Phenotypic traits can be measured in a huge variety of ways, depending on the phenotype. One important type is data collected via survey or interview, where participants self-identify as having certain illnesses, or symptoms. This type of data can suffer from biases due to what people feel comfortable answering[85,86].

An additional challenge is that phenotype data must be connected to genotype data in order to be useful for validating genotype-to-phenotype predictions/ Due to the sensitivity of this kind of information, there are a limited number of these kinds of data sets. Some data sets focus on particular phenotypes, while others are cohort studies that record everything about a cohort (for example the the Avon Longitudinal Study of Parents and Children, ALSPAC[87], and the UK Biobank[88]). Even in data sets that contain the necessary information, it is often not possible for researchers to access the whole data set, due to concerns about de-anonymisation[89].

Some knowledge about how phenotypes are related to each other (e.g. liver cancer is a type of cancer that is found in the liver) is organised in ontologies, which are described in their own section. These ontologies also form a defined vocabulary for terms, with identifiers, definitions, and links to other information.

This huge variety of human phenotype information (questionnaire data, lab results, ontologies, etc) can then be used to investigate the connection between genotype and phenotype.

3.2.5. Methods for measuring the connection between genotype and phenotype

Having collected data genotype and phenotype data, there are a number of different methods of investigating the connection between genotype and phenotype. Some methods focus solely on the "what", seeking to answer the question "*what phenotypes(s)* does this gene have an effect on?", while some focus also on the "how", i.e. answering "*what is the mechanism* behind this phenotype?". Genome Wide Association Studies (GWAS) are the most popular way of finding potential or actual "what" connections, while gene knockouts and building biological pathways is currently the main way of finding "why" connections. There are also many more specific kinds of experiments which can contribute pieces of the puzzle.

Efforts to uncover the links between genotype and phenotype broadly take one of three approaches:

1. **DNA-centric:** Looking closely at one particular variant (e.g. SNP or gene) at a time, and comparing data to a change in only this variable (knocking out a gene). This is a successful, but expensive methodology, but it isn't hugely successful at finding multi-genic (aka complex) traits. In addition to it's success at uncovering simpler traits, these methods create a wealth of information which are collected in databases.
2. **Phenotype-centric:** Looking closely at one particular phenotype and comparing the genetics of people with the phenotype to those without it. Again this is expensive and it also has the problem of finding that in lists of potentially involved genes, some will be there by random chance.
3. **Cross-cutting:** Computational methods which aim to uncover protein or variant function, and protein interactions across the genome, and across phenotypes. These methods tend to rely on data from (1) and (2). I will discuss some of these in the Section 3.4.

Connections to phenotype can be made with different scales and types of genetic features, from SNPs, genes, transcripts, and proteins to networks thereof and variation within populations. Computational biology links these resources well, so that knowledge at these different scales can be investigated, the Gene Ontology Annotation resource for example, connects information from many of these computational and experimental sources at the level of the gene. Similarly, OMIM.org[90] (Online Mendelian Inheritance Map) and the Human Gene Mutation Database[91] are also resources which seek to catalog links between (human) genetic and phenotype information - both of which contain and share resources with Gene Ontology.

3.2.5.1. Genome Wide Association Studies

Genome Wide Association Studies (GWAS) are large observational studies where the genotypes of a cohort with a specific phenotype (e.g. diabetes) are compared to the genotypes of a cohort lacking in that phenotype (i.e. a control group) in order to find genomic loci that are statistically associated with the phenotype. This has been a popular type of scientific enquiry since the first GWAS study in 2005. GWAS generally results in lists of SNPs, often in the hundreds, ordered by p-value. Disentangling which of these SNPs (if any) *cause* the trait (in addition to correlating with it) is tricky, particularly since GWAS specifically interrogates common variants.

The GWAS catalog database[92,93] was founded in 2008, to provide a consistent and accessible location for published SNP-trait associations, which extracts information about experiments from the literature (currently over 70000 associations from over 3000 publications).

Phenome-Wide Association[94] Studies (PheWAS) are an extension of these where multiple phenotypes are investigated at once in the same cohort of people.

3.2.5.2. Gene Knockouts

Insight into gene function can be gained by "knocking out" a gene, preventing it from being translated into a working protein, for example using CRISPR. Combinations of up to four genes can be knocked out in a single experiment. Knocking out a gene can lead to a difference in phenotype, and differences in gene expression, which can be used to help determine gene regulatory networks. There is a lot of existing data on the phenotypic results of mouse knockouts, since they are often used to create mouse models for diseases. Unfortunately, it is not always well-recorded when knockouts lead to no detectable phenotypic change[95].

3.2.5.3. Biological Pathways

Biological pathways are generally built either through a data-centric method, i.e. beginning with gene expression or mass spectrometry data[96], or through a knowledge-centric method, by beginning with a graph based on knowledge from publications and domain experts[97]. There are a number of popular databases which store pathways, for example Reactome[98] and KEGG[99]. These resources are well-linked to other sources of information, for example gene, rna, protein and chemical databases.

3.3. Ontologies

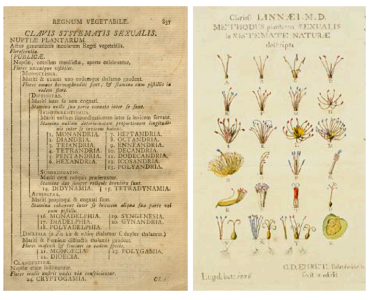


Fig. 3.3 Carl Linneaus developed a system of classifying plants, animals and minerals, including plant classification based on their number of stamens[100]. The left image is a key to this classification system taken from his book, while the right image is a depiction of how the system works, drawn by botanist George Ehret[101].

Cataloguing and classifying has been a successful scientific endeavour in other disciplines (e.g. the periodic table), but it's a cornerstone of biology. Biological classification dates back to the Linnaean taxonomy from the mid 1700s (see Fig.

Linneus and scientific racism

3.3), which described species, their features, and the relationships between them[103]. It also contained some hateful racist ideas. Nonetheless the idea of measuring and categorising the biological world also birthed an enduring tradition of classification in biology, which have led to some of it's most important discoveries.

> Linneus' classifications included a racist hierarchical classification of human beings[102].

Modern biology continues in this tradition of classification, cataloguing biology in ever more (molecular) detail: cells, genes, transcripts, proteins, and pathways. One major way in which this data is synthesised is through the use of ontologies.

3.3.1. What are ontologies?

Ontologies are a way of organising all of the information we have collected in classifying and annotating biological concepts and entities, into a unified framework: one which we can represent, build, and query computationally. Biological ontologies represent knowledge that we have about the relationships between biological entities. Ontologies have classes (called *terms*), which are organised in hierarchies, i.e. such that a term can be a subclass of another. For example, in an ontology of anatomy, we could see that the *left heart ventricle* is an example of a *heart ventricle*, which is part of the *heart*. And more distantly, the *left heart ventricle* is part of an *organ*.

There are ontologies organising all kinds of biological concepts: a number of ontologies that contain anatomical entities (like the heart example) for individual species, ontologies for molecular function, biological processes, diseases, cellular components, etc. What they have in common is that they organise entities through names, descriptions, and IDs, and relate these classifications to one another hierarchically, sometimes with multiple types of relationships (e.g `is_a`, `part_of`). The hierarchy of ontologies can be thought of as having a tree-like structure with one, or just a few root terms which are very general terms that all other terms in the ontology are related to, for example *biological process*, and leaf terms, which are the most specific terms in the ontology (e.g *positive regulation of cardiac muscle tissue regeneration*).

Relations between terms are directional, for example *positive regulation of cardiac muscle tissue regeneration* is a *regulation of cardiac muscle tissue regeneration*, but not vice versa. In such relationships, we say the *parent* term is the more general term closer to the root (e.g. "positive regulation of..") and the *child* term is the more specific term ("regulation of.."). It is not permitted for there to be cycles in ontologies, for example *term A* `is_a` *term B* `is_a` *term A*: ontologies are often DAGs (Directed Acyclic Graphs).

Ontology term identifiers are usually of the form: `XXX:#######`, where `XXX` is an upper-case identifier for the whole ontology, e.g. `GO` for Gene Ontology, `CL` for Cell Ontology, etc. For example, `GO:0008150` is the GO term for *Biological Process*.

Some ontologies also include *annotations*: these relate the terms to other types of information. In the Gene Ontology, there are *annotations* which relate gene functionality to genes, for example. There can also be annotations linking to publications from which the knowledge about the term was obtained.

> **Ontologies summary**
>
> Ontologies:
>
> - Organise information about *terms* into a framework, with relationships between them.
> - Organise terms hierarchically, into Directed Acyclic Graphs, such that there are more specific *child* terms which are subclasses of more general *parent* terms.
> - Have a tree-like structure with the most general terms being the *root* and the most specific being the *leaves*.
> - Allow entities (terms) to be *annotated* with additional information, e.g. annotating gene functions to genes.

3.3.2. How are ontologies created, maintained, and improved?

Biological ontologies are generally created through some combination of manual curation by highly skilled bio-curators and logic-testing (checking for illogical relationships, for example using ROBOT[104]). Creating an ontology is generally a long-term project, with new suggestions and updates to the ontologies being made as new knowledge accumulates, or just as more people have time to add to them. As well as being the work of dedicated curators, contributions to ontologies can usually be crowd-sourced from the scientific community using GitHub issues, mailing list discussions, web forms, and dedicated workshops. In this way, they are similar to other bioinformatics community-driven efforts like structural and sequence databases.

Since they are time-consuming to produce and require such expertise, successful ontologies tend to have (or at least begin with) a quite specific scope, for example the anatomy of a zebrafish. However, there are also cross-ontology mappings and annotations, where terms from one ontology are linked to those in another (e.g. relating gene functions and tissues) or to entities in a database (e.g. gene functions to genes). These also require the work of dedicated curators, who search through literature, assessing various criteria for the inclusion of an annotation (such criteria vary by ontology). Since this is a laborious process, there are also many computational methods to annotate ontology terms automatically.

3.3.3. Examples of ontologies

3.3.3.1. Gene Ontology

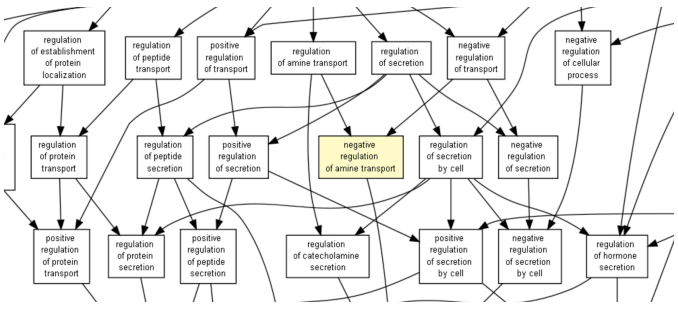


Fig. 3.4 A subsection of the Gene Ontology with arrows showing the existence of relationships (image generated using GOrilla[105])

The Gene Ontology (GO)[106] is one of the first biomedical ontologies, and continues to be one of the most popular. It is a collection of resources for cataloging the functions of gene products and designed for supporting the computational representation of biological systems[107]. It includes:

1. The standard gene ontology, which is a hierarchical set of terms describing functions.
2. The gene ontology annotations (GOA) database[108], which contains manual and computationally derived mappings from gene products to gene ontology terms.
3. Tools for using and updating these resources.

Gene Ontology terms: The Gene Ontology defines the "universe" of possible functions a gene might have (in any species), while the functions of particular genes are captured as annotations in the GOA database[107].

The terms in the GO ontology are subdivided into three types (molecular function, biological process, and cellular component), meaning that GO is actually a collection of three ontologies[106]. Gene products in GO are assumed to carry out molecular-level process or activity (molecular function) in a specific location relative to the cell (cellular component), and this molecular process contributes to a larger biological objective (biological process)[107].

Molecular functions terms describe activities at the molecular level (i.e. that can be undertaken by individual gene product molecules) such as catalysis, transport, and binding. Biological processes terms represent larger scale functions (requiring several molecules), such as regulation, or even behaviour - these stop short of representing biological pathways (GO does not include the types of relationships that would facilitate this).
Cellular component terms describe what part of the cellular anatomy a gene product is part of, e.g. intracellular organelle, ribosome, or cell surface.

The terms in these three sub-ontologies are related to one another by relations, the most common are `is_a` (i.e. is a subtype of); `part_of`; `has_part`; `regulates`, `negatively_regulates` and `positively_regulates`.

Gene Ontology Annotations: Annotations in the GOA database are annotations between GO terms and gene products (proteins, protein complexes or RNA). The annotations include integration to the Uniprot protein function annotations across many species, which have been connected to the controlled vocabulary of GO by skilled biocurators, as well as electronically generated annotations. Evidence codes are provided for annotations which label whether annotations were verified by experts, as well as what type of experimental or computational evidence there is for an annotation. GOA also link to the supporting publications for the experimental annotations.

3.3.3.2. Uberon Ontology

Uberon is a cross-species anatomy ontology[109], whose terms represent body parts, organs, and tissues in a variety of animal species (mouse, xenopus, fly, zebrafish) and specific structures (Neuroscience Information Framework (NIF) Gross Anatomy, Edinburgh Human Developmental Anatomy). It is particularly strong in it's integration to other ontologies, including anatomy ontologies for individual species, the Gene Ontology, Cell Ontology, phenotype ontologies (e.g. mammalian phenotype, human phenotype), the Experimental Factor Ontology (EFO), etc.

3.3.3.3. Other Ontologies

There are many other ontologies which aim to catalogue other aspects of biological experiments and knowledge. Other ontologies which are used in this thesis include:

- The Cell Ontology[110] (`CL`) describes cross-species cell types (from prokaryotes to mammals, but excluding plants). Example relationship: Osteocyte `is_a` Bone Cell `is_a` Animal Cell.
- The Disease Ontology[111] (`DO`) describes human disease. Example relationship: Blastoma `is_a` Cell-type Cancer `is_a` Cancer.
- Human Phenotype Ontology[112] (`HP`) describes "human phenotypic abnormalities encountered in human disease". Example relationship: Motor Seizure `is_a` Seizure `is_a` Abnormal Motor System Physiology.
- The Experimental Factor Ontology[113] (`EFO`) describes experimental setups common to the EBI databases. It is well-integrated with CL, Uberon, and ChEBI (chemical compound ontology). Example relationship: RNA Extraction Protocol `is_a` Nucleic Acid Extraction Protocol `is_a` Extraction protocol.

3.3.4. Why are ontologies useful?

Ontologies can be used by researchers to investigate specific genes, tissues, functions of interest, or more generally to get a big-picture viewpoint on large groups of such entities. With logical reasoning, we can generate inferred relationships between distantly related terms in ontologies, for example $is\_a \cdot part\_of \implies part\_of$. This allows us to find and check relationships that are not in the ontology automatically.

Ontologies and particularly their annotations are varying degrees of incomplete, and this will have an impact on the results of any downstream use of them.

3.3.4.1. Term enrichment

Ontologies are often used to try to make sense of a list of genes that are found to be differentially expressed across different experimental conditions, or that are outputs from GWAS. In the context of GO, a term enrichment analysis can be carried out to see which GO terms are overrepresented (a.k.a. enriched) for a given group of genes, thus saying something about the function of the list of genes.

3.3.5. File formats

There are two major file formats in which ontologies are currently stored. The OBO format is a human-readable format, while the OWL format is more complex, but has more functionality, and for example can be queried using SPARQL (an SQL-like querying language).

3.4. Predictive computational methods

The low cost of sequencing means that databases of sequences have been expanding very rapidly in comparison to other information, which is much harder to determine. Computational predictive methods aim to predict structure or function from sequence in order to bridge this gap. Here I describe some of the challenges and methods in this space, many of which leverage the ontologies and databases described in the previous sections.

3.4.1. Prediction tasks: Protein classification prediction

As previously mentioned, proteins are often classified by structural similarities. This information is often used because researchers identify a gene of interest, but information about it's function or structure (in PDB) has not yet been captured and stored (i.e. the protein is "uncharacterised"). In such cases, it's often necessary to make inferences about protein structure or function based on their similarity to known proteins. This is sometimes done using sequence similarity (e.g. BLAST), but sequence similarity can vary considerably between proteins with the same underlying structure. This is why structural similarity searches based on protein classification are preferred.

3.4.1.1. SCOP

The Structural Classification of Proteins (SCOP) database[38] classifies all proteins with known structure based on their structural similarities, based on the consideration of the protein's constituent domains. The classification is mostly done at the level of families, superfamilies, and folds arranged in a tree structure. Families represent the most similar proteins, which share a "clear evolutionary relationship", while superfamilies represent less close evolutionary relationships, and folds represent the same secondary structure. This protein classification task, while aided by automation, was carried out largely by manual visual inspection.

> **BLAST**
>
> The Basic Local Alignment Search Tool[114], is an extremely popular tool that is used to perform a basic search of nucleotide or amino acid sequences to known sequences, based on statistically significant similarities between parts of the sequence.

SCOP was updated until 2009, but has been succeeded by SCOP2[115]. However, SCOP2 has a different underlying classification system, based on a complex graph, rather than a hierarchy. The CATH (Class, Architecture, Topology, Homologous superfamily)[116] database provides another classification system, which operates hierarchically, but is created mostly via automation, which leads to major differences

between the classifications[117].

3.4.1.2. SUPERFAMILY

SUPERFAMILY[118] is structural annotation procedure and associated web resource, which uses Hidden Markov Models (HMMs) to assign sequences to SCOP domains, primarily at the superfamily level. This is a process analagous to gene annotation.

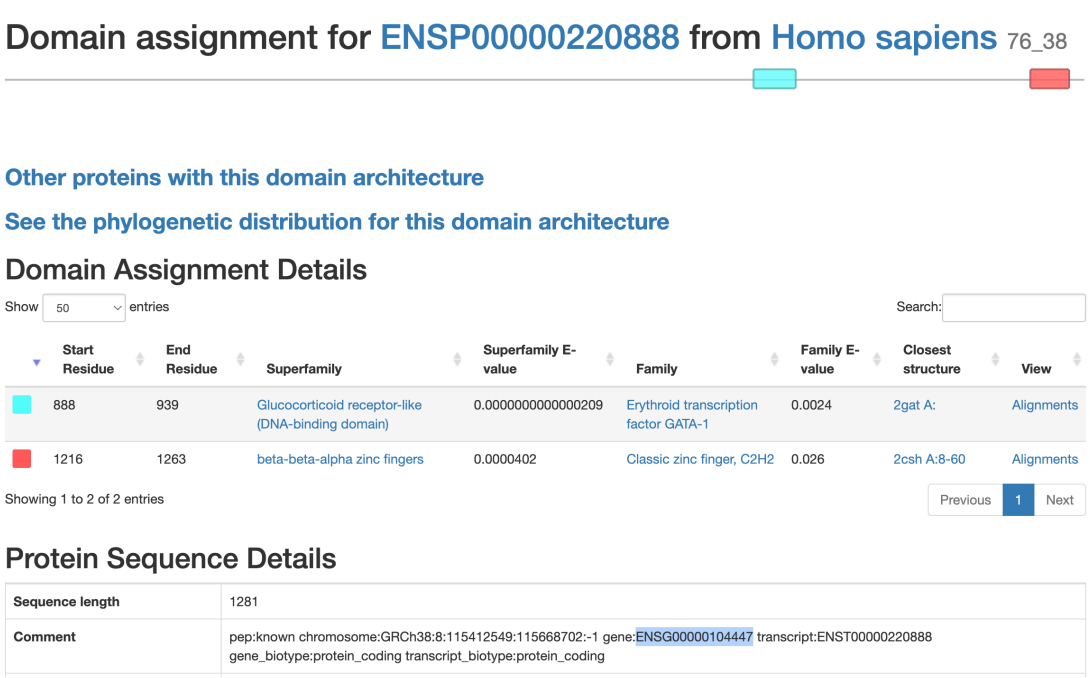


Fig. 3.5 Domain assignments for **ENSP00000220888.5** - a human protein. The image shows the protein is mostly not made up of protein domains, but has two domains assigned, highlighted in blue (*Glucocorticoid receptor-like (DNA-binding domain)*) and red (*beta-beta-alpha* zinc fingers).

This domain assignment is also used as a form of functional annotation and allows the functions of poorly understood proteins to be inferred based on how closely they match known superfamilies. This is how the website is frequently used - as it allows users to search a new sequence and see which domains are predicted to fall in the sequence and where. It is also possible to see domain assignments of known proteins, e.g. by ENSP (Ensembl Protein ID), see Fig. 3.5.

HMMs are very successful at such assignments since pairwise correlations between proteins (or their domains) and other proteins in the family may be weak, but consistently for many proteins; this can be picked up by an HMM. The superfamily level is chosen since it is the broadest level which suggests evolutionary relationships, but SUPERFAMILY also generates assignments at the (stricter) family level.

HMMs are created by first finding closely relating protein homologs for a given protein superfamily using BLAST, and then extending it by comparing the HMM to more distantly related homologs. The resulting HMM library is fine-tuned by some manual curation.

Although SUPERFAMILY's primary resource is it's HMM library, it also integrates a range of other tools for sequence analysis, for example protein disorder prediction (D2P2) and GO annotation (dcGO), as well as a domain-based phylogenetic tree (sTol). In addition, SUPERFAMILY makes available all sequences that it uses to build HMM models, some of which cannot be found elsewhere.

SUPERFAMILY update

> **Contributions in this section**
>
> The SUPERFAMILY website and resources were jointly maintained between members of Computational Biology group (then) at Bristol, which involved replying to user emails, and updates to the website.
>
> In 2014, a SUPERFAMLIY update paper was published. I contributed by editing the paper and adding a small number of proteome sequences in the class Cyanophyceae - 5 of the proteomes available on SUPERFAMILY.

I contributed to SUPERFAMILY's 2014 update[3]. The SUPERFAMILY database of proteomes doubled from 1400 to over 3200 from 2010 to 2014, containing sequences from across the tree of life, including 1714 species and 1544 strains. The update paper described this development, as well as highlighting SUPERFAMILY as a resource for unique proteomes that are not found elsewhere (e.g. Uniprot), and describing the update to the (at the time) most recent human reference genome.

3.4.2. Prediction tasks: Protein function prediction

Human genes can have multiple functions, but currently, we don't even know one function for all of them. Although we have their sequence, some genes are completely functionally unknown to us. Protein function prediction is the task of predicting protein function (usually in terms of ontology terms) from protein sequence.

3.4.2.1. DcGO

The aim of the domain-centric Gene Ontology (dcGO)[119,120] tool is to give insight into uncharacterised or poorly characterised proteins by leveraging information about the content of their constituent protein domains. It annotates domains and combinations of domains (a.k.a. supradomains) to phenotype terms from a variety of ontologies, including the Gene Ontology (GO), Mammalian Phenotype ontology (MP), Disease Ontology (DOID), Zebrafish ontology (ZFA). Domain information comes from SUPERFAMILY, and annotations between (supra)domains and phenotype terms are made below a cut-off of FDR-adjusted statistical associations between the entities. Using phenotypes from a range of species serves to make use of greater numbers of experiments, and therefore increases the number of little-known proteins across species that DcGO can make predictions about.

> **Error types and False Discovery Rate (FDR)**
>
> In making predictions, there are two types of errors that we can make, false positives F_p (Type I errors) in which we wrongly think something is true, and false negatives F_n (Type II errors) in which we wrongly thing something is false. Similarly we can get predictions right in two ways, correctly thinking something is the case (true positives T_p) or isn't (true negatives T_n).
>
> The FDR is a measure of the false positives, which takes into account the number of predictions: $FDR = \frac{F_p}{F_p + T_p}$

3.4.3. CAFA

Critical Assessment of Functional Annotation[5,121,122] (CAFA) is an international community-wide competition for the prediction of protein function, which aims both to stimulate research in the field of protein function prediction, and to measure progress in the field. It has been running approximately every 2-3 years since 2013.

Each CAFA challenge begins by the organisers releasing a large number of target sequences (over one hundred thousand) across multiple species, about which participant teams must make predictions. After the competition closes, the organisers wait 3 months, by which time, new experimentally verified protein functions will be found (representing ~3% of sequences in past competitions) and these are the data set against which the predictors are measured.

Participants can use any additional data they see fit to make predictions, which must be triples containing a sequence ID, ontology term ID (e.g. a GO/HP identifier), and a confidence score between 0 and 1. A score of 1 indicates a completely confident prediction, while a score of 0 is equivalent to not returning the prediction. Each team may submit up to three models, the best of which is ranked.

The target sequences consist of a mixture of “no-knowledge” and “limited-knowledge” sequences. No-knowledge sequences are sequences which upon release have zero experimentally-validated GO annotations to any of GO’s three constituent ontologies (biological process, cellular component, and molecular function). Limited-knowledge sequences are sequences with one or more annotations in one or two GO ontologies, but not all three.

3.4.4. Prediction tasks: Variant prioritisation

Variant prioritisation is a version of protein function prediction in which long list of genes or variants (obtained for example through a GWAS experiment) are narrowed down to a shorter list of variants or genes which are more likely to be causal.

3.4.4.1. FATHMM

Functional Analysis through Hidden Markov Models (FATHMM)[123] is a tool for predicting the functional effects of protein missense mutations using sequence conservation information (via HMMs), which can be (optionally) weighted by how likely a mutation in a protein/domain would be to lead to disease. FATHMM can only score missense mutations because it scores SNPs based on the probability of specific amino acids existing in proteins. Weightings are calculated from the frequency of disease-associated and functionally neutral amino acid substitutions in protein domains from human variation databases (the Human Gene Mutation Database[124] and Uniprot-KB/Swiss-prot[78]).

Consequence files describing whether an amino acid results in a missense, nonsense or synonymous SNP must first be obtained by using Ensembl’s Variant Effect Predictor[125] in order to create input to FATHMM. FATHMM then calculates conservation scores which are a measure of the difference in amino acid probabilities for a SNP according to the HMM, i.e. between a wild type amino acid and it’s substitution. A reduction in amino acid probabilities is interpreted as a prediction of deleteriousness (likelihood to cause harm), and the larger the reduction the greater the predicted harm.

3.4.5. Prediction tasks: Phenotype prediction

Phenotype prediction is the task of predicting phenotypes from genotypes. Specific tests of phenotype prediction might look like matching genotypes to profiles of traits, or predicting specific phenotypes from genotypes.

Although it’s often presented as a separate task, phenotype prediction is closely linked to protein function prediction. When variants on the protein-coding genome are known to be responsible for phenotypes, the assumption is that variant impacts the protein and the protein has a function that causes the phenotype when it behaves differently than usual. This is the assumption that underlies annotations between genes or proteins and phenotype terms, and it’s also the assumption that underlies phenotype “prediction” algorithms like 23andMe or Promethease’s health reports, which count the presence or absence of individual variants thought to be associated with disease, in order to inform potential phenotypes e.g. “You have 2 alleles associated with causing Breast Cancer”.

3.4.5.1. CAGI

Critical Assessment of Genome Intepretation[126] (CAGI) is a prediction competition open to the research community, in the same tradition as CAFA, this time aiming to objectively assess predictive methods for determining the phenotypic impacts of genomic variation across a number of different challenges.

The precision of the best methods in phenotype prediction of rare illnesses is still below 50%[127].

3.5. Sources of bias in computational biology

The wealth of Open resources in computational biology, from databases to predictive methods to ontologies, hold exciting possibilities and are a credit to the collaborative spirit of the field. It’s still important, however, to look at them with a critical eye, in order to be aware of their limits.

3.5.1. Trusting the results of research

> The imposing edifice of science provides a challenging view of what can be achieved by the accumulation of many small efforts in a steady objective and dedicated search for truth.
>
> —Charles H. Townes

We all want to be able to trust the results of scientific research. Not only when it’s our own, but because science builds on itself and building on shaky ground wastes time and money. Moreover, scientific research is generally paid for by tax, and the results that are generated by it drive policy, drug treatments, and innovations. Everyone has a vested interest.

In all fields, science is a search for knowledge. And in all fields, there are concerns about what makes bad, unreliable, un-useful, or biased research; what must be done or not done to uphold science’s claim to truth, or at least reliability.

In contrast to other fields, many bioinformatics datasets have been freely available and accessible on the internet since their inception; in this sense the field is far ahead of others. The issues which affect the reliability of science in general, however, are likely to be present in computational biology, too. This could have strong effects on the research that is reliant on these large ontologies and databases.

3.5.1.1. Science’s self correcting mechanism

Scientific results are often based on statistics, so it’s inevitable that some proportion of published scientific results will not be true simply due to the sample on which the hypothesis was tested. The common wisdom is that this isn’t a problem, as over time, researchers can double-check interesting scientific results, and the literature can be updated to reflect that. This is known as sciences *self-correcting mechanism*. If a result can be replicated in a different circumstance by a different person, it reinforces the likelihood that the result is true. A replication doesn’t have to reveal the exact same level of statistical significance or effect size to be successful, but (usually, depending on definitions) just a similar result.

3.5.1.2. What makes research trustworthy?

	Data: Same	Data: Different
Analysis: same	*Reproducible*	*Replicable*
Analysis: different	*Robust*	*Generalisable*

Table 3.2 Definitions relating to reproducibility, adapted from The Turing Way.

There are different levels of trust that we might have in the results of research. This ranges from a basic trust that the researchers didn't make any mistakes in their implementation (their code is doing what they thought it was) to a trust that the result is trustworthy even in new contexts. Much of the academic discussion surrounding this hinges on the concept of *reproducibility*, for which there are many contrasting definitions. I like the definitions from The Turing Way[128], shown in Table 3.2. This table says for example if you get the same result with the same data and same analysis as the original research, then the result is *reproducible*. And if you get the same result when the data is the same, but the analysis is different (e.g. a different implementation of the code, or a different specific analysis meant to measure the same thing), then the result is *robust*.

Although a generalisable result is the most desirable and interesting, as long as the research is *reproducible*, it can still positively contribute towards our joint scientific knowledge. This definition of reproducibility also requires that everything needed to run the experiment again is provided, including fine details of methods (in computational biology, this is often equivalent to code) and data. In the absence of this, science's self-correcting mechanism is short-circuited.

3.5.2. The reproducibility crisis

> In science consensus is irrelevant. What is relevant is reproducible results.
>
> —Michael Crichton

The reproducibility crisis is the realisation that worryingly large proportions of research results do not replicate. Replication studies have found that only 11% of cancer research findings[129], 20-25% of drug-target findings[129,130], and 39% of psychology findings[131] could be reproduced. Surveys of researchers across disciplines reveal that more than 70% of scientists say they have failed to reproduce another paper's result, and over 50% say they have failed to reproduce their own results[132]. It seems that science's self-correcting mechanism is not working as intended.

3.5.3. Sources of irreproduciblity and how to combat them

The surprising irreproducibility seen is thought to be explained by a range of factors[133] including poor data management, lack of available materials/details of experiments, publication bias, poor statistical knowledge, and questionable research practices such as HARKing and p-hacking. Although it is difficult to estimate, a very small proportion of irreproducible research is thought to be due to fraudulent practices[134] (although these do still happen), and arguably it's explainable simply from our reliance on Null Hypothesis Significance Testing[135].

3.5.3.1. Null Hypothesis Significance Testing

To discuss some of these issues, we first have to understand how scientific hypotheses are usually tested and reported: Null Hypothesis Significance Testing (NHST). This reporting usually consists mostly of a p-value as a measure of statistical significance: the likelihood that a false positive at least this extreme could be obtained just by chance. The threshold for this, usually denoted by α is most often set to 0.05, as recommended by Fischer, however this is not necessarily the most sensible cut-off for science today[136,137], and different fields have differing cut-offs.

Fischer racism

Fischer campaigned for the legalisation of eugenic sterilisation.

> Statistical significance is the least interesting thing about the results. You should describe the results in terms of measures of magnitude –not just, does a treatment affect people, but how much does it affect them.
>
> —Gene V. Glass

Despite the dominance of p-values as main or only reported statistic across scientific fields, they do not imply that a result is interesting (the effect might be small or the hypothesis uninteresting), or even that it's likely to be true. Sometimes the p-value is not even reported, but only whether or not it crossed the p<0.05 threshold.

False positives in NHST

In the context of NHST in research, a false positive would be finding a p-value below our threshold (0.05) for our sample of measurements when there is not an effect in the underlying population from which we are drawing, while a false negative would be finding a p-value above our threshold for our sample when there is an effect present in the underlying population.

3.5.3.2. P-hacking and HARKing

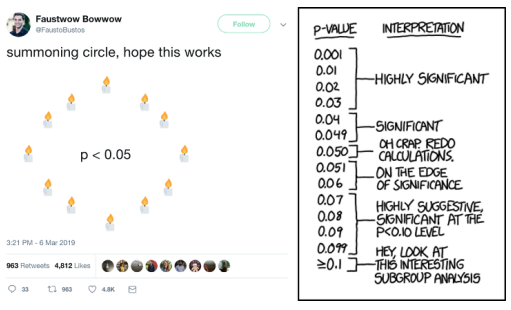


Fig. 3.6 Images that are illustrative of researchers approaches to p-values and p-hacking. The left image is a popular tweet, while the right image is an xkcd comic).

The pressure on scientists to publish means that researchers may be tempted to (or may accidentally, due to statistical ignorance) employ data-mining tactics in order to harvest significant p-values. This practice is known as "p-hacking", and evidence for its existence can be found in distributions of p-values in scientific literature[139], as well as popular culture (Fig. 3.6). This can include rerunning analysis with different models/covariates, collecting data until a significant p-value is reached, or performing 20 experiments and only publishing the results of one as in HARKing.

HARKing

HARKing (Hypothesising After Results are Known) is a variant of p-hacking in which researchers look at their data and/or run significance tests and then add or remove hypotheses based on the results, but then present "significant" results as *a priori* (previously held) hypotheses[138]. Xkcd has a comic for this, too.

> The first principle is that you must not fool yourself – and you are the easiest person to fool. – Richard Feynman

There are several suggested tonics to the problem of uninformative and ubiquitous p-values. Reporting p-values as numbers (e.g. "p=0.0012") rather than in relation to a threshold (e.g. "p<0.05" or "the hypothesis was found to be highly significant") is a starting point. Information about statistical power and effect size should also be provided. In addition to giving researchers reading a paper a better idea of the quality of it, this also allows science to self-correct a little easier, since individual p-values can then be combined into more reliable p-values, using for example Fischer's method[140].

For cases where many hypotheses are being generated at once (for example in GWAS), multiple hypothesis corrections (e.g. the Bonferroni correction[141] or the False Discovery Rate[142]) can be employed to adjust the p-value to account for this.

3.5.3.3. Publication bias

Although with standard p-value and statistical power cut-offs, negative results are more likely to be true than positive ones[135], negative results are much harder to publish. This bias is likely to be responsible for the draw of questionable research practices like p-hacking. It also means that there is a lot of unpublished, negative results which are likely to be repeated, since there is no way that someone could know it has already been done. A highly powered negative result could be very interesting, for example, we know hardly anything about which genes do not appear to affect phenotypes, since these results are not published[95], but they would help enormously with the challenge of creating a gold standard data set for gene function prediction.

Publication bias is usually used to descibe the bias against negative results but there are other forms of publication bias which affect computational biology, for example the discrepency in which genes are studied. Some genes are very famous, racking up thousands of publications, while others are entirely unstudied. Even looking only at human genes, there is a huge divide between the most and least studied genes. This means that there are many functions of genes which will be missing from the gene ontology annotations (for example) for less well-studied genes.

Another example is the bias against replications. Repeats of studies are not commonly published (as they are not novel). Naturally, this discourages people from doing them, or at least from writing them up. This is true for both computational methodologies (where often the code and computational environment needed to replicate the research are not provided), and for experimental work. This means that it is difficult for science to self-correct this work.

3.5.3.4. Code and data availability

A lack of code and data availability, while not necessarily leading to wrong or untrustworthy results, also has a place in making research irreproducible since:

1. Research cannot employ it's self-correcting mechanism unless the experiment can be repeated.
2. If the code or data is obscured, then many of the decisions may also be obscured. Decisions we make in analysis[143,144] and even small details of implementation can effect the results of research, and whether we expect it to generalise to another similar context.

Versions

Different versions of the same package, or even the same well-maintained base software (e.g. in R or Python) can have very different behaviours. For example in `R4.0.0`, `as.numeric(data.frame("10","50","20"))` would return a list `10 50 20`, while earlier versions process the same code assuming these characters to be factors and would return instead `1 3 2`.

This means that it is important for packages to understand how they interact with versions of each package, and also shows how small details in implementation decision-making could effect results.

This is a problem even in computational fields: in computational biology, roughly a third of papers that use code still do not provide it at all[145]. Even then, providing the data or code somewhere is not in itself enough to overcome the problem of irreproducible methods: it must also be usable. The FAIR principles[146,147] provide a framework for ensuring this. In the context of software, this includes sharing your reproducible computational environment (for example, exact versions of the packages that you used).

FAIR data and code

Data and code are *FAIR* if they are *F*indable, *A*ccesible, *I*nteroperable, and *R*eusable for both humans and machines. This includes working with existing standards and softwares, being well-documented, etc.

While there is a growing movement towards transparent and available materials, and taking time to create these (e.g. the slow science manifesto[148]), there is also friction against it due to research's deeply ingrained culture of "publish or perish". This incentivises doing the minimum possible to publish, and disincentivises spending time on quality control or providing useful metadata wherever you can publish without it.

3.6. Proteome Quality Index

Reducing bias and improving data and research quality in computational biology is a broad and active area of research. This section explains one such effort which I contributed to: the Proteome Quality Index (PQI)[2], which provides a variety of individual quality metrics as well as a "star rating" for complete genome sequences.

> **Contributions in this section**
>
> The Proteome Quality Index paper was created as a joint project between the Computational Biology group (then) at Bristol. I contributed to ideas for metrics, code to calculate some of these metrics, and paper editing.

3.6.1. Introduction

Although sequencing, assembly, and protein annotation of new genomes is a challenging and complex task, there are now thousands of organisms with sequenced genomes and identified protein sequences (proteomes). These proteomes can vary hugely in quality; in creating a daily-updated phylogenetic tree, sTOL (sequenced Tree Of Life), it was found that many sequenced genomes were missing vital proteins due to poor sequencing[149]. This could have a far-reaching impact on research results since such genomes are reused by many researchers, particularly in comparative genomics analyses, where omissions of whole proteins and poor accuracy of others are likely to affect results. While quality control and data submission guidelines were more developed in other areas of computational biology, similar guidelines for genome quality were lacking.

In response, eleven different proteome quality metrics were developed, applied to 3,213 proteomes at the time of writing, and a website built to display the results, as well as being presented in a paper[2]. This website was intended to be both a way for users of genome data to look up the quality of a genome in advance of some research and, more importantly as a talking point for quality guidelines for genomes with known proteins. Since the development of PQI, a twelfth metric, DOGMA[150], was added to the website, donated by the Bornberglab.

3.6.2. PQI metrics

PQI's original eleven metrics were either local i.e. "clade-based" (in which proteomes are compared to similar organisms) or global (in which case it is compared to all other proteomes). A clade is a group of organisms that consists of a common ancestor and all its descendants, i.e. is a branch on the tree of life[151], so an appropriate ancestor must be chosen to define the clade. For PQI, since the purpose of these clades was to compare its' constituent proteomes, we wanted clades that had similar variability. This was achieved by choosing parent nodes that are at least 0.01 in branch length away from the proteome (leaf node), and such that the clade contains at least 10 species. Trees and branch lengths to carry out these calculations were taken from sToL[149]. For clade-based metrics, proteomes score well if they have similar scores to the rest of the clade. Descriptions of the 12 different metrics currently in PQI can be seen in Table 3.3.

Metric name	Type	Description	Notes
1. Percentage X-content	Global	Percentage of proteome with amino acids denoted by 'X', excluding the first residue of each protein.	Amino acids that cannot be identified, or can have more than one value are represented by an 'X' in the amino acid sequence[152]. This occurs when coverage of the sequencing is low. Uncertainty in translation start sites mean the first residue of a protein is often uncertain ('X') even in the highest quality proteomes, so these are excluded from this measurement.
2. PubMed Publication Count	Global	The total number of publications related to the genome as listed for that entry in the PubMed database[153].	This is a measure of how well-studied a proteome is, assuming that proteomes that have been studied more will be of higher quality.
3. CEG domain architecture inclusion	Global	Proportion of CEG set which contains homologous domains in the proteome, according to SUPERFAMILY	This method assumes that all eukaryotic genomes should contain a core set of well-conserved eukaryotic genes. This score is not calculated for non-eukaryotes. This was done using the Core Eukaryotic Gene (CEG) library used by the now defunct CEGMA tool[154], which comes from the Eukaryotic Orthologous Group (KOG) sequence orthology database[155]. Domain-architecture similarity is calculated using the SUPERFAMILY HMM library.
4. Percentage of sequences in Uniprot	Global	Percentage of proteome sequences that appear in Uniprot with 100% sequence identity	This metric assumes that the majority of discrepancies between Uniprot protein sequences and the proteome protein sequences are due to proteome inaccuracies.
5. Percentage of sequence covered	Clade-based	Percentage of amino acid residues in proteome sequence that are covered by SCOP domain superfamily assignments, compared to the average for the clade.	This metric measures the portion of structured protein sequences found in the proteome as opposed to disordered regions and gaps. This measure assumes related species have a similar breakdown of these types of proteins. A mismatch could indicate that the parts of the genome that are supposed to be protein-coding are an incorrect length, that it is missing proteins, or contains proteins that it shouldn't.
6. Percentage of sequences with assignment	Clade-based	Percentage of amino acid residues in proteome that have SCOP superfamily assignment according to SUPERFAMILY, compared to the average for the clade.	Related species are assumed to have a similar percentage of domains with SUPERFAMILY assignments to SCOP superfamilies
7. Number of domain superfamilies	Clade-based	Number of proteins assigned to domain superfamilies by SUPERFAMILY compared to average for clade.	Assignment to domain superfamilies was obtained using the SUPERFAMILY HMM Library. The number of superfamilies gives an indication of the diversity of the proteome, so a low number compared to the clade may indicate an incomplete proteome, while a high number could indicate that the proteome contains domain superfamiles that it shouldn't.

Metric name	Type	Description	Notes
8. Number of domain families	Clade-based	Number of distinct SCOP protein domain families that are annotated to the proteome, compared to the average for the clade.	The SCOP protein domain families are annotated to the proteome using a hybrid HMM/pairwise similarity method from the SUPERFAMILY resource. Similarly to the number of domain superfamilies, the number of families gives an indication of the diversity of the proteome at the SCOP family level. Domain families were included in addition to domain superfamilies, since they are more specific and may reveal differences that are not apparent at the superfamily level.
9. Mean sequence length	Clade-based	The average length of proteins in the proteome (in amino acids), compared to the average for the clade.	This measure assumes that mean sequence length of proteins should be comparable with those of related species.
10. Mean hit length	Clade-based	Average number of amino acids in superfamily assignments, compared to the average for the clade.	Longer hits represent better matches to SCOP domains. These are assumed to be similar for similar species.
11. Number of domain architectures	Clade-based	Number of unique domain architectures (combinations of SCOP domain superfamilies and gaps) in the proteome, according to SUPERFAMILY, compared to the average for the clade.	Similarly to the number of domain families superfamilies, the number of domain architectures gives an indication of the diversity of the proteome at the SCOP family level.
12. DOGMA[150]	Clade-based*	Percentage of conserved domain arrangments found	DOGMA compares sequences to conserved domains arrangements across six eukaryotic model species (Arabidopsis thaliana, Caenorhabditis elegans, Drosophila melanogaster, Homo sapiens, Mus musculus and Saccharomyces cerevisiae), or alternative sets for mammals, insects, bacteria and archaea. It is therefore not “clade-based” in the same way as other PQI metrics.

Table 3.3 Description of the quality metrics currently in PQI, 1-11 are original metrics, while DOGMA, the 12th, was added later and donated by the Bornberglab.

3.6.2.1. DOGMA

Since the publication of the PQI paper, the DOGMA metric[150], which scores proteomes based on conserved arrangements of protein domains, has been added to the PQI website. The DOGMA score is similar to the PQI “Number of Domain Architectures” score, but since it is based on using model organisms as a ground truth, it scores these genomes highly, whereas this tends not to be the case for PQI’s clade-based metrics.

3.6.3. PQI features

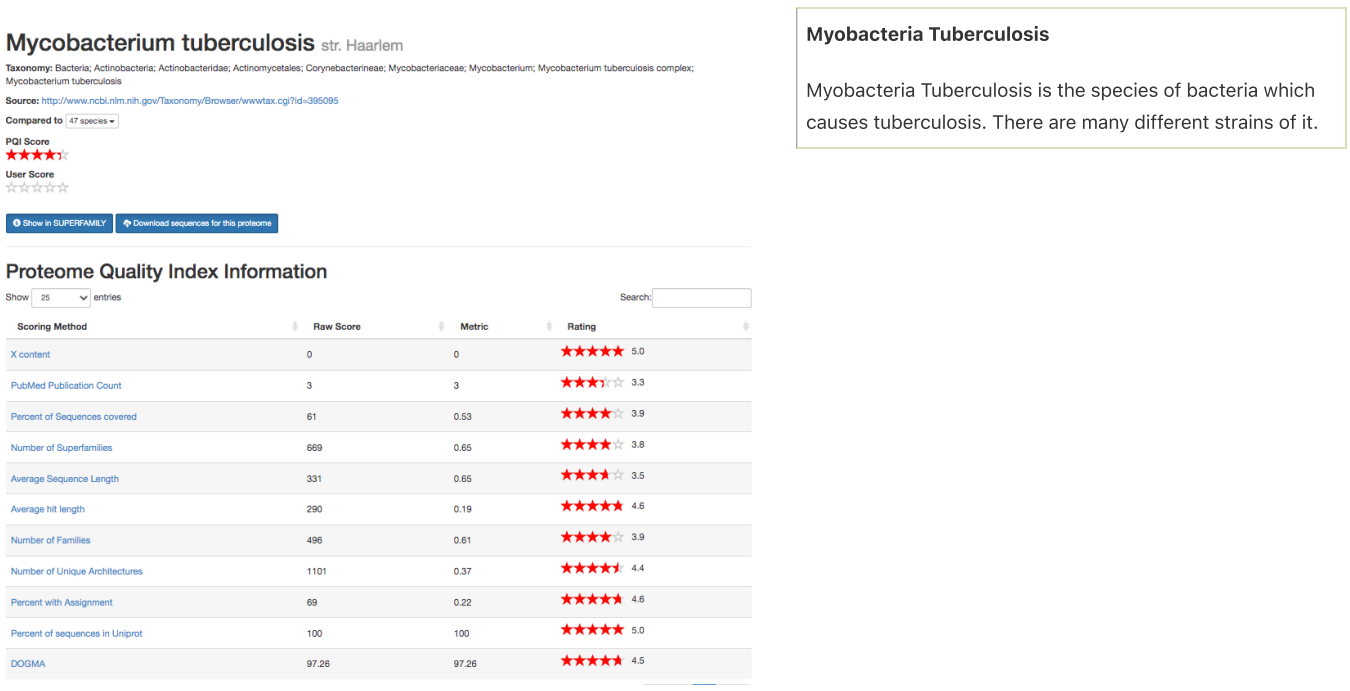


Fig. 3.7 A view of the PQI website for the Haarlem strain of Myobacterium Tuberculosis. For this genome, we see the 12 different metrics with star ratings with a good overall score (4.2), compared to a clade of 47 species. It’s high overall scores in clade-based metrics show that the genome is similar to most other species in the clade.

In addition to the 11 provided metrics (each of which have individual star-ratings), the PQI website provides an overall star-rating scoring system for proteomes, bringing together numerous different metrics which are normalised before being averaged into an intuitive star-rating (1-5 stars) with equal weight given to each metric. Proteomes for particular species can be searched for, filtered by the various ratings,

downloaded, user-rated and commented on. Additional proteomes and metrics can be added/suggested by users via the website and documentation describing this is provided. Fig. 3.7 shows the PQI website for a Myobacteria Tuberculosis proteome with a good overall score.

3.6.3.1. Example usage

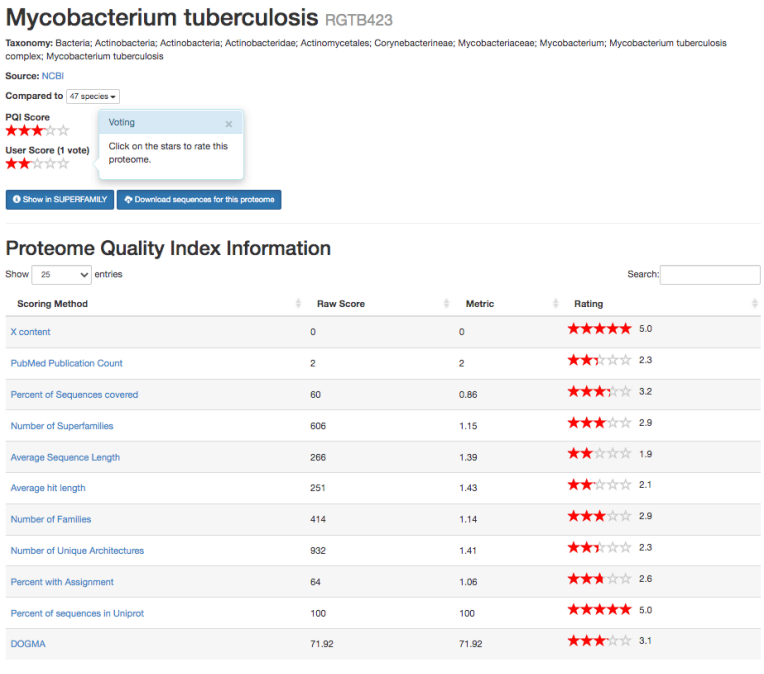


Fig. 3.8 A view of the PQI website for the RGTB423 strain of Myobacterium Tuberculosis. This time the genome has a poor overall score (3.0) according to PQI.

A typical use case for PQI would be to check the metrics of a genome before including it in analysis. Comparing Fig. 3.8 (showing the PQI website for the lower-scoring proteome) to the earlier example in Fig. 3.7, we can see that the RGTB423 strain scores considerably worse in all of the metrics associated with proteins or protein domains. For example, the average sequence length is considerably shorter in RGTB423 than in Haarlem, which makes it very unusual for the clade, meaning that the genome gets only 1.9 stars for this metric. This is indicative of the fact that the proteins in the sequenced RGTB423 strain are incomplete.

3.6.4. Related work

Since the development of PQI there have been further developments relevant to proteome quality.

3.6.4.1. BUSCO

The original BUSCO paper was published slightly before PQI in the same month, and it performs a similar function: assessing the quality of genomes.

> **CEGMA is no longer supported**
>
> The CEGMA tool[154] used as the basis for metric (3) is no longer supported. BUSCO[156] is considered the tool's successor.

BUSCO has a slightly different use-case than PQI in that it is a Python package designed to let people who are assembling and annotating new genomes check the quality of their work. BUSCO is in some ways the "new CEGMA", since CEGMA is now defunct. The difference is mostly that where CEGMA relied on the old KOGs database[157] to define conserved genes (only in eukaryotes), while BUSCO uses the newer OrthoDB database[158] of conserved genes across many organisms.

3.6.4.2. NCBI Assembly database

NCBI's Assembly database[159] has been released since PQI was published. The database tracks how many assemblies there are for each species as well as how many versions of each assembly there has been. This information could be used to weight the importance of proteomes in clade-based metrics. The number of assemblies and versions could also form a separate score.

Sequencing depth (coverage) and read length are also known markers of genome quality[160]. Average coverage for an assembly (except reference genomes) is also available in the NCBI Assembly database[159].

3.6.5. Limitations

There are some limitations of PQI and its metrics:

1. The clade-based metrics downscore unusual genomes. This backfires for unusually good genomes. Model organisms such as Homo Sapiens can get a low score in clade-based metrics for the wrong reasons; they are of unusually high quality compared to those in it's clade. Clade-based scoring also creates irregularities in scoring for clades containing outlying species: the more diverse a clade, the worse every genome's score within it. PQI website's comment and user-rating features can be used to alert its users to these cases, but this remains a major limitation.
2. PQI's intended audience were users of protemes/genomes, rather than creators thereof. It does therefore not release the code for others to calculate these metrics on their work-in-progress genomes/proteomes, unlike BUSCO. This may have limited PQI's ability to positively influence proteome quality.

3.6.6. Potential improvements

3.6.6.1. Change weighting of star-rating

In order to address limitation (1), one option would be to downgrade the importance of clade-based metrics in star ratings. This could be done for all genomes, for model organisms specifically, or preferentially the more they have been sequenced and assembled. In the latter case, the Assembly database[159] could be used as a source of this information.

3.6.6.2. Changes to metrics

PQI was created with the potential to add further quality metrics by other researchers, and remains open to adding these. Here, I give three examples of metrics which would improve PQI. This list is by no means exhaustive. There could be far more additional measures to audit different types of problems in genome sequencing, for example GC content, amino acid bias, or contamination from other genomes.

Inclusion of BUSCO Although the DOGMA metric was added and it is based on BUSCO[156], it might be sensible to add BUSCO, since it has become very popular (thousands of citations), and any proteome/genome quality index would arguably be seen as incomplete without it. The continued inclusion of the CEG Domain Combination Homology metric may also be questioned since CEGMA is no longer being updated (nor is the KEG database upon which it is based). However, BUSCO and CEGMA may be complementary since BUSCO has a weaker requirement for inclusion in the set of proteins, which means that it has more proteins.

Number of assemblies In addition to using the assembly information to choose the importance of clade-based metrics, the number of assemblies for a species and number of versions of an assembly might form a metric of how well-studied an organism is, which may be useful to users.

Technology-based Some "third-generation" sequencing technologies can create much longer read lengths, but potentially lower accuracies. A third category of metrics "technology-based" metrics could exist for proteomes where the metric is only really comparable within similar types of technologies. If technology-based metrics were implemented, it may also be sensible to have some metrics which only exist for specific technologies. For example, for nanopore sequencing, we could implement an indel-based quality metric[161]. Including sequencing-technology-specific metrics may encourage contributions from other researchers who specialise in particular technologies. Sequencing technology is also available in the NCBI Assembly database[159].

3.7. Summary

This chapter described the data pipelines undergone by genotype and phenotype data, before it can begun to be considered for protein function and phenotype prediction tasks.

This included introduction of some of the infrastructure of databases and software that the fields of genomics, bioinformatics and computational biology are built on. While in other fields, data inaccessibility is a major barrier to reproducible research, this is the field that had an online database system that remote computers could access in the 1960s! Huge quantities of catalogued information collected by researchers around the world populates freely available databases, vocabularies, and annotations, creating controlled and shared vocabularies that fuel computational methodologies. This chapter also briefly considered some of the potential sources of error in bias in these data, and attempts to overcome them.

With such a treasure trove of data, from model organisms as well as humans, there is more opportunity than ever for this data to be used to answer some of biology's big questions, such as making genome-wide phenotype predictions. Multi-omics approaches that combine data types have already been successful at elucidating mechanisms behind certain phenotypes[162,163,164].

We should not forget that obtaining an accurate prediction of phenotype and protein function even for a small class of variants, has the potential to greatly impact people, particularly if the prediction is explanatory, e.g. pointing to specific variants or protein domains as the cause of the phenotype. Determining to what extent this data can currently be used for this purpose is the subject of the rest of this thesis.

4. Phenotype prediction with Snowflake

This chapter describes the Snowflake algorithm for phenotype prediction that I developed in collaboration with Jan Zaucha, Ben Smithers and Julian Gough. The development of `snowflake` resulted in a patent[4], of which I am an author, and later a paper[165] (the latest iteration of the tool is now called `Nomaly`). This chapter deals with the functionality and design of the Snowflake algorithm and it's application to the ALSPAC dataset.

At it's heart, Snowflake is a CLI (Command Line Interface) tool and private Python package that allows the user to detect outliers for each phenotype of interest, according to their genotype. Individuals with unusual combinations of variants in highly conserved protein domains associated with a phenotype will score highly for (be indicated as likely to have) a phenotype.

The original idea for Snowflake was Julian's, as well as the initial Perl implementation. The initial translation of the code from Perl to Python was carried out by Ben. Working from Ben's translation, Jan and I both worked on increasing the algorithms functionality and robustness together, before forking the project into two different versions which we each took ownership of.

> 💡 **Contributions in this chapter**
>
> - Writing part of the patent[4] relating to intrinsic dimensionality.
> - Software development to increase and test the algorithm's functionality, including:
>   - With Jan and Ben:
>   - Running with different formats and numbers of inputs and background cohorts
>   - Dealing with missing calls
>   - Development of tools to create input files for Snowflake
>   - Improvements to memory-usage and speed
>   - And individually:
>   - Creation of inputs to Snowflake
>   - Alternative clustering and scoring methods, particularly for intrinsic dimensionality
>   - Confidence score
>   - Scoring outputs
>   - Further improvements to speed and memory usage
>   - Multiple imputation for missing calls
>   - Inclusion of dimensionality reduction
>   - Testing Snowflake on the ALSPAC cohort

4.1. Introduction

The Snowflake algorithm is primarily a phenotype prediction method: it takes data about individuals DNA as input and outputs predictions about which phenotypes each individual has. These predictions are based on how unusual an individual is for variants relating to each phenotype, and are made across a breadth of phenotypes and for missense variants across the protein-coding genome. It does this by combining existing predictions of variant deleteriousness from FATHMM[123] and association of protein domains to phenotypes from DcGO[119], and finding unusual combinations of these variants through clustering individuals against a diverse background cohort and looking for outliers. The phenotype prediction implicitly contains protein function predictions, due to the relationship between protein function and phenotype, and these are the key output of Snowflake. Focusing on protein domains thereby enables predictions in proteins that have not been well-studied, but restricts the number of predictions that Snowflake can make (since phenotypes can be caused by mutations which fall outside of domains). As a protein function predictor, Snowflake seeks rare combinations of SNPs which may influence a phenotype. In other words, Snowflake creates explanatory predictions: it looks for the mechanisms behind complex traits. Such complex traits are currently not well understood, but are thought to cause many human diseases.

4.1.1. Motivation

In chapter 2, we discussed the theoretical mechanism from which phenotype arises from genotype. In summary: differences in DNA cause differences in cell functionality, which interact with the cell environment to create differences in overall phenotype.

As previously mentioned, many recognised phenotypes are medical disorders or their symptoms. Currently to achieve diagnoses for genetic illnesses, specific genes are often sequenced one at a time, since looking at whole genomes would be too time consuming for clinical staff. Patients seeking diagnoses for rare genetic diseases describe the process as an "Odyssey", more than half undiagnosed at any given time. If whole genome (or genotype) based phenotype prediction was possible, only one sample and test would be needed to get a much fuller picture of a person's health, and we would be able to reduce the long and tiring process of obtaining diagnoses for rare genetic diseases. Applied to the plant and animal kingdom, phenotype prediction could also be beneficial in veterinary science and agriculture.

The discovery of underlying mechanisms for complex traits remains a particular challenge. Each prediction in snowflake can be explained by which protein the variant is in, why that protein is predicted to be deleterious, and how common or rare that variant is.

4.1.2. Related work

4.1.2.1. Phenotype predictors and variant prioritisation

Biology databases are home curated, open data that cover genomes across the tree of life, as well as cross-species ontologies of biological processes, diseases, and anatomical entities. There are a number of recent phenotype prediction methods that have had some success in using these resources for either variant prioritisation or use as a clinical diagnostic tool.

There are a class of "knowledge-based" methods, which use knowledge from databases of experimental results (known associations between genes and phenotypes) as the basis for these predictions, for example Phen-Gen[166], dcGO[119], PhenoDigm[167], and PHIVE[168]. The better performing methods in this class, use associations between model organisms and orthologous genes, to leverage the wealth of information that is collected from these model organism experiments.

There are also "functional" methods, like FATHMM[123] and CADD[169], which instead use information about how the molecules and their function may change with different nucleotide or amino acid substitutions, as well as conservation metrics to prioritise variants. These tools rank variants for deleteriousness, but do not link them to specific phenotypes.

Most successful methods of any kind now combine multiple sources of information, some combine both functional and knowledge-based sources. This approach is used within Exomiser[170], which combines PHIVE with many other sources of information such as protein-protein interactions, cross-species phenotype associations, and variant frequency using a black-box classifier. Phenolyzer[171] and Genomiser[172] also take similar approaches of combining many different sources of data.

The aim of these models is mostly to prioritise variants associated with diseases, and they are bench-marked by their ability to identify known variants. Lists of known variants may be purpose-curated from the literature according to specific evidence, or may come from some subset of annotation databases (which in some cases the algorithm may have used as input data). Each phenotype predictor often targets a specific use case (e.g. non-coding variants), and in combination with the varying validation methods used, it is difficult to compare the accuracy of all of these models directly. For this reason, the CAFA competition is very useful in getting a more objective view of the capabilities of these kinds of tools.

Similar approaches have been used as clinical diagnostic tools. PhenIX (Phenotypic Interpretation of eXomes)[173] is a version of PHIVE which is restricted to the "human disease-causing genome" (genes known to cause disease) to make it more suitable for clinical use in diagnosis of rare genetic diseases. It also includes semantic similarity information between inputted symptoms and Human Phenotype Ontology terms, using the Phenomizer[174] algorithms. For PhenIX the measure of success is that it enabled skilled clinicians to find diagnoses for 11 out of 40 (28%) patients with rare genetic diseases, who were not able to be diagnosed through other means.

While these examples are the most similar published work to Snowflake, they are all tested as variant prioritisation tools rather than phenotype predictors.

4.1.2.2. Clustering and outlier-detection in genetics

Clustering algorithms, particularly hierarchical methods, are commonly used in genetics for:

1. finding evolutionary relationships between DNA samples, for example, in reconstructing phylogenetic trees and mapping haplotypes within populations[175].
2. finding functional relationships between genes based on gene expression data[176,177].

For applications in group (1), individuals are generally separated into clusters based on their DNA variants, whereas for (2) samples are separated into clusters based on their gene expression.

Haplotypes and haplogroups

Haplotypes are groups of alleles that are inherited together from a single parent. In humans, Y-chromosomes and mitochondrial DNA (mtDNA) are often analysed for haplotypes to understand ancestry within species since these are passed from parents to children without recombination (only mothers pass on mtDNA, and only fathers pass on Y chromosomes).

Haplogroups are groups of haplotypes, representing major branching points in the within-species phylogenetic tree.

Clustering methods are only very rarely used to cluster individuals in phenotype prediction or variant prioritisation tasks. In one case, clustering individuals based on a combination of genotype and phenotype information has been applied to identify subtypes within emphysema[178] (a lung disease).

4.1.2.3. Overcoming the curse of dimensionality through dimensionality reduction and feature selection

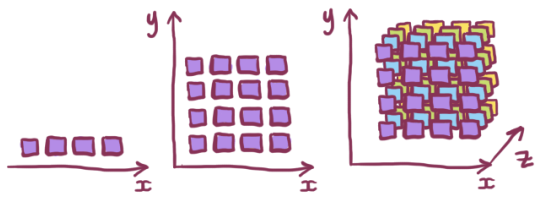


Fig. 4.1 As the number of dimensions increase from 1 dimension on the left to 3 on the right, the number of points needed to cover the space increases exponentially. This means that for a fixed number of points (individuals), increasing the number of dimensions (SNPs) that we cluster on decreases the density of our space exponentially, making it exponentially harder to identify clusters.

The "curse of dimensionality" is a phrase coined by Richard E Bellman, during his work on dynamic programming[179], but has since proved relevant in many different mathematical and data-driven fields. While it's used colloquially as a catch-all complaint about high-dimensional data, the "curse" specifically refers to the sparsity of data that occurs exponentially with an increase in dimensions. This leads to various problems in different fields[180] including general difficulty in reaching statistical significance and reduced usefulness of clustering, distance, and outlier metrics. This can easily be a problem in genetics since we have tens of thousands of genes, and hundreds of thousands of variants as dimensions that we may want to cluster over.

As Fig. 4.1 illustrates, increasing the number of dimensions that we cluster over makes it exponentially harder for us to identify clusters in the data given a fixed number of individuals in our cohort. In extreme cases, all samples or individuals look equally distant from each other in the sparse, high-dimensional space.

The curse of dimensionality can be partially reduced by choosing a clustering or outlier detection method which is more robust to the number of dimensions. However, these still have limits, and in order to overcome these, it is necessary to reduce the number of dimensions in some way, this process is called feature selection. This can be done through careful curation of important features, through variance cut-offs, or by dimensional reduction methods like Prinipal Component Analysis (PCA) or Multi-dimensional Scaling (MDS) which project the data into a different coordinate system and then discard some of the newly calculated dimensions.

4.2. `Snowflake` Algorithm

This section introduces the Snowflake algorithm, and gives an overview of how it works, as well as a description of it's functionality and which parts of this I contributed to.

4.2.1. Approach

Snowflake belongs to a small number of phenotype prediction methods that aim to predict across many phenotypes and many genotypes. Although it's designed primarily as a *phenotype* prediction algorithm, it also implicitly makes *protein function* predictions. Snowflake calculates a score for each phenotype for each individual (according to their genotype), and a cut-off is then used to convert these scores into binary predictions.

In contrast to the black-box approaches that are currently most successful in terms of accuracy, Snowflake aims to create **explanatory** predictions. This feature means that Snowflake has more utility in discovering new genetic mechanisms for disease. For any prediction (high-scoring genotype-phenotype pair according to Snowflake), Snowflake also reports which SNPs contributed to the prediction, how deleterious the SNP substituion is, and how rare the mutations are in the population. Such an explanation is only possible because Snowflake restricts itself to missense mutations, since the deleteriousness is calculated by FATHMM from this information. In summary, Snowflake's design makes it more useful for finding candidate mutations that are responsible for phenotype, not for accurately predicting an individual's phenotype.

The Snowflake phenotype prediction method works by identifying individuals who have unusual combinations of deleterious missense SNPs associated with a phenotype. The phenotype predictor uses only data about missense SNPs in coding regions of globular proteins, so it can only be expected to work well where phenotypes are determined primarily by these kinds of mutations.

This method combines conservation and variant effect scores using FATHMM[123], inference about function of protein domains using dcGO[119], and human genetic variation data from the 2500 genomes project[181] to predict phenotypes of individuals based on their combinations of missense SNPs.

4.2.2. How does it work?

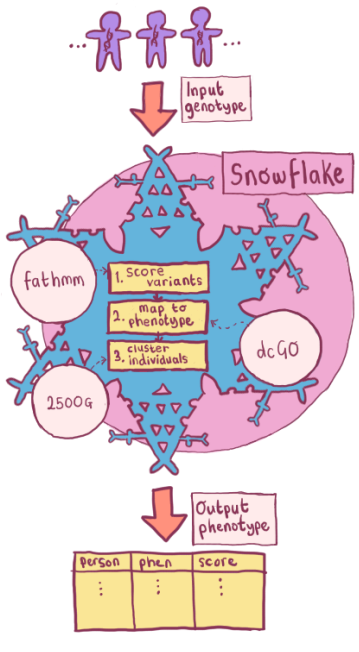


Fig. 4.2 Flowchart showing an overview of the phenotype predictor. Scores are generated per allele using SUPERFAMILY, VEP, FATHMM and dcGO for both the input genotype(s), and the background genotypes. These data points are then combined into a matrix, which is then clustered.

Fig. 4.2 shows how an overview of how Snowflake operates. One or more genotypes (in VCF or 23andMe format) are needed: this is the *input cohort*. The algorithm can then be divided into four main steps:

1. Score variants for deleteriousness, using FATHMM and VEP

- This is the process of creating the `.consequence` files.

1. Map variants to phenotype, using dcGO and SUPERFAMILY

- For each phenotype i, a list of SNPs is generated such that the SNPs are associated with the phenotype (according to dcGO), and the SNPs are present in the input individuals. These list of SNPs is stored in `.snp` files.

1. Cluster individuals using OPTICS per against all others and (optionally) a diverse background set from the 2500 Genomes Project[181].

- Cluster using SNPs as features.

1. Extract a score and prediction.

Further detail on these steps is provided below.

Step 1 and 2 are actually carried out through creating Snowflake inputs, which I describe in greater detail it's own section.

Each of these steps is modular, meaning that it's possible to use another method to predict the deleteriousness of variants, to map variants to phenotype, or to cluster individuals or to extract the score.

4.2.2.1. SNPs are mapped to phenotype terms using DcGO and dbSNP

DcGO[119] is used to map combinations of protein domains to their associated phenotype terms, using a false discovery rate cut-off of 10^{-3} or less. SNPs are therefore mapped to phenotype terms by whether they fall in a gene whose protein contains domains or combinations of domains that are statistically associated with a phenotype. In order to do this, DcGO makes use of SUPERFAMILY[118] domain assignments, and a variety of ontology annotations (GO[106], MPO[182], HP[183], DOID[184] and others).

Using DcGO means that phenotypes are only mapped to protein if the link is statistically significant due to the protein's contingent domains. This leaves out some known protein-phenotype links, where the function may be due to disorder for example rather than protein domain structure.

Known phenotype-associated variants are therefore added back in using dbSNP[66].

4.2.2.2. SNPs are given deleteriousness scores using FATHMM

The phenotype predictor uses the unweighted FATHMM scores[123] to get scores per SNP for the likelihood of it causing a deleterious mutation. This is based on conservation of protein domains across all life, according to data from SUPERFAMILY[118] and Pfam[185].

This method gives SNPs the same base FATHMM score for being deleterious, no matter which phenotype we are predicting them for. It is therefore the combination of SNPs per phenotype, and their rarity in the population that determines the phenotype prediction score.

4.2.2.3. Comparison to a background via clustering

Individuals are compared to all others through clustering. This usually includes comparing each individual to the genetically diverse background of the 2500 genomes project[181].

Clustering is the task of grouping objects into a number of groups (clusters) so that items in the same cluster are similar to each other by some measure. There are many clustering algorithms, but most are unsupervised learning algorithms which iterate while looking to minimise dissimilarity in the same cluster. A number of options were implemented for the predictor, but OPTICS is used as a default for theoretical reasons, which I describe in the next section.

4.2.2.4. Phenotype score

The OPTICS clustering assigns each individual to a cluster (or labels them as an outlier). Depending on the phenotype term, the cluster is expected to either correspond to a haplogroup or a phenotype. In cases where the cluster refers to a haplogroup, we are interested in the outliers of all clusters, i.e. the local outlier-ness. In cases where the cluster is the phenotype, we are interested in the outlying cluster, i.e. the global outlier-ness.

A local score, L_{ij} can be defined as the average Euclidean distance from an individual to the centre of it's cluster, or for individuals that are identified as outliers by OPTICS, 2 multiplied by the distance to the centre of the nearest cluster.

A global score G_{ij} can be defined as the distance of the cluster to the rest of the cohort.

The global-local score is designed to balance these sources of interest. It sums the two scores, adjusting the weighting by a cluster size correction factor, μ_γ:
$GL_{ij} = L_{ij} + \mu_\gamma \cdot G_{ij}$

Such that: $\mu_\gamma = \frac{exp(\gamma\frac{n-n_j}{n})-1}{exp(\gamma)-1}$ where γ is a parameter representing how strongly we wish to penalise large clusters, n is the over all number of individuals and n_j is the number of individuals in a cluster.

The global-local score was inspired by the tf-idf score popular in Natural Language Processing bag-of-word models.

TF-IDF

Term Frequency, Inverse Document Frequency is a common and basic measure in NLP which attempts to measure how representative a term (word) is of a document. It is defined by $tfidf = tf(t,d) \cdot idf(t,D) = (f_{t,d}) \cdot (\frac{N}{absd \in D : t \in d})$ where $f_{t,d}$ is the frequency of a term t in a document d, N is the number of documents, and \({abs{d \in D : t \in d}\) is the number of documents containing the term.

The global-local score is always greater than zero, but it has no upper limit because the largest possible score than an individual can have depends on the number of high-scoring SNPs and the magnitude of the FATHMM score. For this reason, the global-local score is transformed to a score that is comparable between phenotypes, using a formula first introduced in the earlier Proteome Quality Index paper[2]:

$$s_{trans}(p) = \sqrt[3]{\frac{e^{-\left(1+\frac{s_{rank}(p)}{N}\right)^{\varphi}}}{e^{-\varphi}} \cdot \frac{s(p)}{\sum_p s(p)} \cdot \frac{s(p) - \min s(p)}{\max s(p) - \min s(p)}}$$

Where $s_{trans}(p)$ is the transformed score for a person, $s(p)$ is the original score, s_{rank} is the rank of a person within a phenotype, and φ determines the importance of rank.

The transformed score is usually negative, with a small number of scores per phenotype being 0 or slightly above, which are interpreted as positive predictions.

4.2.3. Functionality

`Snowflake` is implemented as a CLI (Command Line Interface) tool, written in Python with the following commands:

- `snowflake create-background`
- `snowflake create-consequence`
- `snowflake preprocessing`
- `snowflake predict`

The functionality described above all happens within `snowflake predict`, but in order to use `snowflake predict`, there are also three commands which create files needed to run the predictor.

4.2.4. Features added to the predictor

As mentioned, the phenotype predictor was already prototyped when I began working on it. However, considerable time was spent developing, bug-fixing, and extending this prototype. Here, I describe my contributions to this in detail.

4.2.4.1. Different running modes

The original version of the phenotype predictor could only be ran one individual compared to a background set at a time. In order to allow for a wider range of inputs (which will be necessary to validate the predictor), support for a wider range of genotype formats and running modes was developed, including:

- Can be run with one person against a background
- Can be run with multiple people (VCF) against the background
- Can be run with or without the background set if there are enough people in the input set.
- Support for different 23andMe genotype file formats (from different chips).

As the predictor was developed to perform in different running modes, it became clear that it would be necessary to streamline the algorithm. This included parallelisation (possible due to the independence of different phenotype terms), and various data storage and algorithm adjustments.

Implementing these running modes and increases in efficiency was a collaborative effort between myself, Jan, and Ben.

4.2.4.2. Adding SNP-phenotype associations from dbSNP

As mentioned in the overview, using DcGO as the only SNP-phenotype mapping leaves out some known associations that are not due to protein domain structure. Adding dbSNP[66] associations to the predictor was one of my contributions to this software.

4.2.4.3. Dealing with missing calls

Genotyping SNP arrays often contain missing calls, where the call can not be accurately determined. This is an obstacle to the phenotype predictor if left unchecked as it can appear that an individual has a very unusual call when it is really just unknown. Since most people have a call, the missing call is unusual, and this is flagged.

The most sensible solution to this problem is to assign the most common call for the individual's cluster (i.e. combination of SNPs). This prevents a new cluster being formed or an individual appearing to be more unusual than they are. However, there is a downside to this approach when there are many missing calls. Adding all missing calls to a cluster that was only slightly more common than the alternatives can lead to the new cluster containing the missing data dwarfing the others. To fix this, SNPs with many missing calls were discarded.

Alternatives such as assigning the most common call for that SNP only, or assigning an average score for that SNP dimension by carrying out a "normalised cut"[186] are untenable since they can create the same problem we are trying to overcome: the appearance of an individual having an unusual combination of calls.

4.2.4.4. Reducing dimensionality

Some phenotypes have large numbers of SNPs associated with them - too many to assign individuals to clusters. I added a dimensionality reduction step in the clustering, and tested different clustering methods designed for use on high-dimensional data. These changes are explained later in this Chapter.

4.2.4.5. Confidence score per phenotype

The phenotype predictor outputs a phenotype score for each person for each phenotype. Our confidence in these scores depends on the distribution of scores, as well as the scale of them. A distribution of scores with distinct groups of individuals is generally preferable, since most phenotypes that we are interested in are categorical or it is at least more useful to highlight phenotypes that can be predicted this way (i.e. if there are 100 groups with varying risk of a disease, that would be less useful than knowing there are 2 groups with high/low risk).

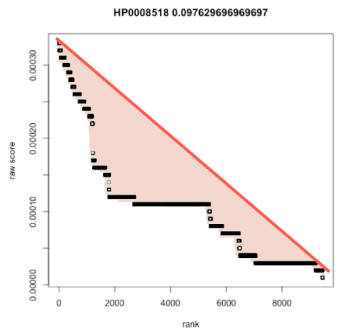


Fig. 4.3 An illustration of how the confidence score per phenotype is calculated.

I developed a simple method of prioritising predictions according to these requirements. A confidence score is achieved by plotting the ranked raw score and measuring the area between a straight line resting on this the most extreme points and the line itself, as illustrated in Fig. 4.3. Since this measure takes into account the size of the raw scores, these confidence scores can be compared across phenotypes.

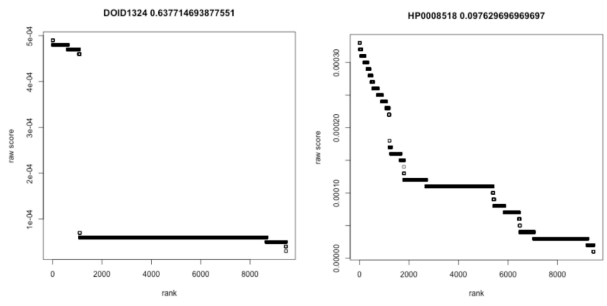


Fig. 4.4 Ranked scores for `DOID:1324` - the disease ontology term Lung Cancer (left) and HP:0008518 - the human phenotype ontology term for Absent/underdeveloped sacral bone (right). These represent an interesting and uninteresting distribution of scores, respectively.

Fig. 4.4 shows an example of an interesting and uninteresting distribution. These distributions mostly depend on the number, population frequency, and FATHMM score of the SNPs associated with the phenotype term.

4.3. Creating Snowflake inputs

This section describes the sources and pipelines for creating inputs to Snowflake, including necessary files that are packaged alongside it.

There are four main inputs required by Snowflake which must be created:

1. dcGO phenotype mapping file: high-quality mappings between protein (supra)domains and phenotype terms from 16 biomedical ontologies - provided as part of Snowflake
2. Background cohort: genetic data of diverse individuals (`.vcf`, VCF format) - default human option provided as part of Snowflake
3. Consequence file: deleteriousness scores per SNP (`.Consequence`, format specific to `snowflake`) - default human option provided as part of Snowflake
4. Input cohort: genetic data of person or people of interest (`.vcf`, VCF format) - provided by user

The dcGO mapping file works for all organisms and only needs remaking if dcGO or SUPERFAMILY have significant updates. Alternatively, a slimmed version can be created that contains only the ontologies of interest to a specific organism which reduces Snowflake's running time, and here I explain how I made a slimmed version of this file for ontologies of interest to humans. The background cohort and consequence file must be created once for each organism/genome build. Here, since I have only used Snowflake in predicting human phenotypes, I walk through the creation of the human background cohort and consequence file. Lastly, the input cohort file must be created per input cohort. Here I run through the creation of a VCF file from 23andMe genotype files.

> **Consequences of input cohort SNPs**
>
> It is possible to create a consequence file for any cohort VCF file, i.e. it would be possible to make an input cohort consequence file. However, since Snowflake can only cluster our input cohort against the background for SNPs which overlap between the two cohorts, we only need to create one such file for either the input or the background. Here I've chosen to do it for the background cohort since then this file can be reused for many different input cohorts.

4.3.1. DcGO phenotype mapping file (human)

It is simple to create the dcGO mapping file since dcGO provides the required files for download on the SUPERFAMILY website. The website provides 2 files of high-coverage mappings for each ontology supported by dcGO: one that maps between SCOP domains and ontology terms, and another that maps between SCOP supradomains (combinations of domains) and ontology terms. Currently, Snowflake does not support the inclusion of dcGO supra-domain assignments.

> **High-coverage versus high-quality**
>
> DcGO provides two versions of it's GO mappings: high-coverage and high-quality. The high-quality mapping contains only those associations between domains and phenotypes that are supported by single-domain proteins, while the high-coverage version contains some associations that have been inferred from known associations to multi-domain proteins. The high-coverage mapping contains roughly 10 times as many GO terms and 10 times as many protein domains compared to the high-quality version.
>
> For all other (non-GO) ontologies, dcGO provides only the high-coverage version.
>
> The high-coverage mappings were used in Snowflake to increase the coverage of the phenotype predictions, i.e. so that more SNPs could be included and more phenotypes.

The ontologies that contain interpretable phenotype terms for humans are:

- Disease Ontology[184] (DO)
- Human Phenotype Ontology[112] (HP)
- Gene Ontology[106] (GO), specifically terms from the `biological_process` subontology.
- MEdical Subject Headings[188], (MESH) - on the dcGO website this is found under CTD diseases
- Mammalian Phenotype Ontology[187] (MP)

Some of these ontologies aren't designed only (or even primarily) for humans, but since they contain terms which are relevant to humans and the associations are made based on protein domains found in human proteins, these ontologies were chosen. While dcGO supports many other ontologies (e.g. Zebrafish and Xenopus ontologies) based on protein domains that can be found in humans, I made the decision that the trade-off between the additional time needed to run the phenotype predictor and sift through the results of these ontology terms was not worth the increase in coverage gained from whichever of these terms were meaningful.

Incusion of the Mammalian Phenotype ontology

The Mamalian Phenotype (MP) ontology[187] is based on phenotypes seen in mice. While mice do have many anatomical and phenotypic similarities to humans, and are thus often used as models for human diseases, there are of course differences and some terms don't directly translate or are likely to be defined differently for humans (e.g. `MP:0001933` *abnormal litter size* or `MP:0002068` *abnormal parental behaviour*) and have different mechanisms behind them. There are also many terms which do make sense for humans (e.g. `MP:0002166` *altered tumour susceptibility* or `MP:0011117` *abnormal susceptibility to weight gain*) and may represent interesting results. The suitability of MP terms must be considered term-by-term, but the terms were included in the human phenotype mapping file to increase coverage of both proteins and phenotypes.

To create the Snowflake input for humans, files for the five relevant ontologies were downladed (for DO, HP, GO, MESH, and MP), then concatenated, and the GO cellular component and GO molecular function terms were removed (these term identifiers were extracted from the `GO_subontologies` field of the `Domain2GO_supported_only_by_all.txt` file).

domain_type	domain_sunid	ont_term_id	ont_term_name	ont_subontologies	information_content	annotation_origin_1direct_0inherited	ont_id
fa	55528	HP:0000925	Abnormality of the vertebral column	Phenotypic_abnormality	1.123852	0	HP
sf	53822	MESH:D013568	Pathological Conditions, Signs and Symptoms	CTD_diseases	0.581857	0	MESH
sf	48619	GO:0006658	phosphatidylserine metabolic process	biological_process	2.596963	1	GO_BP

Fig. 4.5 An excerpt of the dcGO mapping file `human_po.txt`, showing mappings between phenotype terms from a range of ontologies, and SCOP sun IDs.

Fig. 4.5 shows the type of content inside the phenotype ontology file. the `information_content` field is a measure of how specific the phenotype term is, for example *Pathological Conditions, Signs and Symptoms* is a very general term, while *Abnormality of the vertebral column* is more specific and *phosphatidylserine metabolic process* is the most specific shown.

Ontology	Number of dcGO assignments	Number of unique terms assigned	Number of unique domain `sunid`s assigned
HP: Human phenotype ontology	**15984**	**1978**	**562**
MeSH: Medical Subject headings	**5761**	**620**	**473**
DO: Disease ontology	**7538**	**614**	**566**
GOBP: Gene ontology, biological process	**304064**	**9546**	**3135**
MP: Mammalian phenotype ontology	**24208**	**2528**	**719**
Total (all included ontologies)	**357555**	**15286**	**3146**

Table 4.1 Summary statistics of the coverage of human-related phenotype terms.

Table 4.1 shows the number of assignments, unique phenotype terms, and unique domains covered in the dcGO phenotype mapping per ontology and in total. The file contains **357555** assignments, **15286** unique ontology terms, and **3146** unique domains. This constrains the proteins and phenotypes which Snowflake can make predictions about.

4.3.2. Background cohort

As described in the overview, Snowflake requires genetic "background" data to compare the individuals we are interested in against, i.e. so that meaningful clustering can take place. Although Snowflake has the functionality to be run with any background data set, the choice of data set is constrained since it must contain representative genetic data from from the entire population. The 1000 Genomes project[181,189] was therefore the perfect choice as it was designed to map common human genetic variation across diverse populations, despite being much smaller than some other cohorts available, such as UK Biobank[190].

4.3.2.1. Data acquisition: the 1000 Genomes project

The 1000 Genomes project[181,189] ran from 2008 to 2015, with the aim of comprehensively mapping common human genetic variation across diverse populations. The project sequenced individuals whole genomes, and released data in two main phases:

- Phase 1: 37.9 million variants, in 1092 individuals, across 14 populations
- Phase 3: 84.4 million variants, in 2504 individuals, across 26 populations

Data from the 1000 Genomes project are always used for the background cohort to Snowflake, with data from the 1000 Genomes project Phase 3[181] used as a default, and earlier experiments using data from Phase 1[189]. Here I describe the process of creating the phase 1 VCF file (`1000G.vcf`), but the same process is followed to create the larger phase 3 VCF file (`2500G.vcf`).

VCF

When individual humans have their whole genomes sequenced, this is compared to the human reference genome. The alleles at each location are commonly stored in Variant Call Format (VCF) files; a much more compact format compared to storing the entire genome. VCF files describe the locations on the genome of variations between individuals, given by chromosome, position, variant identifiers (e.g. rsID), and then the calls at those locations for each individual.

For both phases of the 1000 Genomes project, data are provided as VCF files for each chromosome. Both data sets are available through the International Genome Sequencing Resource[191] (IGSR); phase 1 VCFs (GRCh37) can be downloaded via FTP at `ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase1/analysis_results/integrated_call_sets/`, and phase 3 VCFs (GRCh37) can be downloaded at `ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/`.

After downloading these files, the per-chromosome VCF files were combined into one large VCF file (for all chromosomes). The consequence file must be created at this stage, before the final input VCF can be created.

4.3.2.2. Create final input VCF

In order to create the final input VCF, we must remove SNPs from the VCF file that are not in the consequence file. This reduces the file size dramatically, since many recorded SNPs are either synonymous, nonsense, non-coding or multi-allelic. The VCF file is then Tabix-indexed[192], which increases the speed of Snowflake.

4.3.3. Consequence file

The consequence file contains the "consequences" of one amino acid being changed to another, which is used to weight which SNPs are the more important dimensions for clustering a phenotype. It also contains the mapping between SNPs and proteins, and SNPs and domain architecture.

This file is generated using:

- Ensembl's VEP (Variant Effect Predictor) tool to map from chromosomes and locations to SNP type (e.g. missense/nonsense/nonsynonymous), and to protein ID
- FATHMM for scoring the deleteriousness of missense mutatations
- SUPERFAMILY for mapping from protein IDs given by VEP to their domain architectures (SCOP family/superfamily)

4.3.3.1. Run the Variant Effect Predictor tool

In order to get a list of SNPs to input to FATHMM, we must first determine which SNPs in the background data are missense SNPs. This can be done using Ensembl's VEP (Variant Effect Predictor) web tool[125], which takes a VCF as input. This first 10 columns of the combined VCF file is used as input to VEP since only these columns are needed, and making the file smaller reduces processing time.

4.3.3.2. Query FATHMM and SUPERFAMILY for the SNPs of interest

To get the consequence file, VEP output was first filtered to contain only biallelic missense SNPs, and was then used as input to query the FATHMM and SUPERFAMILY databases for the unweighted conservation scores and the domain assignments.

4.3.3.3. Summary

#CHROM	POS	calls	snp_id	ENSP_id	prot_sub	HMM	position	ref_prob	mut_prob	SUPERFAMILY	Sup_e_val	FAMILY	Fam_e
1	**69224**	A/T	NaN	ENSP00000334393	D45V	SF0037432	44	0.524940000001	4.26011	81321	2.93e-78	81320	0.(
	1290695	G/C	NaN	ENSP00000307887	T136S	SF0042359	95	1.83872	2.56474	48726	3.14e-10	48727	0.0
	1290695	G/C	NaN	ENSP00000344998	T35S	SF0047556	61	1.72866	2.45993	48726	0.000485	49159	0
	1290695	G/C	NaN	ENSP00000399229	T136S	SF0042359	95	1.83872	2.56474	48726	3.29e-10	48727	0.0
	3392588	T/C	NaN	ENSP00000367629	Y479H	SF0040099	5	2.69907	4.0507	50729	3.73e-18	50730	0
	3392588	T/C	NaN	ENSP00000367629	Y479H	SF0050917	5	2.69907	4.0507	48065	3.53e-57	48066	8.09e
	3392588	T/C	NaN	ENSP00000408887	Y183H	SF0040099	5	2.69907	4.0507	50729	1.32e-18	50730	0
	3392588	T/C	NaN	ENSP00000408887	Y183H	SF0050917	5	2.69907	4.0507	48065	3.27e-38	48066	0.00

Fig. 4.6 An excerpt of the consequence file `2500G.consequence`, showing mappings between SNPs, protein IDs, mutant and reference probabilities from FATHMM, and SCOP sun IDs (via SUPERFAMILY).

As we can see in An excerpt of the consequence file 2500G.consequence, showing mappings between SNPs, protein IDs, mutant and reference probabilities from FATHMM, and SCOP sun IDs (via SUPERFAMILY)., the consequence file can contain multiple rows for the same SNPs since the SNP may appear in multiple proteins, since proteins can have overlapping reading frames. Less frequently, SNPs that fall in multiple proteins may also fall in more than one domain families or even superfamilies.

The output `.consequence` file defines the upper limit of the number of SNPs Snowflake can make phenotype predictions based on. In practice, this will often be a much smaller number as for SNPs to be used they must be:

- measured in the input cohort (which is often genotyped, and therefore contains far fewer variants)
- within protein domains that exist in the dcGO mapping file in addition to the consequence file.
- a SNP where there is some variation between the background and input

Like the VCF file, the Consequence file is the indexed with Tabix[192] to increase Snowflake's speed.

4.3.4. Input cohort

4.3.4.1. 23andMe file formats

The SNPs which are measured in a genotyping experiment depends on the chip used. Since launch, 23andMe have used a number of different Illumina chips for their genotyping service. These chips capture information for different SNPs, and vastly different numbers of SNPs. This means that in order to combine data from different chips, many loci can not be used.

23andMe have their own file formats, which must be convered to VCF in order to use Snowflake. One of these formats is tab separated and very similar to VCF, but for their API, 23andMe also stored genotype data in long strings (~1 million characters, twice the number of loci on the chip) of the form `AAAACCTTTT__CC__`, where every 2 nucleotides corresponds to a given SNP on the 23andMe chip with a rsID, chromosome and position. To convert this compressed format to VCF, 23andMe provided a genotype snp map file (`snps.data`), which is different for each 23andMe chip, that gives rsIDs, chromosome and position for each index, which looks like:

```
# index is a key for the /genomes/ endpoint (2 base pairs per index).
# strand is always +1.
index   snp     chromosome      chromosome_position
0       rs41362547      MT      10044
1       rs28358280      MT      10550
2       rs3915952       MT      11251
```

So, according to this file, the first two characters of the genotype string correspond to `rs41362547`, then the next two correspond to `rs28358280`.

The `create_data` function of snowflake contains the functionality to create a VCF file from 23andMe API string formats, for three different chips.

4.3.4.2. Genome builds

The files currently generated for snowflake use the GRC37 (hg19) human genome build. This is the version that is currently supported by FATHMM, 23andMe, and phase 1 of 1000G.

> **Position and BED formats**
>
> The positions of the variants (SNPs) that are recorded can vary in format: essentially this comes down to a change in formatting, the fact that sometimes counting starts at 0, and sometimes it starts at 1, and sometimes ranges are closed on one side e.g. `chr1 127140000 127140001` in BED format is equivalent to `chr1:127140001-127140001` in "Position" format[193].

4.4. Preprocessing

In the preprocessing stage, the `snowflake preprocess` command looks at all the inputs together, in order to filter them only for the useful parts before running the predictor.

In particular, `snowflake preprocess` always calculates the following in all running modes:

- A list of SNPs associated with each term (`.snp` files)
- Sets of equivalent terms, i.e. terms which have the exact same set of SNPs associated with them (`.polist` files)

And it also calculates the following if an input cohort is provided:

- A list of overlapping SNPs between the background and input cohort
- A combined VCF file containing only these snps, including dealing with ambiguous flips.

In this section, I will run through and explain the preprocessing step for the 1000 genomes only (no input cohort) as this represents an approximation of the maximum number of SNPs that `snowflake` can predict on, since the 1000 genomes project uses WGS (Whole Genome Sequencing).

4.4.1. Combining VCF files, a.k.a. missing SNPs and ambiguous flips

Due to the cost, far more humans have been genotyped than have had their whole genomes sequenced. Genotyped and WGS (Whole Genome Sequencing) data look similar once in a VCF file, but the data cannot necessarily be treated the same in both cases.

4.4.1.1. Missing SNPs in VCF files

Many VCF files only store the differences between individuals in the file, a SNP being missing from a VCF file does not necessarily mean that the original sequencing or genotyping didn't record the calls at that position.

If combining two genotyped files, we would want to discard all SNPs that are not measured by both chips, but when combining a genotype VCF file with a WGS VCF file, we usually want to keep all SNPs from the genotyped VCF (since these locations will also have been sequenced by WGS).

4.4.1.2. Ambiguous Flips

The majority of input data to the predictor is 23andMe data. In testing earlier versions of Snowflake with the 2500G background and a cohort of 23andMe genomes, it became clear that for many phenotypes, the background was forming a separate cluster to the cohort. This led to the realisation that there are 23andMe calls which had the opposite ratio of wild type:mutant than the 2500 genomes. Some further reading revealed this to be a known problem[194], which may be due to ambiguous flips[195].

Implausible distributions of SNPs in the input cohort (given the background) are therefore discarded using a cutoff.

4.5. Considerations for Clustering SNPs

In this section, I discuss the implementation of clustering and outlier detection methodologies for SNPs.

Snowflake is essentially a method for finding unusual combinations of SNPs, within sets of SNPs associated with a phenotype. Since there will be many rare combinations of SNPs just through randomness, the deleteriousness score (via FATHMM) decides which rare combinations are of interest.

The clustering takes place per phenotype, with the distinct phenotypes and the SNPs associated with them are chosen according to dcGO. This therefore determines how many dimensions (SNPs) we are clustering on for any given phenotype

4.5.1. Combinations of SNPs

Given N SNPs of interest, there are 3^N different options for individual's combinations of calls for biallelic SNPS, since there are three different options for each SNP `WW` wild-wild homozygous, `MW`/`WM` heterozygous, or `MM` mutant-mutant heterozygous.

For our purposes, heterozygous SNPs are considered the same whether they are mutant-wild `MW` or wild-mutant `WM`, since we assume they would create the same balance of proteins in a cell.

> **Linkage Disequilibrium**
>
> Linkage disequilibrium is the measure of how much more often alleles are found together than would be expected if they were randomly distributed. SNPs are not independent, and we wouldn't expect alleles to be randomly distributed for many reasons, for example because:
>
> - SNPs are inherited together through genetic linkage, due to being on the same gene or nearby genes from one another.
> - we would expect combinations of SNPs to reflect the structure of the population, e.g. individuals who are geographically close to one another are more likely to have similar genetics.
> - particular combinations of SNPs may be fatal or otherwise prevent people from passing them on.

The combinations of SNPs are not distributed randomly based only on the frequency of each SNP independently, this is what's known in population genetics as linkage disequilibrium.

4.5.2. Choice of clustering methodology

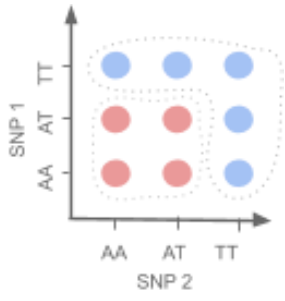


Fig. 4.7 A drawing indicating how the combinations of SNPs we might expect to cause disease would represent a non spherical relationship between SNPs.

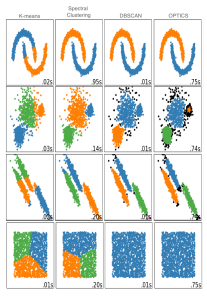


Fig. 4.8 Comparison illustrating differences between the implemented clustering methods. Image adapted from sklearn documentation[196].

The original implementation of the phenotype predictor used k-means clustering[197]. This wasn't suitable for the predictor, since we expect combinations of SNPs to form non-spherical shapes (see Fig. 4.7), and k-means cannot achieve this (see Fig. 4.8).

Spectral clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), IDOS, OPTICS and LOF (Local Outlier Factor) were also implemented. This involved automation of parameter selection, to enable clustering to be performed automatically on thousands of phenotype terms. These methods have theoretical pros and cons with respect to the predictor. For example, OPTICS and DBSCAN do not need the number of clusters as an input, but instead require the minimum number of points required to form a cluster and a radius from each point to consider as part of a cluster, which has more meaning in this context. They also automatically output outliers to clusters, which will affect the resulting phenotype score, potentially in unseen ways - particularly as it is difficult to visualise high-dimensional data. OPTICS is the default setting, as in addition to not requiring a number of clusters, it can identify clusters of differing densities (a quality that DBSCAN lacks - as can be seen in the second row of Fig. 4.8). A final informed choice between these options requires a large benchmarking set.

4.5.2.1. Choice of distance metric

The phenotype predictor's original distance metric was non-linear, such that the homozygous calls were further from each other than the distance via the heterozygous call, as shown in Fig. 4.9. Non-linear distance metrics mean that it is not possible to create a location matrix rather than a distance matrix. This is required for some types of clustering.

A linear distance metric which also captured the increased likelihood of homozygous alleles to be disease-causing (Fig. 4.10) was developed to enable this, and to better represent the biology. In this version, the popularity of an allele decides which homozygous call the heterozygous call is more functionally similar to.

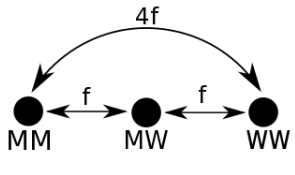


Fig. 4.9 Original non-linear distance metric. MM denotes homozygous mutant alleles, WW denotes homozygous wild type alleles, and MW denotes heterozygous alleles. The FATHMM score for the SNP f, defines the distance between the wild type and mutant alleles.

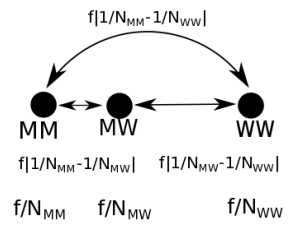


Fig. 4.10 Linear distance metrics. MM denotes homozygous mutant alleles, WW denotes homozygous wild type alleles, and MW denotes heterozygous alleles, and N represents the number of people with that allele call. The FATHMM score for the SNP f, defines the distance between the wild type and mutant alleles.

4.5.3. Necessity of a genetically diverse background set

Rare variants are generally specific to continental groups[198]. Since snowflake is essentially a detector of rare variants (weighted by deleteriousness), it is not hard to imagine how this could lead to many spurious phenotype prediction results if one african was compared to many europeans (or vice versa). Particularly, this is true, because Snowflake calculates it's phenotype score by summing distances between individuals over all phenotype-related SNPs (which could be over 100).

4.6. Testing Snowflake on ALSPAC data

4.6.1. Motivation

In order to test Snowflake, I needed a data set that had a wealth of phenotype and genotype information.

4.6.2. The ALSPAC cohort study

The Avon Longitudinal Study of Parents and Children, ALSPAC[87] is a cohort of over 14,000 families from the Avon area with children born in 1991-1992. It is also known as "the Children of the 90s" study. Many of these families continue to be part of the study to this day, including some of their own children through an extension of the project: children of the children of the 90s (COCO90s).

A wealth of phenotype information (over 80,000 variables) has been collected from these families over the years, through a series of voluntary surveys and clinics, including genotyping of over 9000 children using 23andMe.

ALSPAC's phenotype information, while extensive, is not mapped to phenotype terms in ontologies. All data held by ALSPAC can be searched for in the ALSPAC variable catalogue, after which it can then requested per variable or data type. At the time of writing, the cohort is around 30 years old, meaning that there is little information about phenotypes that manifest later in life, for example Alzheimer's or heart disease. Many phenotype terms may not have any measurements, and there may be many variables associated with some others.

4.6.3. Experiment Design

Due to the identifiable nature of the data, our ethics application did not allow us to access many different phenotypes to perform a cross-phenotype validation of the predictor. Instead, we were granted access to all of the genotype data only first, then allowed to request a small number of phenotypes of interest after running Snowflake.

4.6.3.1. Choosing test phenotypes

I created a shortlist of phenotypes of interest by first restricting the set of scores to phenotypes for which Snowflake makes a prediction within the ALSPAC cohort, then ordering this list by the phenotype confidence score, to ensure that Snowflake could give confident predictions for phenotypes that were requested. I then mapped these to ALSPAC phenotypes by searching the ALSPAC variable catalogue. This resulted in the four : `MP:0001501` *Abnormal Sleep Pattern* (measured using `FJCI250` *Sleep symptom score*), `MP:0001933` *Abnormal litter size* (measured by `mz010a` *Pregnancy size*), `MESH:D001259` *Ataxia* (measured by `kw2030` *Child ever thought to have a problem with clumsiness/coordination*), and `HP:0001249` *Intelligence/intellectual disability* (measured by `f8ws150` *Child had special needs*).

4.6.4. Results

4.6.4.1. Phenotype prediction

Unfortunately, the results of phenotype prediction with Snowflake on ALSPAC data show that snowflake cannot currently be used to accurately predict the following phenotypes in the ALSPAC cohort: `MP:0001501` *Abnormal Sleep Pattern* (measured using `FJCI250` *Sleep symptom score*), `MP:0001933` *Abnormal litter size* (measured by `mz010a` *Pregnancy size*), `MESH:D001259` *Ataxia* (measured by `kw2030` *Child ever thought to have a problem with clumsiness/coordination*), and `HP:0001249` *Intelligence/intellectual disability* (measured by `f8ws150` *Child had special needs*). These were assessed by bootstrapping the F_{max} scores for each phenotype, which gave very low f_max scores (<0.1) and p>0.05 for each phenotype.

Meanwhile other top-scoring phenotypes could not be tested since they did not map well to available ALSPAC phenotypes, e.g. `HP:0007703` *Abnormality of retinal pigmentation* and `HP:0001120` *abnormal corneal size*.

4.6.4.2. Variant function

In addition to testing the test phenotypes, it's possible to look at highest-scoring SNPs (i.e. variants present in people with the highest snowflake "phenotype score") for all phenotypes (even those we did not have phenotype data for). In this way snowflake can be used as a kind of variant prioritisation tool.

These showed some reassuring results:

- **1. Rediscovering known SNPs**: The highest-scoring phenotype in the ALSPAC study was *Abnormal fat cell morphology*. It's highest scoring SNP `rs6659176` is known (via GWAS) to be associated with obesity[199].
- **2. making plausible cross-species variant function predictions:** One SNP, `rs2287780`, was predicted to be responsible for the phenotype of abnormal fetal growth. It is listed in SNPedia[200] as being linked with vitamin B12 and folate production, which are well-known to be important for fetal development. This SNP was flagged up by an ontology term associated with mouse genetics ("abnormal litter size"), meaning it is possible to find candidate human SNP/phenotype associations using information about other species.
- **3. making plausible multi-SNP trait predictions** a combination of 2 SNPs is required for the phenotype (ataxia – a disfunction of the central nervous system) to be predicted. One predicted (`rs13436090`) has been associated with autosomal dominant cerebellar ataxia (ADCA)[201]. The second SNP (`rs2269961`) is a new candidate SNP, a protein from which (Tocopherol-associated protein 2) is thought to be responsible for transporting vitamin E (a deficiency of which can cause spinocerebellar ataxia[202]).

4.6.5. Discussion

Snowlflake's phenotype prediction results were disappointing but could have a number of possible interpretations.

4.6.5.1. Selection of phenotypes

For valid ethical reasons, it was only possible to request a small number of phenotypes from ALSPAC. Snowflake isn't expected to work for all phenotypes, since we know there are many other mechanisms (some not even genetic) behind phenotype besides only the disruption of proteins through missense SNPs in protein-coding genes. The fact that Snowflake was not successful in predicting any of the requested phenotypes could be an indication that none of these phenotypes has this mechanism behind it. Alternatively, it could be an indication that Snowflake is not a successful method for predicting phenotypes even in these cases, and there are many reasons why that could be the case since Snowflake relies on a lot of other pieces of research and software to function.

In selecting phenotypes, I chose phenotypes where (1) Snowflake was expected to make a confident prediction and (2) ALSPAC data could be used to validate this prediction. These two restrictions narrowed down the phenotypes considerably and the majority of the top-predicted phenotype terms (Abnormal Fat Cell Morphology, Abnormal Fetal Development) did not map cleanly to ALSPAC phenotypes.

Since I chose phenotypes primarily by looking at the distribution of scores for Snowflake, our lack of promising results could also be an indication that the phenotype-score (finding interesting distributions of phenotypes) is unsuccessful.

A more interesting test of Snowflake's abilities might be to choose phenotypes with high heritability, and specifically high missing heritability.

I also regret not considering the ethical implications of the choices of phenotypes to predict.

4.6.5.2. Overlap between training and validation data

ALSPAC is a popular cohort study for use in GWAS analyses. As such, many links between genotype loci and phenotypes have been published from this data set, for example loci associated with birthweight[203], asthma and allergies[204], autism spectrum disorders[205], problematic peer relationships[206], and lung function[207]. This data informs GOA genotype-phenotype annotations, and through dcGO this data could already be present in Snowflake, meaning that there may be in some sense, overlap between the training and validation data, which could overestimate the accuracy of the predictor on unseen data[208].

For this reason, I do not think that it is sensible to try any further to use ALSPAC to validate or test Snowflake.

4.7. Discussion

4.7.1. Background

- 1000 genomes has different priorities than us: does not care about rare SNPs - most likely to cause rare diseases
- As diverse a bg set as we can get, but not very diverse.
- Size/diversity of background set constrains how many SNPs we can hae
- Same problem as PQI, the results are very sensitive to our background set (test?)

4.7.2. Difficulty in finding a test set

The `snowflake` project could be considered "blue-sky" curiosity-led research. The motivation for creating `snowflake` was our curiosity in seeing if the resources of Computational Biology could0 be used for the practical outcome of creating phenotype predictions. This was far from incremental, since other leading approaches predicted phenotypes on a phenotype-per-phenotype basis, or restricted the problem to prioritising variants. We can only test `snowflake` on data sets with both genetic and phenotype information across many phenotypes, which means it is very difficult to conclusively test (we have very low statistical power over all phenotypes).

It is disappointing that the phenotype predictor does not produce statistically significant results. However, the phenotype predictor may yet be useful for revealing candidate SNPs for certain kinds of diseases, and when a suitable data set becomes available (e.g. through the growing number of publicly available genotypes on platforms such as OpenSNP[209]), this method will still be ready to be tested. An alternative validation would be experimentally testing a prediction (e.g. with knockouts) of a phenotype with a highly interesting distribution of scores.

–>

4.7.3. Limitations

4.7.3.1. Genotype data

Genotype chips contain only a small fraction of the known disease-causing variants. For example, 23andMe tests for only 3 of thousands of known variants on the BRCA1 and BRCA2 genes implicated in hereditary cancer.

4.7.3.2. Equivalent terms

Despite much development effort, there remain some idiosyncrasies to the predictor. For example, DcGO can map multiple terms to the same set of SNPs. This can sometimes be a diverse group of phenotypes which do not tend to co-occur in individuals and when this occurs, it is likely that we cannot make a good prediction. A semantic similarity measure, such as GOGO[210] or Wang's method[211] could be used to check this, and update the confidence score accordingly. It might be possible that constraining DcGO to use only more closely related species rather than the whole tree of life might be preferable for this task, however, this would be a trade off, as this change would also affect the predictor's coverage of both phenotypes and variants.

4.7.3.3. Coverage of variants: Synonymous SNPs, nonsense and non-coding variants

There are also clearly many aspects of the molecular biology mentioned in chapter 2 that are not represented in the model used by the phenotype predictor. For example nonsense mutations, synonymous SNPs, regulatory networks, and non-coding variants. Updating the predictor to include these things could potentially give the predictor enough power to be validated on existing data sets.

For example, non-coding variants could be included by extending `dcGO` annotations to SNPs in linkage disequilibrium, and using the non-coding version of `FATHMM`, `FATHMM-XF`[212].

4.7.3.4. Localised expression

Another example is that dcGO does not take account of the environment of the cell (e.g. tissue-specific gene expression) in its' predictions. Although domains which are statistically associated with phenotype can be present in a protein, there is no guarantee that the protein will have the opportunity to impact the phenotype (be transcribed).

In investigating some of the ALSPAC phenotype predictions, I identified that some of the predicted dcGO relations between proteins and ontology terms may not be expressed in the tissue of interest. This makes sense, since dcGO makes predictions on the basis of structure, but it's common in molecular biology that cells, proteins or genes have theoretical functionality that is repressed or silenced by another mechanism, for example most human transposable elements or silenced, or in this case, repressors preventing gene expression in some cell types. Filtering out predictions for SNPs in these repressed genes is therefore a potential route to improve Snowflake, and this is the focus of the next part.

4.7.4. Ethics self-assessment

> Your scientists were so preoccupied with whether or not they could, they didn't stop to think if they should.
> – Dr Ian Malcolm, Jurassic Park, Michael Crighton

Like the creation of dinosaurs, the Snowflake methodology itself (rather than a particular use of it) is not the sort of research that usually requires ethical review by Institutional Review Boards (IRBs). This is because most IRBs focus on issues of informed consent, data privacy, and other matters which could cause legal problems for universities, while Snowflake's core methodology uses only publicly available data. As I previously mentioned, there are more general (wider, societal) ethical considerations relating to research in predicting phenotype.

With this in mind, I performed a self-assessment of the worst-case scenario outcomes of this research, in order to understand potential issues and think about what precautions should be put in place to avoid them. These extend out from this research itself, imagining future deployments. To this, I used the Data Hazards framework: a framework that is currently under development, and which I am currently working with the data science research community to develop. Table 4.2 contains the hazards that I felt applied to Snowflake, the reasons why, and what I recommend could be done to prevent these worst-case scenarios.

Data Hazards

The Data Hazards framework is a set of resources to help data science and methodological researchers apply ethics hazards labels to their work. These resources include a set of hazard labels inspired by COSSHH chemical hazard labels with suggested safety precautions, as well as materials to help researchers apply them.

Just as we still use bleach, but do so wearing gloves, data hazard labels are not necessarily statements against doing or using the research, but rather an appeal to "handle with care". Similarly, data hazards aren't meant to be statments of certainty or likelihood of the hazards occuring. They represent worst-case scenarios and apply not only to the specific research project, but also future impacts of further deployments.

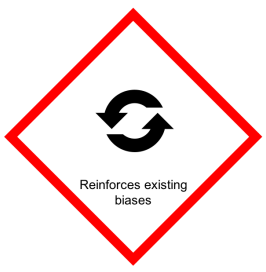
Label name	Label description	Label image	Reason for applying	Relevant safety precautions
Contains Data Science	Data Science is being used in this output, and any negative outcome of using this work are not the responsibility of "the algorithm" or "the software", but the people using it.		Snowflake uses data, makes predictions, and uses unsupervised learning.	When snowflake is deployed in new contexts (e.g. patent licenses sold), it should be done with the understanding that the licensee becomes accountable for using it responsibly.
Reinforces existing biases	Reinforces unfair treatment of individuals and groups. This may be due to for example input data, algorithm or software design choices, or society at large.		Project does not check that the algorithm works just as well for non-white races, and we would expect it to work less well for them since they are less represented in the input data linking variants and diseases[213].	Snowflake's efficiacy should be tested separately for each demographic that any deployment may effect.
Ranks or classifies people	Ranking and classifications of people are hazards in their own right and should be handled with care.		Project does not check that the algorithm works just as well for minority groups, who are less likely to be represented in the input data linking variants and diseases.	• Snowflake's efficiacy should be tested separately for minority groups, before deployment outside research (e.g. healthcare). • Appropriate phenotype terms should be curated before deployment (e.g. removing things like social behaviours, "intelligence" related terms, etc) • When or if to share rankings should be consdered carefully.
Lacks Community Involvement	This applies when technology is being produced without input from the community it is supposed to serve.		The communities of people with the phenotypes have no current involvement in this process.	Relevant communities should be asked about their feelings towards phenotype prediction before deployment in order to curate a list of appropriate phenotype terms.
Danger of misuse	There is a danger of misusing the algorithm, technology, or data collected as part of this work.		The phenotype predictor is not expected to be accurate for all phenotypes, but It could even be used to try to predict phenotypes that are caused by the environment or regions of DNA it does not consider, if these are defined as genetic phenotypes in other literature.	If deployed outside of research, Snowflake should be tested for different types of phenotypes and which ones it does work for should first be understood.
Difficult to understand	There is a danger that the technology is difficult to understand. This could be because of the technology itself is hard to interpret (e.g. neural nets), or it's implementation (i.e. code is hidden and we are not allowed to see exactly what it is doing).		Doesn't use "black-box" machine learning (e.g. deep learning), but has closed source code and a complicated data pipeline.	• If published for research, the code should be Open sourced and the code should be thoroughly documented and tested. • If provided for members of the public, explainers should be created similar to those that 23andMe has.
Privacy hazard	This technology may risk the privacy of individuals whose data is processed by it.		Individual's genetic data is required to run the phenotype predictor. This has many privacy risks, for example identification, use by insurers, being contacted by unknown relatives.	• Ensure there is explicit and well-informed consent from any future participants/users. • Store data securely.

Table 4.2 The seven data hazards which I assessed as applying to Snowflake.

Despite the tongue-in-cheek use of the Jurassic Park quote opening this subsection, I do think that phenotype prediction is something that we should attempt, due to its potential to help people. In "stop[ping] to think" about it, however, I applied 7 of the 11 existing data hazard labels, and set out some specific precautions for using it that I hope will be seriously considered by anyone using the method further. While some of these may seem far-fetched, Snowflake has already been trialled by a genomic analysis company used in clinical decision support.

The question of whether we "could" predict phenotype accurately is also a huge ethical barrier to using it at present. Currently, it's not clear to what extent, or for which types of variants, the phenotype predictor works. The next chapter explains my attempts to validate the predictor using the ALSPAC dataset.

4.7.5. Future work

4.7.5.1. Dependencies, interoperability & simulating data

The phenotype predictor relies heavily on all forms of it's input data: `dcGO`, `FATHMM` and the background cohort. `dcGO` decides which SNPs we consider at all for a phenotype, while `FATHMM` decides to what extent SNPs within that set would be interesting if we see a rare combination. And how rare the combination appears is defined by the background cohort. A limitation of this method is that it's hard to test Snowflake's approach to combining these types of data and clustering independently from these inputs.

I believe a synthetic (simulated) dataset would be important for testing any future iteration of Snowflake.

4.7.5.2. Update input components

All three of Snowflake's input components (`dcGO`, `FATHMM` and the background cohort) have many possible choices - and while it is most important to find good test data, should that be found, finding the best choices of components would be a priority.

For example, Snowflake uses FATHMM-MKL rather than the newer and much more accurate FATHMM-XF. FATHMM-MKL is constrained to build 37 of the human reference genome which is no longer up to date.

4.7.6. Conclusions

While the results of my application of Snowflake to ALSPAC were disappointing, my technical contributions to Snowflake included finding and fixing crucial bugs, which allowed it to go on to it's latest and most successful iteration as Nomaly[165].

Snowflake (and Nomaly) represent a highly novel approach to phenotype and variant function prediction, and it is possible that it's limitations can be overcome as new datasets become available.

5. Filtering computational predictions with tissue-specific expression information

This chapter presents a more focused approach to improving phenotype and protein function predictions. I present a prototype filter for protein function prediction methods (Filip) which I developed for the CAFA3 competition, which filters out predictions where the gene is not expressed in the tissue relating to the phenotype. This approach was prompted by the discovery that this is one of the sources of noise in Snowflake, as described earlier.

> 💡 **Contributions in this chapter**
>
> - The Filip method for filtering protein function prediction based on tissue-specific gene expression.
> - Entering predictors using Filip in the CAFA3 competition, which contributes to the CAFA3 paper[5],

5.1. Introduction

As we explored in chapter 2, there is a complex web of interactions between proteins and other molecular machinery that lead to phenotype. Our current understanding of how phenotype arises from genotype tells us that knowing which proteins *can* be produced isn't necessarily enough of a clue to tell us about phenotype. Liver cells and heart cells have the same DNA, but it's how that DNA is used (what genes are expressed in the cell) that leads to the difference between those cell types. Since larger scale phenotypes will follow from cellular differences, we expect gene expression data to be a useful measure for phenotype prediction.

This is backed-up by data: disease-associated genes are generally over-expressed in the tissue they cause symptoms in, with the exception of cancer-associated genes[214,215]. This can and has already been leveraged effectively as part of some gene and variant prioritisation methods [216,217].

5.1.1. Motivation: improving phenotype and protein function prediction

The performance of the snowflake predictor led me to focus my efforts on finding an answer to a much smaller piece of the genotype-to-phenotype puzzle. As mentioned in the previous chapter's discussion, some predictions of a protein's phenotype are incorrect because the protein is not produced, even though they do have a structure that means that they could be involved in the pathway if they were present. To understand if this is the case, we need to know as a minimum if a gene is *ever* expressed a relevant context. This would rule out, for example, proteins that are predicted to be associated with eye health, but are only ever produced in developing limbs.

Machine learning methods are currently the most successful class of protein function predictors. While this is promising for answering one aspect of the problem ("what are the functions of a given protein?"), they do not always attempt to answer how or why. Structural or sequence methods that estimate protein function based on, for example conservation or structure, counter this problem, but they are currently less accurate: one of the reasons for this might be the lack of inclusion of cell context. Few of these methods include tissue-specific gene expression information (such data was completely missing in the first and second CAFA competitions). Filtering out predictions where genes are never expressed in a relevant tissue may help in protein function prediction, just as in phenotype prediction.

5.1.2. When are transcripts "expressed"?

The idea behind Filip is that some proteins are predicted to affect phenotypes that they are unable to affect, because the environment in the tissue or cell means that the protein isn't around to perform it's function (or fail to). And we have a measure of gene expression for which many proteins have `0` counts (and therefore `0` TPM) in many tissues, so we *could* apply the filter to this cut-off. But is it meaningful to do so?

Like all chemical reactions, transcription is a stochastic process; there is an element of randomness; to describe if a transcription event will happen at a specific moment you have to use statistics. Genetically identical organisms with the same environment have different measured gene expression patterns[218] and the same can be said for single cells from the same organism[219]. The reason that it's hard to predict with precision whether a given protein will be transcribed at a given moment is that it depends on the concentration of different molecules in the cell and the energy of the system. Transcription events which have a very low probability of occurring will happen sometimes and we will measure this. If we sequenced the transcriptome in infinite depth, we might expect all transcripts to be expressed at some level.

When we look at expression data for a sample, it will just be a snapshot of the transcription in that sample, and one that isn't necessarily representative of what's happening all the time. Very low count values in a sample are extremely common, and these are usually considered to be difficult to distinguish from transcriptional noise: low levels of transcription with little effect are often randomly happening in the cell. In addition to the biological stochasticity (which could possibly create phenotypic differences), RNA-Seq is sensitive to technical experimental artefacts (batch effects) due to differences in RNA extraction and library preparation[223]. In both cases, it is low counts where this is most difficult to correct for. So it isn't necessarily meaningful to take all genes expressed above `0` TPM as a sensible cut-off for whether a gene counts as "expressed" or not in a tissue: when I dichotomise proteins as "expressed" or "not expressed", I am using this as a convenient shorthand for "meaningfully expressed" or "not meaningfully expressed".

> **Transcriptional noise**
>
> Transcriptional noise is variation in rates of transcription due to the implicit stochasticity of the reaction process. The implication is that many transcripts with low counts do not play a big role and cells are known to have mechanisms to protect themselves from this noise[218,220]. Since it is difficult to distinguish between meaningful and non-meaningful and expression, in differential expression analyses it is common to remove low count transcripts[221,222]. Similar noise occurs in the process of translation (translational noise).

We also know that proteins that do cause phenotypes are likely to be highly expressed in tissues related to the phenotype. This means that we definitely want to keep protein-phenotype predictions where proteins are produced at high levels in the tissue of interest. The question becomes: when do TPM levels become low enough that we would want to exclude them?

5.2. Algorithm

In order to overcome the problem of predictors containing erroneous predictions due to a lack of gene expression information, I have created a lightweight tool which allows researchers to filter their phenotype or protein function predictions using tissue-specific gene expression information.

Drawing on the noble tradition of scientists naming things badly, I call this Filip as it is for **Fil**ter**i**ng **p**redictions.

> **Naming things**
>
> > There are only two hard things in Computer Science: cache invalidation and naming things
> >
> > – Phil Karlton
>
> Aside from the above joke, there is also evidence in the literature[224] to suggest that strained acronyms exist across scientific disciplines.

5.2.1. Overview

Fig. 5.1 illustrates Filip's two-step approach, which aims to filter out predictions for proteins which are not created in the tissue of interest (related to the predicted phenotype). The filter is a simple rule-based tool, which is designed to be used on top of any protein function predictor, but would provide the most value for predictors that rely on structural or sequence similarity.

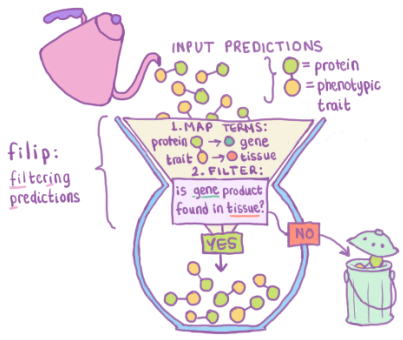


Fig. 5.1 An illustration showing how Filip works. It's a two-step process where protein-phenotype predictions are expected as input. In step 1,*preprocessing*, proteins are mapped to genes, and phenotypes are mapped to tissues. In step 2, *filtering*, Filip filters out any predictions where for which the gene is not expressed in the tissue.

5.2.2. Inputs

Three types of input are needed for Filip:

1. Protein function predictions
2. Normalised gene expression data.
3. A map from gene expression samples to Uberon tissues.

5.2.2.1. Protein function predictions

Protein function predictions must be links between Protein identifiers and phenotype terms from GOBP, HP, MP or DOID ontologies. This is the standard for CAFA competitions.

5.2.2.2. Gene expression file

If Filip was a filter coffee machine, the gene expression (GE) file would be the (reusable) filter: it is the part that determines what can and cannot pass through the filter and it can be used with any kind of input predictions (coffee). Once we have the GE file, it can be reused for any different protein function predictor, as long as it predicts phenotype terms related to the samples in our GE file.

The user must also determine a cut-off: the minimum gene expression level to count as "expressed". The higher the cut-off the more genes will count as unexpressed, and therefore more predictions will be filtered from the original list.

5.2.2.3. Sample-tissue map

Some GE datasets will include a sample-to-Uberon map as part of their metadata (e.g. FANTOM5). For those that don't, the ontolopy Python package can be used to map between samples tissue names and their Uberon tissue.

5.2.3. Step 1: Preprocessing

The preprocessing file outputs:

1. A phenotype-to-sample map, which stores a list of column indices in the gene expression file which Filip should use for filtering each phenotype.
2. A protein-to-gene map, which maps between proteins present in the input predictions and genes present in the input GE file.

Mapping between phenotype and sample is the most invovled part of Filip: it relies on Ontolopy (next Chapter) to create this mapping.

5.2.4. Step 2: Filtering

The filtering step takes the orginal inputs, preprocessing outputs, and a GE cutoff as input. It outputs a reduced list of predictions that are still valid (are expressed above the cut-off on average across the samples).

5.3. Data

This section describes the gene expression data used for creating and validating Filip, including it's provenance as well as any necessary data-cleaning.

I also describe the benchmarking data I used to develop Filip.

In addition to this, Filip requires the input of a protein function or phenotype prediction method, but this (and the data required for this) is described in Validation method.

5.3.1. Expression data: FANTOM5

The Filip method requires expression data to inform whether or not predictions should be filtered out. The FANTOM5 data set was chosen for this purpose.

FANTOM5 represents one of the most comprehensive collections of expression data in terms of tissue and cell type. It consists of expression data, captured using the CAGE technique. FANTOM5 collected a combination of human, mouse, health, and disease data, as well as time courses and cell perturbations. At the time of developing it was the latest data output of the FANTOM consortium.

The FANTOM Consortium

The Functional ANnoTation Of the MAmmalian genome (FANTOM) consortium was established as the human genome project was nearing completion when researchers had a parts list of human biology, but few of the functions of these parts (genes) were known. The consortium has run a range of large scale collaborative projects in five rounds to further this goal. The first FANTOM project used only the mouse genome, but later versions also included human.

My reasoning for choosing FANTOM5 data as the input gene expression data to test Filip was:

- The data set has a good coverage of different tissue types, which I hoped would be helpful in Filip having a good coverage of predictions.
- The data set has an ontology of samples, which is already linked to Uberon tissue terms, making the mapping process much easier.
- For the purpose of Filip (getting measure of whether a cell is meaningfully expressed in a tissue of interest), choosing bulk RNA-Seq over scRNA-Seq makes sense, as it is a measure of many more cells.

I chose the version of the FANTOM5 data that:

- had been reprocessed using the hg38 reference genome (the original FANTOM5 data was processed using hg19)[225].
- contained annotated information about the samples, as this information could be used to aid in mapping.
- available in TPM format.

5.3.1.1. Data files and acquisition

I downloaded the following files from the FANTOM website:

- the FANTOM5 CAGE peaks expression data containing expression in counts per CAGE peak, and mappings to transcript id (ensembl ENST id), HGNC id and entrez gene ID. The long sample labels in this file are also a source of metadata (including sample identifiers (FANTOM accession numbers)).
- FANTOM's human sample information file containing text descriptions about sample, for example FANTOM accession numbers, tissue, age, sex, disease, etc, which is necessary for data cleaning.
- the FANTOM5 ontology containing an obo file mapping between FANTOM accession numbers, Uberon and cell ontology (CL) terms.

FANTOM5 Accession numbers

Each FANTOM *sample* has an accession number of the form `FF:#####-#####`. These numbers are used in all three of the FANTOM5 data files. Note: some samples have repeat measurements per sample.

5.3.1.2. Initial FANTOM5 data cleaning: sample info file

5.3.1.2.1. Sample categories

Restriction to primary cell and tissue samples:

The human FANTOM5 sample information file contains four categories of samples (in the `Characteristics [Category]` field):

- **time courses**: RNA extracted from samples being measured over time as cells change types during cell development and differentiation (`783` samples), e.g. `'FF:12265-130A6'` - ***Lymphatic Endothelial cells response to VEGFC, 01hr20min, biol_rep1 (MM XIX - 6)***.
- **primary cells**: RNA extracted from cultures of cells recently isolated from tissues, before undergoing proliferation with nutrients specific to the cell type (`561` samples), e.g. `'FF:11216-116B1'` - ***Urothelial cells, donor0***.
- **cell lines**: RNA extracted from immortal cell lines (which unlike primary cells can keep undergoing division indefinitely) (**268** samples).
- **tissues**: RNA extracted from post-mortem tissues, which may be pooled or individual donors (`183` samples), e.g. `'FF:10012-101C3'` - ***brain, adult, pool1***.
- **fractionations**: RNA extracted from parts of cells (fractionations) (`21` samples), e.g. `'FF:14310-155C8'` - `'Fibroblast - Aortic Adventitial donor3 (cytoplasmic fraction)'`.

HeLa cell line

The FANTOM5 experiment contains HeLa cell lines samples (e.g. sample `FF:10815-111B5`).

HeLa is short for Henrietta Lacks, the woman whose cells were the source of this first immortal cell line. Henrietta was a black woman who lived in Baltimore, Maryland. Her cells were taken without consent during a hospital biopsy for an aggressive cervical cancer, which she died from at age 31 in 1951.

Companies continue to profit from the sale of these lines of cells, since such cell lines have several practical advantages over primary cells, notably their immortality, low variability (compared to primary cells, which vary depending on cell donor characteristics such as age and sex), and ease of keeping alive (without the need for e.g. specific nutrients).

Some companies have recently begun to pay reparations for this injustice[226].

I restricted the data set to only *tissues* and *primary cells*, as these represent the closest approximations to *in vivo* biology. Immortal cell lines are often expressed differently than their primary counterparts[227,228], and time courses and fractionations do not represent any particular tissue.

Sample Type:

As mentioned, tissues can come from a pool, or individual donor. This information can be found in the `Charateristics [description]` field. I combined this information with information from the `Characteristics [Category]` field to create an additional `Sample Type` field that describes whether a sample is a `tissue - pool`, `tissue - donor` or `primary cells` sample.

Technical and biological replicates:

The FANTOM accession numbers are per sample, not per measurement. Samples for which there are repeat measurements (technical replicates) will show up multiple times in the expression file. FANTOM technical and biological replicates are indicated in long labels of the annotated expression FANTOM file, by the inclusion of "tech_rep" or "biol_rep" in the long sample labels e.g. `tpm.Dendritic%20Cells%20-%20monocyte%20immature%20derived%2c%20donor1%2c%20tech_rep1.CNhs10855.11227-116C3.hg38.nobarcode`. These were used to create additional fields for the human samples table.

Technical and biological replicates

Usually *technical replicates* refer to repeated measures of the same sample, while *biological replicates* refer to separate samples which have been treated in the same way (e.g. different donors)[229].

In FANTOM, the "biol_rep" and "donor" label are both used to denote biological replicates.

Note: there is an error in the original transcript expression file for one of these identifiers (`tpm.Dendritic%20Cells%20-%20monocyte%20immature%20derived%2c%20donor1%2c%20rep2.CNhs11062.11227-116C3.hg38.nobarcode`) such that it is missing the "tech" part of the the replicate label. There is a hard-coded fix for this accession when I read in the input file and the FANTOM data curation team was informed.

After restricting the data set to *primary cell* and *tissue* type samples, there are `58` remaining samples which have biological replicates (between 2 and 3 replicates each), and `8` sets of samples with technological replicates (2 replicates each).

Age and age range:

The age of the sample source donor(s) is available through two fields in the human sample information file: `Characteristics [Developmental stage]`, and `Characteristics [Age]`. These fields contain description-like text, which are somewhat inconsistent, for example, "3 year old child", "3 years old child", "25 year old", "76" and "76 years old adult" all feature in the same column, amongst other errors. These were standardised into a new field (`Age (years)`). This field does not seek to include multiple ages (i.e. when the sample comes from a pool of donors). There is a complementary (i.e. no overlap) field (`Age range (years)`), which contains age ranges for the `46` samples that contain multiple ages. In both columns, some samples contain fetal samples, in which case, I convert age (range) to a negative decimal (converted to years before birth).

There were also some discrepancies between ages and developmental stages in the FANTOM human samples file. For example, sample `FF:10027-101D9` is labelled as *thymus, adult, pool1* in the *Description* field, but as *0.5,0.5,0.83 years old infant* in the *Developmental Stage* field. Sample `FF:10209-103G2` had an age of 'M' and a sex of '28'. I reported both these discrepancies: and the latter has since been fixed in the FANTOM file, and for the former, I hardcode the age to `NaN`.

Sex:

The `Characteristics [Sex]` field contains information about the sex of the sample source donor(s). Similarly to age, due to the consortium nature of FANTOM5, the entries of this field are not consistently labelled. They undergo data cleaning into 4 categories: male, female, mixed (pool with both male and female samples), and unlabelled.

Disease and tissue mapping:

The disease status of samples (e.g. healthy/non-healthy) is not straight-forwardly labelled in the human sample file, so requires some basic text-mining (and cross-referencing with ontology terms). Similarly, there is a `Characteristics[Tissue]` field in the human samples file containing some manually mapped tissue types, but as I point out with an example in the exploratory data analysis, these do not contain ideal mappings for Filip.

The continued data processing of these components is described in the methdology section, after the introduction of `uberon-py` (the package developed to do this).

5.3.1.3. Initial FANTOM5 data cleaning: expression file

The tidied and restricted sample data, is combined with the FANTOM5 CAGE peaks expression data file and processed to create a protein-centric expression file. The CAGE peaks have already been cleaned by FANTOM (labeled as "fair") meaning that CAGE peaks do not overlap.

CAGE peaks with associated proteins:

The CAGE peaks represent all kinds of mRNA transcripts, not only those which map to protein-coding gene, for example "RNA genes" representing pseduogenes or long non-coding RNAs. The FANTOM file provides mappings to Uniprot IDs (`uniprot_id`), and these are used to discard the CAGE peaks that do not map to protein-coding genes: this takes us from `209912` to `58592` rows (CAGE peaks).

CAGE peaks mapped to one gene only:

CAGE peaks are mapped to genes based on overlap with the gene, so it is not always clear which gene a CAGE peak maps to. For simplicity, and to remove the potential of wrongly mapped genes being used in Filip, protein-coding CAGE peaks (those which are mapped to at least one `uniprot_id` by FANTOM) but that map to multiple genes are removed. These can be found by looking at either the `hgnc_id` or `entrezgene_id` gene identifier columns. The choice of gene ID matters, since there are discrepancies between gene ID databases: in this case, choosing `hgnc_id` finds all those CAGE peaks found by using `entrezgene_id`, and more, so these are removed. This represents a total of `579` CAGE peaks that map to multiple genes according to the given identifiers.

Proteins that map to multiple genes:

For Filip, the expression was calculated per protein (since it is protein function predictions that it is filtering), rather than per CAGE peak (summing the TPMs of all CAGE peaks mapped to a protein to get the total for that protein) as in the original data, or as is often presented per gene. This gave `56554` rows of "protein expression" data.

Of these, there were then `59` rows of data (corresponding to `21` proteins) for which each protein maps to multiple genes. This happens when different genes are translated to make identical protein products, for example the H4 human histone protein is encoded by 14 different genes at different loci, across three different chromosomes. It used to be the case that Uniprot would map these genes to the same Uniprot ID, but more recently different Uniprot IDs are used to capture where the proteins came from. These small number of rows were also removed for simplicity.

5.3.1.4. Exploratory Data Analysis

Samples:

After restricting the samples to those which are primary cells or tissues, there were `744` remaining samples.

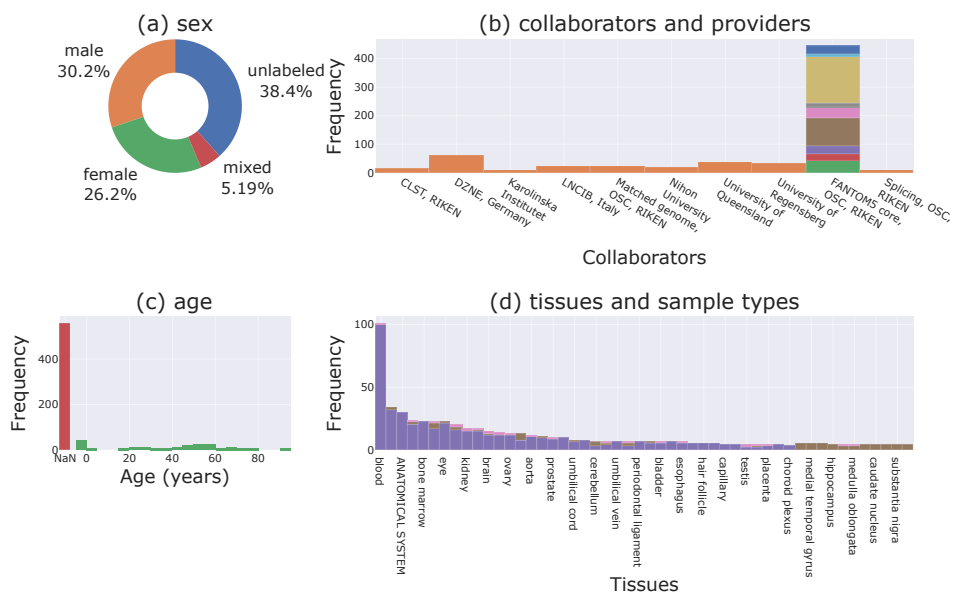


Fig. 5.2 (a) sex: a donut plot showing the sex labels of samples. (b) collaborators and providers: a stacked histogram showing the `10` most common collaborators, and `10` most common providers. (c) age: a histogram of age of sample donors (this does not include the `46` samples which have age ranges due to pooled donors of various ages). (d) tissues and sample types: a histogram showing the `50` most common tissues, spread across the different types of samples (primary cells, tissue donors, and tissue pools).

Sample metadata: Looking at the FANTOM5 data (see Fig. 5.2), overall we see that the samples are very varied, across ages, sex, sample providers, and collaborators, although (d) shows that the majority of samples are *primary cell* samples, and very few are *tissue - pool* samples. Secondly, we can see that after careful cleaning, some metadata is missing, i.e. 38.4% of samples have unknown sex (a), most collaborators did not label the sample provider (b), and most samples do not have a labelled age (c).

Sample Tissues: In Fig. 5.2 subplot (d), we can also note some interesting things about the tissue types provided by the Fantom Human Samples file. `30` primary cell samples are labeled *ANATOMICAL SYSTEM*. If we look closer at these samples, we can see that it is theoretically possible to map some of these samples to tissues (see Fig. 5.3).

There is also the question of how general or specific the human sample categories are. There are `101` samples which are mapped to *blood* (Fig. 5.3 (d)), but when we come to map the FANTOM5 tissues to phenotypes, this may be too broad a category. Similarly, there are `47` with less than three samples each (not pictured) that may be too narrow to map to phenotypes, and a more accurate picture of that phenotype would come from taking a more general tissue.

	Charateristics [description]	Characteristics[Tissue]	Characteristics [Category]
Charateristics [ff_ontology]			
FF:11923-125H6	Fibroblast - Gingival, donor7 (aggressive peri...	ANATOMICAL SYSTEM	primary cells
FF:11933-125I7	Olfactory epithelial cells, donor1	ANATOMICAL SYSTEM	primary cells
FF:11938-126A3	gamma delta positive T cells, donor2	ANATOMICAL SYSTEM	primary cells
FF:11960-126C7	Smooth muscle cells - airway, asthmatic, donor1	ANATOMICAL SYSTEM	primary cells

Fig. 5.3 An example of four *ANANTOMICAL SYSTEM* tissues, with tissue-specific cells, indicating that they could be mapped to tissues. For example sample `FF:11922-125H5` is a gingival fibroblast, which are one of the main constituent cells of gum tissue.

We can also see in Fig. 5.3 that this data set, though having undergone some data cleaning, still contains disease samples (e.g. "aggressive periodontitis").

Protein-centric TPM:

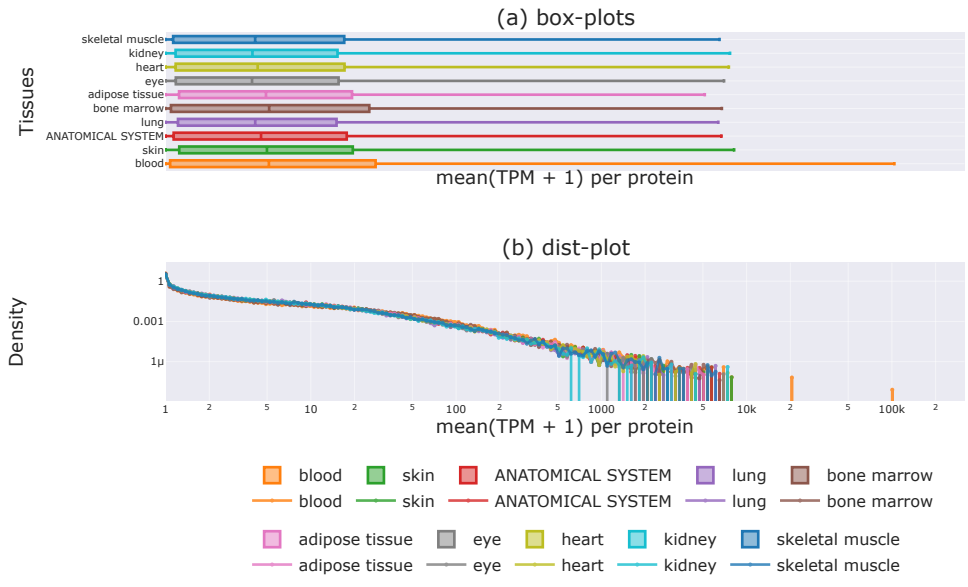


Fig. 5.4 (a) box-plots showing the distribution of mean (TPM+1) values (note: logarithmic x axis) for the top `10` most common tissue and primary cell samples in the FANTOM5 human data. (b) density-plots showing the distribution of mean (TPM+1) values for the top `10` most common tissue and primary cell samples in the FANTOM5 human data on log-log axes.

As expected, Fig. 5.4 shows similar distributions of expression per tissue since the data is TPM normalised (since TPM normalises samples by sample library size to account for sequencing depth), with the characteristic long tail.

5.3.2. Cell, tissue, and phenotype mapping data

I also used the following datasets to aid in mapping to a common set of identifiers:

- the uberon extended ontology OBO file from the uberon website to assist in mapping cells and tissues.

5.3.3. "Training" set: CAFA2

During development, I tested Filip by comparing DcGO only and Filip-plus-DcGO on data from the 2nd round of the CAFA competition: CAFA2. I chose to use the CAFA2 data because rather than a larger set of annotations (such as those available from SwissProt-KB or GOA) because it provided a way of validating on unknown targets. I.e. if I made predictions with DcgO using the version of GO from the time the challenge was launched, and I use the groundtruth data provided by CAFA2, then I could compare my results with those in the CAFA2 competition and I could look at my results on unknown targets. Although Filip was not literally "trained" on this data in a machine-learning sense (it doesn't have any formalised parameters), I had access to the "groundtruth" data as I was developing CAFA2.

This was the most recent round of CAFA for which there were "groundtruth" data available at the time of development.

5.3.3.1. Data files and acquisition

The data consisted of:

- CAFA2 targets: a list of proteins which the CAFA2 competition was soliciting predictions for.
- CAFA2 ground truth data: experimentally validated associations between proteins and GO terms, divided by category.

Both of which could be found within the CAFA2 paper[121]'s Supplementary Material.

CAFA2 targets: CAFA2 provided targets from species across the tree of life: bacteria (10 species), archaea (7 species), and eukaryotes (10 species). Since tissue-specific gene expression data (which Filip requires) is not available for all species, I only used the human targets (in `data/CAFA2-targets/eukarya/sp_species.9606.tfa`).

FASTA format

The `sp_species.9606.tfa` file is a FASTA file containing information about `20257` proteins, each with a CAFA2 identifier (e.g. `T96060000001`), Uniprot Entry Name (the mnemonic identifier for the protein, e.g. **`1433B_HUMAN`**), and the amino acid sequence, as in the following excerpt:

> FASTA is a text-based file format for proteins, where each letter represents an amino acid (except X, which represents any amino acid).

```
>T96060000001 1433B_HUMAN
MTMDKSELVQKAKLAEQAERYDDMAAAMKAVTEQGHELSNEERNLLSVAYKNVVGARRSS
WRVISSIEQKTERNEKKQQMGKEYREKIEAELQDICNDVLELLDKYLIPNATQPESKVFY
LKMKGDYFRYLSEVASGDNKQTTVSNSQQAYQEAFEISKKEMQPTHPIRLGLALNFSVFY
YEILNSPEKACSLAKTAFDEAIAELDTLNEESYKDSTLIMQLLRDNLTLWTSENQGDEGD
AGEGEN
```

CAFA2 benchmark: The CAGA2 benchmark data was available in the `/data/benchmark` directory of the CAFA2 Supplementary Data. It includes:

- **Lists** of different types of targets for which there is groundtruth data (in `/data/benchmark/lists`): each line of these files is a CAFA2 protein identifier (e.g. `T96060015767`). The lists are separated into different files according to species, source phenotype ontology (e.g. `HP`, `GO`), and protein type (type1 = No Knowledge, type2 = Limited Knowledge). There are `7` files for human.
- **Groundtruth** associations (in `/data/benchmark/groundtruth`): tab-separated CAFA protein identifiers and phenotype ontology terms, e.g. `T96060000002 HP:0000348`), organised into `8` separate files by source phenotype ontology, and whether the proteins are experimentally annotated to the exact term, or whether an association can be inferred due to a ontology relationship.

5.4. Validation method

In order to fairly test Filip, I entered it in the third CAFA competition, where it could be independently assessed by other researchers. In CAFA, each researcher can enter up to three methods, so I tested Filip by entering DcGO alone, and DcGO plus Filip, so that I could compare their performances.

5.4.1. Test set: CAFA3

After initial development, I entered DcGO only, and Filip-plus-DcGO into the CAFA3 competition in order to test Filip on an unseen dataset.

This meant that I did not download the CAFA3 ground-truth, as this analysis was done by the CAFA3 team, but only the CAFA3 targets, these continue to be available through the CAFA website.

Again, I used only the human targets (file `target.9606.fasta`). This is again a FASTA file, with the same format as for CAFA2, this time containing `20197` targets proteins.

5.4.2. Filip inputs for validation

As previously described, three types of input are needed for Filip:

1. Protein function predictions
2. Normalised gene expression data.
3. A map from gene expression samples to Uberon tissues.

I described the gene expression data and metadata for (2) and (3) used for validation in the previous section.

5.4.2.1. Creating protein function predictions (DcGO)

I used DcGO as a test since I knew that it's structure-centric prediction method didn't include any gene expression information.

To create the input to DcGO, I used:

- BioPython[230]'s `Bio.SeqIO` interface for reading CAFA FASTA files.
- SUPERFAMILY[118] domain assignments for Homo Sapiens.
- UniprotKB[231]'s mapping tool to create a mapping between the UniprotKB id's provided by CAFA and the ENSP ID's used by SUPERFAMILY.

The script to create the UniprotKB IDs is **`available here`**, to create the input for DcGO is **`here`**. Then, to create the DcGO-only entry, I used the `DcGOR` library[232] (the `dcAlgoPredictMain` function).

The DcGO predictions contain only **15192949** of **20257** proteins and **15749** phenotype terms (all of which are `GO` terms).

5.4.3. Running Filip

I used an early version of ontology to map between uberon tissues and phenotypes. I describe this process in detail in Section 6.5: for CAFA3, I used phenotypes present in DcGO predictions as targets, and looked for mappings only including Uberon terms (not Cell Ontology terms).

The cut-off was chosen by plotting the distribution of TPM expression and choosing a value below which there appeared to be little noise (50 TPM) between biological and technical replicates.

5.4.4. Validation Methodology

This confidence score allows for a range of possible sets of predictions, depending on the threshold parameter τ. Precision (the proportion of selected items that are relevant), and recall (the proportion of relevant items that are selected) are defined in terms of true positives t_p, false positives f_p, and false negatives f_n:

$$precision = p = \frac{t_p}{t_p + f_p}$$

$$recall = r = \frac{t_p}{t_p + f_n}$$

Precision-recall curves are generally used to validate a predictors performance, but the F_1 measure combines these into a single measure of performance:

$$F_1 = 2\frac{precision \cdot recall}{precision + recall}$$

Since the precision and recall will be different for any τ, the F_{max} score is the maximum possible F_1 for any value of τ.

CAFA validation can either be term-centric or protein-centric. For each option, submissions are assessed per species and for wholly unknown and partially known genes separately.

5.4.4.1. Limitations of validation method

There is no penalty for making a broad guess, or reward for making a precise one. This is one of the reasons that the naive method does so well: for example it is not penalised for guessing that the root term of the GO BPO ontology Biological Process is related to every gene.

Due to the nature of the validation set, it's possible that the best-scoring CAFA methods simply predict which associations are likely to be discovered soon (i.e. associations to genes people are currently studying, which is well-predicted by genes that have recently been studied).

5.5. Filip results

5.5.1. CAFA 2

5.5.1.1. F_{max} Improvement

F_{max} DcGO	F_{max} DcGO + Filip
0.408	0.409

Table 5.1 CAFA2 data f-max results for DcGO and filip

During development, I validated Filip using the original DcGO CAFA2 submission, using the CAFA2 targets. The F_{max} score was calculated for human BPO, combining both *No Knowledge* and *Limited Knowledge* targets. Table 5.1 shows that Filip provides a small benefit to the F_{max} score.

5.5.1.2. Bootstrapping

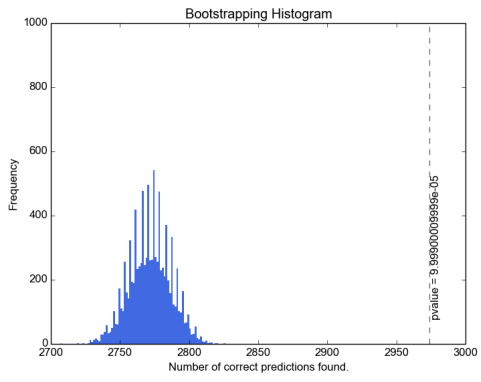


Fig. 5.5 This bootstrapping histogram shows the distribution of the number of correct predictions found when deleting a random selection of DcGO predictions (the same number as are discarded by the filter). The dotted lines shows the number of correct predictions found by the filter. The low p-value (9.99×10^{-5}) shows the low probability of the filter performing at least as well as it has (in terms of the number of correct predictions) by random chance.

The small improvement is due to Filip filtering out 85,637 GOBP human protein predictions, only 23 of which were true according to the CAFA2 ground truth, meaning that 99.973% of filip's predictions (on what to filter in or out) were correct.

To ensure that this is a better success rate than we would expect by chance, I performed a bootstrapping test by taking out random sets of 85,637 predictions from the DcGO set and measuring the number of true positives remaining in the set. This was repeated 100,000 times to create Fig. 5.5, and calculate the p-value $p < 0.001$, meaning that the filter performed far better than chance.

5.5.1.3. Relationship between incorrect terms

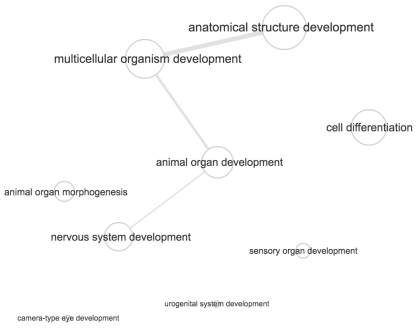


Fig. 5.6 A grouping/summary of the incorrectly excluded predictions. Larger circles represent terms which are parents to more of the input terms. Lines represent the relationships between pictured terms.

I also looked to see whether there was any relationship between the 23 incorrectly removed predictions. Interestingly, all of these incorrectly filtered out predictions were for GOBP terms which were related to development (e.g. tissue development, anatomical structure development, epithelium development, organ morphogenesis). Fig. 5.6 shows relationships between the incorrectly mapped GO terms, created using ReviGO[233]. The fact that the incorrectly filtered out terms are all related to development may be due to a lack of tissue-specific developing tissues in the FANTOM5 data-set used by Filip.

5.5.2. CAFA 3

The same kind of improvement is seen by the independently calculated CAFA3 results (validated by the CAFA3 team). I entered two models into CAFA3: DcGO only and DcGO plus Filip, for Human and for Gene Ontology Biological Process terms only.

In all categories, Filip improved DcGO by 0.02 F_{max} (see Table 5.2). This was not enough to be a competitive model (ranked between 33 and 38 out of 67 for this category). Despite this, this result does show that the improvement was reproduced in another data set, carried out by other researchers.

Type	Mode	F_{max} DcGO	F_{max} DcGO + Filip
No Knowledge	Partial Assessment	0.326	0.328
No Knowledge	Full Assessment	0.326	0.328
Limited Knowledge	Partial Assessment	0.503	0.505
Limited Knowledge	Full Assessment	0.503	0.505

Table 5.2 CAFA3 f-max results for DcGO and filip

5.6. Discussion and Future work

However incrementally, Filip met its goal of increasing the precision of DcGO predictions: Filip was correct in 99.973% of it's predictions, showing that this could be a useful approach if coverage can be increased. In addition, 100% of wrongly filtered out (true) predictions, appear to be explainable due to the sample condition of the gene expression data (development-related phenotypes, with a lack of development age tissue-specific samples).

The results show that including gene expression information does improve results of a structure-based predictor, and that this improvement is extremely unlikely to be due to chance. The overall improvement is very slight, but this could be be improved by a more comprehensive coverage of gene expression data for tissues, and/or by an improved mapping of tissues to ontology terms.

5.6.1. Coverage

We have seen that Filip was successful for 99.973% of it's "choices", but that the number of decisions it could make were not enough to usefully boost the performance of the predictor it was tested on. This reveals that the limited success of Filip on the CAFA data is due to it's poor mapping coverage.

Although the FANTOM5 data set has a excellent coverage of tissue types and number of samples, the filter is nonetheless limited to the tissues it contains. This low coverage of tissues limits the number of predictions that Filip can filter out, and as we saw in the results, this is the major bottleneck for it's performance. However, by combining baseline gene expression sets from multiple sources, the coverage of tissues and therefore phenotype terms might be improved.

In addition, although proteins, tissues and protein functions may be present in multiple species, Filip only currently measures if a gene is expressed in the human tissue of interest. This further reduces the coverage of CAFA predictions that Filip could be tested on.

In addition to the influence of the input gene expression data set, the poor coverage is also limited by the quality of the mapping, which is reliant on the input metadata and ontologies. In this initial test of Filip, I did not include mappings via Cell Ontology terms. Including Cell Ontology terms can increase the coverage (see Table 6.1).

5.6.1.1. Practical difficulties in finding and creating alternative input data

In theory, Filip could be used with any other RNA-Seq data set with a wide range of tissues. In practice, however, finding a data set with the appropriate spread of tissues and cell samples (and furthermore, detailed metadata about these samples) is difficult.

The Ontolopy package (next chapter) does make mapping samples to Uberon tissues possible, even when only names or descriptions of the tissues are present.

5.6.2. Wrongly filtered out tissues

100% of wrongly filtered out tissues were "development" terms. This could mean that time is another way in which cell context should be considered, for example we shouldn't filter out predictions for development phenotypes if we only have adult/not-fetal tissues, and perhaps vice versa. Should we include developing tissue samples (e.g. from a fetus) as evidence that a gene is expressed in a tissue type related to an adult phenotype that manifests after development? This is another question for which it would be necessary to increase the coverage of the GE data set(s) used by Filip to answer.

In the future, when new datasets allow the coverage of Filip to be increased, I would expect there to be other aspects of Filip that could improve predictions. For example, currently all protein isoforms contribute to the gene expression cut-off for a gene, but this may not always be sensible and is something to keep in mind for the future.

5.6.3. Future work

There is some additional work that needs to be done with Filip to get it to the stage of being ready for publication. This primarily includes testing on more protein function prediction methods, and software engineering work necessary to release it as a resource for others to use in future CAFA competitions or similar (e.g as a public Python package/command line tool). The mappings between phenotype terms and tissues should also be made available at this time, so that other people can easily interrogate these for individual genes.

I plan to do this work in time for the next CAFA challenge, by which time I would like to improve the coverage using the mapping improvements made in the rest of this thesis, specifically those made by including the cell ontology (in addition to Uberon) - explained in the next chapter, and using the combined tissue-specific gene expression data set. I would also like to enter this competition using the naive predictor plus Filip, to test Filip's potential as a standalone predictor.

5.6.3.1. Speed

For usability, it would speed up the process of running Filip on new predictors considerably to pre-calculate the tissue-phenotype pairs. Currently, this process is done within Filip, which makes it somewhat slow.

5.6.4. Limited testing

Filip has only been tested on DCGO protein function predictions and using the FANTOM5 data, meaning that we cannot be sure that these results with generalise outside of this. Testing on furher data (as already discussed), but also with different base predictors would be key.

5.6.4.1. Protein abundance

As noted earlier, gene expression data is a measure of mRNA abundance, not protein abundance, and it is not especially well correlated with it. Furthermore, when it is available, protein abundance data outperforms mRNA data for predicting gene function[234]. By choosing a sensible cut-off for gene expression, we do discard some of the transcriptional noise which characterises some of the difference between mRNA and protein abundance, which is good. Still, Filip would almost certainly throw away more false positive predictions if the mRNAs that are destined for degradation weren't present in the input data.

There are some attempts to predict protein abundance from mRNA abundance[235]. It would be interesting to investigate if these predictors can improve the performance of Filip.

6. Ontology

This chapter describes Ontology: a small Python package that I created for manipulating OBO ontology files. This chapter also includes some uses of ontology, for example mapping between samples and phenotypes, and other uses. For example, it is particularly useful for finding inconsistencies/disagreements between data sources, which enabled me to contribute back to improve some of the resources that Ontology relies on.

At time of writing, `Ontology v1.1.1-beta` (PyPi, GitHub, Docs) is the current release.

> **Contributions in this chapter**
>
> The contributions in this chapter include:
>
> - Creation of the Ontology package and it's documentation.
> - Contributions to improve the FANTOM5 and Uberon ontologies, based on using the package to discover data inconsistencies.

6.1. Introduction

6.1.1. Motivation

> **What are ontologies again?**
>
> Ontologies are controlled vocabularies of terms and relationships. You can read more about them in Section 3.3.

I created Ontology in order to create a high-coverage mapping between tissues and gene expression samples, which I hoped would aid in phenotype and protein function prediction. An earlier version of the package was used to create a mapping between gene expression sample names or identifiers to phenotypes that are known to affect that type of tissue for Section 5. Removing wrong-tissue predictions proved successful in improving protein function predictions, but was constrained by a low coverage, despite using one of the most extensive tissue-specific gene expression experiments. In order to improve this coverage, I needed to extend it's functionality to allow mapping more generally between samples and phenotypes according to their tissues.

There are many gene expression data sets, and the reporting for tissue metadata is not at all standardised between them. This is true even within databases of gene expression data where great care has been taken to harmonise the metadata such as the Gene Expression Atlas. If tissue type is recorded at all, it is usually manually given a label tissue using a name (e.g. "blood", "kidney"), or perhaps as part of the sample name ("blood adult donor1"). In other cases, cell type might be recorded instead (e.g. "leukocyte", "cardiac fibroblast"). In other circumstances still, the samples might be annotated to existing ontologies, and some even have their own ontologies of samples (such as FANTOM5). Names like *blood* can be useful, but if you'd like to compare across samples, then it's helpful to have a controlled vocabulary such as ontology terms: that way even a computer can figure out what *mature basophils*, *plasma* and *fibrinogen complex* have in common.

In addition to the benefits of ontologies' controlled vocabulary (the terms themselves), they also contain a wealth of additional information and links to other ontologies. For example, Uberon contains information about synonyms for different anatomical entities: the *pituitary stalk* is also known as the *infundibular stem* which is *part of the brain* that *connects to the hypothalamus*. Ontologies are therefore also sources of text that could be used to map sample names to terms. Once samples are mapped to ontologies they can leverage on all of the information inside them, for example, to find all the samples that are capable of *hormone secretion*.

6.1.2. OBO files

There are two file formats which rule the ontology world.

Open Biomedical Ontology (OBO, `.obo` files) is the format that biomedical ontologies such as Gene Ontology or Human Phenotype Ontology were originally built in. Meanwhile, the other file format is the Ontology Web Language (OWL), which is built upon XML. Although it has not always been the home of biomedical ontologies, many now release both OBO and OWL versions. Both file types are human-readable, although the OBO format is a little easier (for humans) to edit and read directly, and is generally considered easier to work with. The major benefit of the OWL format is that it is formally axiomised and there exists a large suite of tools available for performing logical reasoning (e.g. using HermiT and SPARQL)

I found that at the time of creating, I needed files which were only available in OBO format, and OBO-to-OWL converters were not able to extract all the information that I needed.

6.1.2.1. Anatomy of an OBO file

OBO files are text files. The top of an OBO file contains metadata about the ontology itself, for example it's version in terms of format (`format-version`) and contents (`data-version`), name (`ontology`), the definitions of any subsets of ontology terms (`subsetdef`), the definition of synonym types (`synonymtypedef`), among many others.

Below are some example lines from this top section of the extended Uberon ontology file:

```
format-version: 1.2
ontology: uberon/ext
data-version: uberon/releases/2021-02-12/ext.owl
subsetdef: non_informative "abstract class brought in to group ontology classes but not informative"
synonymtypedef: HUMAN_PREFERRED "preferred term when talking about an instance of this class in Homo sapiens"
```

The rest of the file has the following format, a blank line followed by `[Term]` indicates a new term is being defined, followed by different types of attributes, such as `id`, `name`, definition (`def`), external reference (`xref`), links to parent terms `is_a`, and other relationships such as `part_of` or `located_in`. See for example the *pupillary membrane* term, `UBERON:0002269`:

```
[Term]
id: UBERON:0002269
name: pupillary membrane
def: "The pupillary membrane in mammals exists in the fetus as a source of blood supply for the lens. It normally
atrophies from the time of birth to the age of four to eight weeks."
[http://en.wikipedia.org/wiki/Persistent_pupillary_membrane]
xref: FMA:77663
xref: http://en.wikipedia.org/wiki/Persistent_pupillary_membrane
xref: MA:0001293
is_a: UBERON:0000158 ! membranous layer
is_a: UBERON:0004121 ! ectoderm-derived structure
relationship: located_in UBERON:0001771 {source="MA-modified"} ! pupil
relationship: part_of UBERON:0000922 ! embryo
relationship: part_of UBERON:0001769 ! iris
```

6.1.3. Purpose

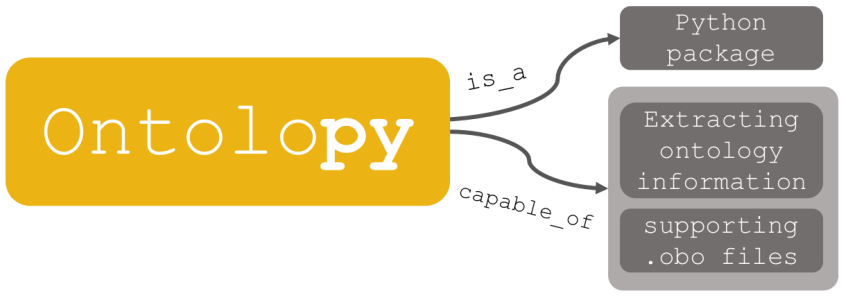


Fig. 6.1 Figure showing the purpose of Ontolopy as an ontology.

> **Purpose statement**
>
> Ontolopy makes it easier to work with OBO ontologies in Python using familiar data types such as Python `dict`s (dictionary objects) and Pandas `DataFrame`s. In particular, this supports finding and propagating relationships between ontology terms (such as tissue and phenotype terms), and enables matching of sample names to ontology terms.

To my knowledge, no other existing package fulfils this particular need, however there are of course many other tools for working with ontologies.

> **Propagating**
>
> I use the word *propagate* to applying information about one ontology term, to describe it's parents or children. For example `UBERON:0009865` (*Hatschek's pit* - part of a fish-like lancelet) is `capable_of GO:0070254` (*mucus secretion*) and also `is_a UBERON:0006846` (*surface groove*).
>
> If we propagate this relationship to the parent term, we find out that *surface grooves* can be `capable_of` *mucus secretion*.

6.1.4. Other available tools

There are a number of other tools that are available for building and logically querying ontologies, but these tend to be standalone platforms like Protégé (a Desktop platform primarily for building ontologies) or OWL-specific like OwlReady2. I tested Protégé (using it's built-in reasoner HermiT with SPARQL queries) for extracting uberon-phenotype mappings, but perhaps due to the size of the ontologies, this was prohibitively slow.

Ontologies are widely used by biomedical researchers, mostly for ontology enrichment analyses. There are easy to use tools in popular programming languages like R and Python for performing specialised analyses (such as GO enrichment, like GOATools[236], or the older goenrich) but tools for querying them generally are either very specialised or browser-based (like Ontobee[237]).

Pronto[238] is a nice Python package which is one of the exceptions to this rule, however it has some "missing" (missing for my needs, out of scope for them) functionality: being able to propagate relationships between nodes (terms).

OntoBio[239] is another Python package with similar functionality, which remains in active development. It is made by the same team as GOATools. It has a rich functionality in terms of querying common attributes of biological ontologies, for example their synonyms or definitions. Again, the missing functionality for me is to be able to merge and query ontologies for relationships.

It would be beneficial for Ontolopy to make use of Pronto or OntoBio's underlying data structures as I discuss in Section 6.7.2.4.

6.2. Functionality

This section describes the low-level functionality of Ontolopy: what it can do. For examples of how this functionality is of practical use, please see the examples sections. You can find a full and up-to-date API Reference in the documentation.

The functionality of the package can be summarised as follows: Ontolopy takes OBO files and:

1. makes them into an intuitive Python object (which subclasses a Python `dict`, meaning that you can do everything with it that you can do with this familiar and useful data type.
2. provides a set of tools for doing some useful manipulations and queries to these objects, which are particular to ontologies. This includes for example propagating relationships between terms, finding leaf/root terms, and merging ontologies.
3. Further to this, it provides an extra class for manipulating and querying the Uberon anatomy specifically.

> **Subclass**
>
> Python is an object-orientated language, meaning that it's designed so that classes can inherit from one another.
>
> If a class subclasses another, it means it inherits it's attributes and methods.

6.2.1. Structure

Ontolopy is organised into three submodules, each centred around classes with the same names: `opy.Obo()` for OBO ontology objects, `obo.Relations()` for finding relationships between terms in an ontology object, and `opy.Uberon()` for finding tissue mappings. These three submodules are automatically loaded with `import ontolopy`.

> **`import ontolopy as opy`**
>
> Note: you will see `ontolopy` shortened to `opy` in code segments.

6.2.2. Working with OBO ontologies

The `opy.obo` module contains the following callables that make it easier to work with OBO ontologies:

> **Callables**
>
> In Python callables are functions, classes, and class methods that you can call, i.e. where you use syntax like `function()` or `MyClass()` to run some code.

`load_obo`(file_loc[, ont_ids, discard_obsolete])	Loads ontology from *.obo* file at *file_loc*.
`download_obo`(data_name[, out_dir])	Download obo from a list of known locations.
`Obo`([source_dict])	Creates *Obo* ontology object from *dict* with ontology terms for keys, mapping to term attributes and relations.
`Obo.merge`(new[, prefer])	Recursively merges *new* into *self* and returns a merged *Obo* ontology.

6.2.2.1. The `Obo` class

The `Obo` class is an OBO ontology object, which subclasses `dict`. New `Obo` objects can be created from nested dictionaries. At the top level of the dictionary, keys are terms and values are dictionaries. This dictionary structure also allows you to add new terms.

Obo reference

class `ontology.obo.Obo`***(source_dict={})***

Creates *Obo* ontology object from *dict* with ontology terms for keys, mapping to term attributes and relations.

Each key/term is a dictionary with key: value pairs mapping either:

1. Attribute (*str*) to value (*str*), e.g. *'name': 'scapula'*
2. Type of relationship (*str*) to term identifiers (*list*), e.g. *'is_a': ['UBERON:0002513']*

Info: Obo stands for Open Biological Ontology: a popular file format for building biological ontologies.

`__init__`***(source_dict={})***

Initialise self from a source dictionary.

Parameters: **source_dict** – *dict* mapping terms to their attributes and relationships.

Methods

`merge`(new[, prefer])	Recursively merges *new* into *self* and returns a merged *Obo* ontology.

Attributes

`leaves`	Leaf terms are the most specific terms in the ontology; they have no children, only parents (a *set* object).
`terms`	The ontology terms (a *dict_keys* object).

Obo usage example

```
import ontology as opy

new_ontology = opy.Obo({'TERM:000001': {'name': 'Example term'}})
new_ontology['TERM:000002'] = {'name': 'Second example term', 'is_a': ['TERM:000001']}
```

6.2.2.2. Merging ontologies

It's also possible to merge (a list of) ontologies into the base ontology. This can be useful for investigating relationships between ontologies. For example, to find relationships between samples and tissues, that might go via cells, you may want to merge a sample ontology, cell, and tissue ontology to find all possible relationships.

Obo.merge reference

`Obo.merge`***(new, prefer='self')***

Recursively merges *new* into *self* and returns a merged *Obo* ontology.

Parameters:
- **new** – Obo object (or list of objects) to add.
- **prefer** – prefer 'self' (base Obo) or 'new' (new Obo)

Return merged: A merged Obo

6.2.2.3. Loading ontologies from file

While creating ontologies from dictionaries is useful for adding bespoke terms, most of the time we want to load an official and curated OBO from a file.

load_obo reference

`ontology.obo.load_obo`***(file_loc, ont_ids=None, discard_obsolete=True)***

Loads ontology from *.obo* file at *file_loc*.

Parameters:
- **file_loc** – file location - path to stored obo file.
- **ont_ids** – list of ontology ids, e.g. *['UBERON', 'CL']*
- **discard_obsolete** – if True discard obsolete terms.

Returns: *Obo* ontology object.

6.2.2.4. Downloading OBO files

It's also possible to download OBO files, either from a list of popular OBO files by name, or via a URL.

download_obo reference

`ontology.obo.download_obo`***(data_name, out_dir='../data/')***

Download obo from a list of known locations.

Parameters:
- **data_name** – Name of OBO you wish to download.
- **out_dir** – Directory in which to save OBO file.

Return out_file: path to saved file.

6.2.3. Finding relationships

The most key functionality in Ontolopy is the ability to infer relationships between terms, across ontologies (be it between tissue terms and phenotype terms, or something else). This functionality is inside the `opy.relations` module and handled by the Relations class.

Relations reference

`relation_path_to_text`(relation_path, ont)	Converts from a relation string e.g. "UBERON:123913.is_a~UBERON:1381239" to a text version,
`Relations`(allowed_relations, ont[, sources, ...])	

6.2.3.1. The `Relations` class

The `Relations` class finds relationships of certain types between sources and targets. It subclasses a Pandas DataFrame since that is a convenient and familiar format for the relationship information to be returned.

***class* `ontology.relations.Relations`(*allowed_relations: list*, *ont*, *sources=None*, *targets=None*, *source_targets=None*, *excluded=None*, *col_names=None*, *mode='any'*)**

`__init__`(*allowed_relations: list*, *ont*, *sources=None*, *targets=None*, *source_targets=None*, *excluded=None*, *col_names=None*, *mode='any'*)

Pandas Dataframe containing relationships between *sources* and *targets* terms according to *ont*. Finds relationships that do not pass through *excluded* terms and uses only *allowed_relations*. We keep looking until we find a relation to a target (if mode == 'any') or we run out of leads.

Parameters:

- **allowed_relations** – a list of allowed relations, e.g. ['is_a', 'part_of']
- **sources** – list of sources. For mode *all* must be a list of source-target tuple airs.
- **mode** – 'any' or 'all' - 'all is looking for specific term1-term2 pairs, while 'any' is looking for any relationship between something in specific source and anything in targets.
- **targets** – list of targets.
- **source_targets** – list of tuples of source-target pairs. Do not provide source or targets if using this parameter. Only runs in "all" mode.
- **ont** – Obo ontology object.
- **excluded** – a list/set of terms which are explicitly not being searched for (which may otherwise match the targets). Useful e.g. if we want to look for any tissue targets with prefix 'UBERON', except for very general ones. Does not allow relationships that pass through this term.
- **col_names** – Alternative column names for the output of Relations Data Frame, by default is ['from', 'relation_path', 'relation_text', 'to']

To find relationships, the code loops through sources, and for each source it will look at the `allowed_relations` to find relationships with other terms, then for each of these terms it will look for relationships with other terms in the same manner, etc.

Internally, Ontology stores these relationships as a list of strings, where each string details the relations between the source term and other terms, e.g. `UBERON:123913.is_a~UBERON:1381239.is_a~UBERON:987890`. Let's call these strings *relation paths*.

Cyclic relationships are not permitted (a term can only be present in a relation path once). Relationships continue to be searched for until either the ontology provided can no longer add any new relation paths OR we found what we were looking for.

In "any" mode, finding what we're looking for means finding any target term as the last term in the relation string, while in "all" mode, we must find all target terms for the source term.

The `mode` parameter can be either `any` or `all`, and this represents whether we are looking for relations from our source terms to any one target term, or to all target terms for which we can find a relationship. It is much quicker to run in "any" mode, so this mode is the default, and it is preferable when we simply need the most direct mapping between our source and target terms, for example we want to know which (one) tissue does the sample map to best?

The "all" mode tends to be more useful when we are equally interested in the targets as the source terms for example: when looking at mappings between tissues and phenotypes, there is likely to be many different phenotypes that a tissue can exhibit and we are equally interested in all of them.

Provide either `sources` and `targets` OR `source-targets`. It's possible to provide a list of `sources` and a list of `targets`, OR a list of tuple `source-targets`. It does not make sense to provide both. The latter option only works in `all` mode: i.e. we are interested in all source-target pairs. Essentially, the `sources-targets` option provides a quicker way of running `Ontology` in "all" mode when we know in advance which specific pairs of sources and targets we are interested in. If `sources` and `targets` are provided and `mode==all`, then `Ontology` will generate a combination of all possible sources and targets (removing `excluded` target terms if provided).

6.2.3.2. Converting "relation paths" to text

Since relationships are internally stored as relation paths as explained above, it is useful to turn these strings into more readable text, which is what the `relation_path_to_text` function does.

relation_path_to_text reference

`ontology.relations.relation_path_to_text`(*relation_path*, *ont*)

Converts from a relation string e.g. "UBERON:123913.is_a~UBERON:1381239" to a text version,

e.g. "heart is a circulatory organ".

Parameters:

- **ont** – opy.Obo() ontology object.
- **relation_path** – path describing the relationship between two terms, e.g. "UBERON:123913.is_a-UBERON:1381239"

Returns:

6.2.4. Creating Uberon Mappings

The `opy.uberon` submodule contains the specific tools for working with the Uberon ontology: finding mappings between tissues and phenotypes via ontology terms by making use of the Relations class, as well as doing this mapping using text, and comparing these two mappings. The vast majority of this functionality sits in the `Uberon` class.

`uberon_from_obo`(obo)	Creates an *Uberon* object from an *Obo* object.
`Uberon`()	An UBERON-specific ontology object.

6.2.4.1. The `Uberon` class

Calling the `Uberon` class itself simply checks if there are any `Uberon` terms in the merged ontology, and then allows the ontology to be used to create Uberon sample-to-tissue mappings, through class methods (which should be called separately).

There are three parts to the process in creating Uberon mappings, the functionality for which lives in three different `Uberon` class methods:

1. **Mapping via name:** Map from sample-to-tissue via informal tissue names given in experimental design information (e.g. "eye stalk") to an Uberon term (`UBERON:0010326`, Optic Pedicel).
2. **Mapping via ontology term:** Map from CL cell types (e.g. `CL:0000235`, Macrophage), sample ontology term to Uberon tissues (e.g. `UBERON:0002405`, Immune system). Or from sample ontology terms (like FANTOM terms, such as `FF:10048-101G3`, *Smooth Muscle, Adult, Pool1*) to Uberon terms (`UBERON:0001135`, Smooth Muscle Tissue). Returns relationships between source term and Uberon term.
3. **Create sample-to-tissue mappings and disagreements between mappings** based on (1) and (2).

class `ontology.uberon.Uberon`

An UBERON-specific ontology object.

`__init__`()

Initialise self from a source dictionary.

Parameters: **source_dict** – *dict* mapping terms to their attributes and relationships.

Methods

`__init__`()	Initialise self from a source dictionary.
`sample_map_by_ont`(sample_ids[, exclude, ...])	Map tissues from sample names to uberon identifiers.
`sample_map_by_name`(sample_names[, to, ...])	Map tissues from sample identifiers to uberon identifers.
`get_overall_tissue_mappings`(map_by_name, ...)	Combines the two mappings *map_by_name* and *map_by_ont* to create an overall mapping and disagreements.

6.2.4.2. Mapping from sample to tissue via name using `Uberon.sample_map_by_name`

Informal tissue names are mapped from Uberon term identifiers by checking for exact name matches to Uberon term names and their synonyms in the extended Uberon ontology.

If an exact match does not exist, individual words from the phenotype term name or synonyms are then searched for exactly. First stop words are removed, using the base list in the Natural Language Toolkit (`nltk`) Python Package[240] (e.g. and, or), and a small number of manually curated phenotypic stopwords (e.g. "phenotype", "abnormality"). This would mean that the HP (Human Phenotype) term "abnormality of the head and neck" would search for the words "head" and "neck" in the UBERON terms, and would be mapped the terms of the same name (but never to "neck of radius" - which is related to bone). In cases where multiple terms are found, a common parent would be searched for, in this case the result is "craniocervical region" .

Stop words

Stop words are words that are filtered out before processing text using Natural Language Processing (NLP) methods. These are usually very common words (e.g. "and", "the"), or word which are meaningless in the context of the analysis.

Uberon.sample_map_by_name reference

`Uberon.sample_map_by_name`(*sample_names*, *to=None*, *col_names=None*, *xref=None*, *synonym_types=None*)

Map tissues from sample identifiers to uberon identifers.

Parameters:
- **sample_names** – map from sample identifiers to tissue/sample descriptors/names for values. May be *dict* or *pd.Series*
- **to** – list of ontology prefixes that you want to map to.
- **xref** – An ontology identifier (e.g. FMA) the presence of which denotes a preferred term.
- **col_names** – Column names of returned relationships

Returns:

6.2.4.3. Mapping from sample to tissue via ontology term using `Uberon.sample_map_by_ont`

The `sample_map_by_ont` function uses the Relations class in "any" mode to find relationships via ontologies in much the same way described above. This is essentially a wrapper that provides convenient default settings for allowed relations and targets.

Mappings can be made via any term in the merged ontology, which allows mappings that cannot be made through Uberon alone, for example: Macrophage - monocyte derived, donor3 `is_a` Human macrophage sample `derives_from` Macrophage `is_a` Monocyte `is_a` Leukocyte `part_of` Immune System, which means this sample is derived from part of the immune system.

Uberon.sample_map_by_ont reference

`Uberon.sample_map_by_ont`(*sample_ids: list*, *exclude=None*, *relation_types=None*, *to=None*, *child_mapping=False*)

Map tissues from sample names to uberon identifiers. Will only work if ontology contains Uberon + Sample terms.

Parameters:
- **sample_ids** – list of sample identifiers
- **exclude** – list of tissues to exclude, i.e. because they are too general.
- **relation_types** – list of relation types in ontology that relate to position in body.
- **to** – list of ontology prefixes that you want to map to.
- **child_mapping** – If True, searches children instead of parents.

Returns:

Mapping by term: child mapping Some samples may be pools of cell types that may come from more than one anatomical location. In this case, there will be no regular mapping, since no parent terms will have a mapping to a tissue. In this case, we can look at tissue mappings (in the usual way, described above), for all of the children of our parent term of interest. I call this mode "child mapping" and it is off by default.

So, for example *melanocytes* are are melanin-producing cells found in many different places in the body (skin, hair, heart), and therefore they don't (nor any of their parents) map to a specific Uberon term. If we choose `child_mapping==TRUE`, then for this term, we will get a list of all Uberon terms that cells of this type can come from. This mode isn't currently used in the context of the rest of this thesis.

6.2.4.4. Getting overall mappings and finding disagreements using `Uberon.get_overall_tissue_mappings`

As described, Ontolopy has two methods of mapping to tissues, and it also provides a method of harmonising these two mappings, and for finding any disagreements between them. This can be very useful for revealing logical inconsistencies in either the mappings or the ontologies (as was the case in the FANTOM5 example).

Uberon.get_overall_tissue_mappings reference

`Uberon.get_overall_tissue_mappings`(*map_by_name*, *map_by_ont*, *rel=None*)

Combines the two mappings *map_by_name* and *map_by_ont* to create an overall mapping and disagreements.

Parameters:	• **map_by_name** (class: *pd.DataFrame*) – mapping from sample to tissue via sample name, from *Uberon.sample_map_by_name*. • **map_by_ont** (class: *pd.DataFrame*) – mapping from sample to tissue via sample ontology ID, from *Uberon.sample_map_by_ont*.
Parm rel:	list of relation strings allowed between name and ontology mappings to count as not a disagreement.
Returns:	(*overall_mapping*: mapping from sample to tissue combining both sources, *disagreements*: disagreements between "by name" and "by ontology" mappings.)
Return type:	(class: *pd.DataFrame*, class: *pd.DataFrame*)

6.3. Ontolopy tools and practices

This section briefly describes the tools and practices that Ontolopy is built upon.

6.3.1. Practices

Development philosophy:

Ontolopy aspires to Research Software Engineering best practice, including:

- Automated testing with pytest which are continuously integrated with GitHub Actions.
- Semantic Versioning to make the package versions informative and useful for others.
- Thorough documentation, which is versioned. This means that you can always reach the documentation corresponding to the version of the software you are using - you can access this at `/versions/{tagged_version}`, e.g. `v1.0.2-beta` is here.) The documentation is also built automatically using GitHub Actions.
- Keeping a small number of dependencies, which are: `numpy`[241] and `pandas`[242] for general data manipulation, and `validators` (for validating URLs)

Continuous Integration

Continuous integration, delivery, and deployment enable running tests and updating packaging frequently by automatically doing these things when changes are made to a version-controlled repository[128].

Open Source:

Ontolopy is Open Source (with an MIT license), and available on GitHub. This means that:

- anyone can contribute to it. I provided developer guidance and "good first issues" to reduce the barrier to this.
- anyone can download, use, reuse, or adapt the source code for their own work. This is made easier by the fact that Ontolopy is distributed via the Python Package Index.

Style:

Ontolopy uses consistent programming style and conventions to make it easier for others to work with (these were adapted from the MetaWards package developer guide[243]:

- Python-style naming conventions:
  - Packages: lowercase (single word)
  - Classes: CamelCase
  - Methods, Functions, Variables: snake_case
- Functions with leading underscores (e.g. `_extract_source()`) are meant for internal use only.
- Relative imports should be used at all times, with imports ideally delayed until they are needed.

6.3.2. Tools

Packaging:

Packaging is carried out automatically using GitHub actions whenever a new version of the software is "tagged" via GitHub. This uses the **`twine`**[244] Python Package.

Testing:

Tests are automatically run whenever Ontolopy code is changed on either the GitHub `main` or `dev` (development) branches. This is achieved with GitHub Actions and the **`pytest`**[245] Python Package.

Logging:

Python's in-built **`logging`** module is used to integrate logging messages from dependencies as well as adding useful logging messages for Ontolopy users. This allows informative messages to be printed to the console or to a log file.

Documentation:

Ontolopy's documentation is hosted on GitHub pages here, and built using **`Sphinx`** with the **`pydata-sphinx-theme`** theme[246], and it is automatically built using GitHub Actions whenever there are changes to the development branch or when there is a new release. It also makes use of following tools:

- **`peaceiris`**'s GitHub pages action - to automatically update a GitHub pages site in a GitHub Action.
- **`sphinx.ext.autosummary`** - to automatically build an API Reference from docstrings in code.

In addition to the above tools which were built by others, I wrote a small local sphinx extension to create the versioned documentation. This, in turn, uses gitpython and pygithub.

6.4. Example uses: mapping samples to diseases or phenotypes

There are a number of potential uses for Ontolopy. In this section, I show two simple examples to demonstrate this usefulness. These show how Ontolopy can be used to:

1. Find disease-related samples
2. Find samples of pluripotent stem cells (cells that can turn into different tissue types)

Then in the next section I give the more detailed and complex example of creating a mapping between samples and tissues (which is what Ontolopy was created for specifically), and how this was used to find inconsistencies in the FANTOM5 data.

6.4.1. Inputs

The examples of using Ontolopy in this Chapter use input files from FANTOM5[247] (for samples) and Uberon[109] (the cross-species anatomy ontology).

6.4.1.1. FANTOM5

Large experiments sometimes include an ontology of samples instead of or (more frequently) in addition to a samples information file. The data from the FANTOM5 experiment[247] is one such example of this. I have already explained the FANTOM5 data in more detail but for now the only things we need to keep in mind are that:

1. The FANTOM5 experiment measures transcript expression in a wide variety of samples, across many tissue and cell types.
2. FANTOM5 provide an ontology of samples as well as a sample information file (containing short text descriptions of samples).

Choosing ont_ids

When we read in ontologies using Ontolopy, if you do not provide `ont_ids`, Ontolopy will keep all ontology term identifiers of the form `LETTERS:NUMBERS`, but often there are many external reference terms (`xref`s) that make the ontology object larger for no gain, so it's recommended to provide them.

When providing `ont_ids`, it's important that you keep all terms that you're interested in, as everything else is discarded.

	Source Name	Charateristics [description]	Characteristics [catalog_id]	Characteristics [Category]	Chracteristics [Species]	Characteristics [Sex]	Characteristics [Age]	Characteristics [Developmental stage]	Characteristics[Tissue]	Charateristics [ff_ontology]
FF:10002-101A5	10002-101A5	SABiosciences XpressRef Human Universal Total ...	B208251	tissues	Human (Homo sapiens)	mixed	NaN	UNDEFINED	unclassifiable	
FF:10016-101C7	10016-101C7	heart, adult, pool1	0910061 -7	tissues	Human (Homo sapiens)	mixed	NaN	70,73,74 years old adult	heart	
FF:10018-101C9	10018-101C9	liver, adult, pool1	0910061 -9	tissues	Human (Homo sapiens)	mixed	NaN	64,69,70 years old adult	liver	

Fig. 6.2 An excerpt of the FANTOM sample info file, showing sources of text-based information, e.g. "heart, adult, pool1" in the *Charateristics [description]* field, and mapping to ontology term in the index (`FF:10016-101C7`).

Fig. 6.2 shows an excerpt of the FANTOM Samples Information file. This kind of file is typical of transcription experiments: a csv file containing hand-entered text-based information, using non-specific lay terms for samples e.g. "heart".

The FANTOM ontology file links specific FANTOM samples to more general types of FANTOM samples and to Uberon tissues and CL cell types.

For example an excerpt of the FANTOM ontology OBO file is:

```
[Term]
id: FF:0000076
name: hepatic sinusoidal endothelial cell sample
namespace: FANTOM5
synonym: "hepatic sinusoidal endothelial cell" EXACT []
is_a: FF:0000002 ! in vivo cell sample
intersection_of: FF:0000002 ! in vivo cell sample
intersection_of: derives_from CL:1000398 ! endothelial cell of hepatic sinusoid
intersection_of: derives_from UBERON:0001281 ! hepatic sinusoid
relationship: derives_from CL:1000398 ! endothelial cell of hepatic sinusoid
relationship: derives_from UBERON:0001281 ! hepatic sinusoid
created_by: tmeehan
creation_date: 2011-03-01T04:51:50Z
```

6.4.1.2. Uberon

As I mentioned in Section 3.3.3.2, Uberon is a cross-species anatomy ontology with excellent linkage to other ontologies. As we can see above, the FANTOM5 ontology links FANTOM samples to Uberon. This means that the Uberon[109] extended ontology OBO file can then be used to further link the samples to human disease or gene ontology terms.

For example, here is an excerpt of the Uberon extended OBO file (non-consecutive lines for brevity), showing how the Uberon extended ontology could be used to link a FANTOM sample to a GO term:

```
[Term]
id: UBERON:0001281
name: hepatic sinusoid
alt_id: UBERON:0003275
def: "Wide thin-walled blood vessels in the liver. In mammals they have neither veinous or arterial markers."
[http://en.wikipedia.org/wiki/Hepatic_sinusoid, ZFIN:curator]
synonym: "hepatic sinusoids" RELATED []
synonym: "liver hepatic sinusoids" EXACT [EHDAA2:0000999]
synonym: "liver sinusoid" EXACT []
intersection_of: part_of UBERON:0002107 ! liver
relationship: part_of UBERON:0004647 ! liver lobule
relationship: part_of UBERON:0006877 {source="https://github.com/obophenotype/uberon/wiki/Inferring-part-of-
relationships"} ! vasculature of liver
property_value: homology_notes "(...) the amphibian liver has characteristics in common with both fish and terrestrial
vertebrates. (...) The histological structure of the liver is similar to that in other vertebrates, with hepatocytes
arranged in clusters and cords separated by a meshwork of sinusoids and the presence of the traditional triad of portal
venule, hepatic arteriole, and bile duct.[well established][VHOG]" xsd:string {date_retrieved="2012-09-17",
external_class="VHOG:0000708", ontology="VHOG", source="http://bgee.unil.ch/", source="DOI:10.1053/ax.2000.7133
Crawshaw GJ, Weinkle TK, Clinical and pathological aspects of the amphibian liver. Seminars in Avian and Exotic Pet
Medicine (2000)"}

[Term]
id: UBERON:0002107
name: liver
def: "An exocrine gland which secretes bile and functions in metabolism of protein and carbohydrate and fat,
synthesizes substances involved in the clotting of the blood, synthesizes vitamin A, detoxifies poisonous substances,
stores glycogen, and breaks down worn-out erythrocytes[GO]." [BTO:0000759, http://en.wikipedia.org/wiki/Liver]
synonym: "iecur" RELATED LATIN [http://en.wikipedia.org/wiki/Liver]
is_a: UBERON:0002365 {source="BTO", source="EHDAA2", source="GO-def"} ! exocrine gland
is_a: UBERON:0004119 ! endoderm-derived structure
is_a: UBERON:0005172 ! abdomen element
is_a: UBERON:0006925 ! digestive system gland
disjoint_from: UBERON:0010264 ! hepatopancreas
relationship: contributes_to_morphology_of UBERON:0002423 ! hepatobiliary system
relationship: produces UBERON:0001970 ! bile
relationship: site_of GO:0002384 ! hepatic immune response
relationship: site_of GO:0005978 ! glycogen biosynthetic process
relationship: site_of GO:0005980 ! glycogen catabolic process
property_value: external_definition "Organ which secretes bile and participates in formation of certain blood proteins.
[AAO]" xsd:string {date_retrieved="2012-06-20", external_class="AAO:0010111", ontology="AAO", source="AAO:BJB"}
property_value: function_notes "secretes bile and functions in metabolism of protein and carbohydrate and fat,
synthesizes substances involved in the clotting of the blood, synthesizes vitamin A, detoxifies poisonous substances,
stores glycogen, and breaks down worn-out erythrocytes[GO]." xsd:string
```

These excerpts show how `FF:0000076` (*hepatic sinusoidal endothelial cell sample*s) are `derived_from` the *hepatic sinusoid* which is `part_of` *the liver* the `site_of` *hepatic immune response*, *glycogen biosynthetic process* and *glycogen catabolic process*. There are many such relationships in these files: Ontolopy provides an easy way of extracting these.

> ⚠ **Warning: not all relationships are easy to interpret**
>
> In this case, we do not have enough information to infer that *hepatic sinusoidal endothelial cell sample*s are a `site_of` (for example) the *hepatic immune response* because it could be another, disjoint, part of the liver that is the site of this. We can also not rule it out: a more specific annotation in the future might enable us to find this out with these files.
>
> However, this information could still be useful in Computational Biology. If we don't know exactly where a process takes place, we may want to cast a wider net and look at all samples which are part of a larger tissue we know exhibits the process we are interested in.
>
> This is something to be aware of in general when using Ontolopy: if you are only interested in straight-forward relationships, then you often need to think carefully about the types of relationships that you ask for: `part_of` relationships need particular care.

6.4.2. Example 1: Finding disease-related samples

This first example shows a simple use-case of Ontolopy, where we are looking for relationships to any term in an ontology: in this case any relation to a Disease Ontology term (representing human diseases). This Ontolopy query can be done with only the FANTOM ontology.

As we just saw, to extract relationships from ontologies (whether using Ontolopy or with any other method), you have to think about the types of relations that you are interested in. For example, if we are interested in finding samples which are models for `DOID` disease terms, then we want to ask for `DOID` targets only, and `is_a` and `is_model_for` relationships only.

```
Finds 566 disease relations in 0.057 seconds
```

Ontolopy can quickly (less than half a second) retrieve this information, very compactly (in one line of code if we wanted). We can see an excerpt of the output in Fig. 6.3. This would be useful for example if we wanted to remove disease samples from the samples we were looking at.

from	relation_path	relation_text	to
FF:10050-101G5	FF:10050-101G5.is_model_for~DOID:5844	heart, adult, diseased post-infarction, donor1 is model for myocardial infarction	DOID:5844
FF:10051-101G6	FF:10051-101G6.is_model_for~DOID:114	heart, adult, diseased, donor1 is model for heart disease	DOID:114
FF:10399-106A3	FF:10399-106A3.is_a~FF:0101883.is_a~FF:0100740.is_model_for~DOID:8692	acute myeloid leukemia (FAB M5) cell line:THP-1, rep3 (fresh) is a acute myeloid leukemia cell line sample is a myeloid leukemia cell line sample is model for myeloid leukemia	DOID:8692
FF:10400-106A4	FF:10400-106A4.is_a~FF:0101883.is_a~FF:0100740.is_model_for~DOID:8692	acute myeloid leukemia (FAB M5) cell line:THP-1, rep1 (revived) is a acute myeloid leukemia cell line sample is a myeloid leukemia cell line sample is model for myeloid leukemia	DOID:8692
FF:10405-106A9	FF:10405-106A9.is_a~FF:0101883.is_a~FF:0100740.is_model_for~DOID:8692	acute myeloid leukemia (FAB M5) cell line:THP-1, rep3 (thawed) is a acute myeloid leukemia cell line sample is a myeloid leukemia cell line sample is model for myeloid leukemia	DOID:8692

Fig. 6.3 The top 5 lines of the disease-related FANTOM samples that Ontology found.

6.4.3. Example 2: Find tissues that are capable of cell differentiation

This second example showcases a different and slightly more complex example where:

1. **We want to look for relations to a specific term** rather than a general one: in this case `GO:0030154` *cell differentiation*.
2. **We need to use an external ontology** (Uberon), so we use the `merge` function.
3. **We need to chain two queries and stick them together**. The `derives_from` relation in the context of the FANTOM5 ontology can mean "extracted from" or "extracted from and then do lots of things to it". To rule out the latter type of samples we only want to ask for *in vivo* samples (`is_a` *in vivo sample* `FF:0000002`) that `derives_from` cell types that are `capable_of` *cell differentiation* (`GO:0030154`).

> **The difference between "derives from" and "develops from"**
>
> The Uberon extended ontology contains relations **`derives_from`**: a very general term that just means one comes from the other in some sense and **`develops_from`** which means that the two are developmentally connected, i.e. `CL:0000005` *fibroblast neural crest derived* cell `develops_from` `CL:0000008` *migratory cranial neural crest cell*, but `FF:0100003` *intestinal cell line sample* `derives_from` `UBERON:0000160` *intestine*.

```
Finds 254 relations to cell differentiation in 0.108 seconds
```

from	relation_path	relation_text	to
FF:11214-116A8	FF:11214-116A8.is_a~FF:0000094.derives_from~CL:0002569.is_a~CL:0000134.is_a~CL:0000048.is_a~CL:0000034.is_a~CL:0011115.capable_of~GO:0030154	Mesenchymal stem cell - umbilical, donor0 is a human mesenchymal stem cell of umbilical cord- Sciencell sample derives from mesenchymal stem cell of umbilical cord is a mesenchymal stem cell is a multi fate stem cell is a stem cell is a precursor cell capable of cell differentiation	GO:0030154
FF:11224-116B9	FF:11224-116B9.is_a~FF:0000024.derives_from~CL:0000576.is_a~CL:0011026.is_a~CL:0011115.capable_of~GO:0030154	CD14-positive Monocytes, donor1 is a human CD14-positive monocyte sample derives from monocyte is a progenitor cell is a precursor cell capable of cell differentiation	GO:0030154
FF:11227-116C3	FF:11227-116C3.is_a~FF:0000044.derives_from~CL:0000576.is_a~CL:0011026.is_a~CL:0011115.capable_of~GO:0030154	Dendritic Cells - monocyte immature derived, donor1, rep1 is a human monocyte immature derived dendritic cell sample derives from monocyte is a progenitor cell is a precursor cell capable of cell differentiation	GO:0030154
FF:11229-116C5	FF:11229-116C5.derives_from~CL:0000576.is_a~CL:0011026.is_a~CL:0011115.capable_of~GO:0030154	CD14+ monocyte derived endothelial progenitor cells, donor1 derives from monocyte is a progenitor cell is a precursor cell capable of cell differentiation	GO:0030154
FF:11240-116D7	FF:11240-116D7.is_a~FF:0000165.derives_from~CL:0000594.is_a~CL:0000680.is_a~CL:0000055.is_a~CL:0011115.capable_of~GO:0030154	Skeletal Muscle Satellite Cells, donor1 is a human skeletal muscle satellite cell sample derives from skeletal muscle satellite cell is a muscle precursor cell is a non-terminally differentiated cell is a precursor cell capable of cell differentiation	GO:0030154

Fig. 6.4 An excerpt of the output of Ontolopy's found FANTOM samples that are or derive from cells that are `capable_of` cell differentiation (`GO:0030154`).

Again Ontolopy can retrieve this information compactly (2 lines of code), and in less than half a second. An excerpt of the output is shown in Fig. 6.4. This would be useful if we wanted to look at expression in tissues that are capable of cell differentiation, for example.

6.5. Example use: mapping samples to tissue-related phenotypes

This section presents a more sizable example of using Ontolopy, the task it was developed for: mapping from samples to tissue-related phenotypes. This is a substantial challenge, in part because it requires:

1. Mapping over many different types of ontology terms (`FF`, `UBERON`, `CL`, `GO`).
2. Using more complex relations such as `part_of`.
3. Mapping from text as well as using the mapping from ontology, which then requires combining the two mappings into an overall mapping and investigating any disagreements between mappings.

This task can be generally divided into the following parts:

1. Creating sample (`FF`) to tissue (`UBERON`) mapping, including looking at disagreements between mappings.
2. Tissue (`UBERON`) to phenotype (`GO` Biological Process - GOBP) mapping.
3. Combining the above, to create the final sample (`FF`) to tissue-related phenotype (`GO`) mapping.

We are using the same input data as described in the previous section for this example.

6.5.1. Creating sample-to-tissue mappings

To create the mapping between FANTOM sample ID (`FF:XXXXX-XXXXX`) and tissue (`UBERON:XXXXXX`), we use the `Uberon` class. The `Uberon` class has three useful functions for creating this mapping:

1. `sample_map_by_ont`: creates a mapping via ontology.
2. `sample_map_by_name`: creates a mapping via sample or tissue names.
3. `get_overall_tissue_mappings`: combines the two mappings to create a more comprehensive overall mapping.

6.5.1.1. Load data and pre-filter

In order to do this, I load the input FANTOM5 ontology and sample information files.

6.5.1.2. Mapping by ontology

The `sample_map_by_ont` function is a wrapper function which calls `relations.Relations`, and excludes too-general Uberon tissues such as anatomical structure, tissue, anatomical system, embryo, and multi fate stem cell. We use a merged sample (FANTOM) and tissue (Uberon) ontology as input.

The inclusion of the Cell Ontology (CL) terms (which are included in the Uberon OBO file) is important to retrieve a mapping for as many samples as possible. Table 6.1 shows how the inclusion of CL terms in the input ontology significantly changes the mapping coverage, and that mapping via ontology alone (with CL terms used) is fairly good.

Mapping name	Number (and percentage) of all mapped samples	Number (and percentage) of all unmapped samples	Run time
By ontology: using Uberon tissues only	**441 (24.28%)**	**1375 (75.72%)**	**0.11 seconds**
By ontology: using Uberon tissues and CL cells	**1457 (80.23%)**	**359 (19.77%)**	**0.13 seconds**
By name: using tissue column	**1263 (69.55%)**	**553 (30.45%)**	**8.99 seconds**
Combined: combining by name and by ontology including CL	**1652 (90.97%)**	**164 (9.03%)**	**N/A**

Table 6.1 Table comparing the difference in coverage (mappable samples) for different mapping techniques.

The unmapped samples do contain some samples that we wouldn't expect to be able to map to tissue (defined as those that do not have a tissue provided in the samples information file, or that are labelled as `unclassifiable`, `UNDEFINED_TISSUE_TYPE`, or `ANATOMICAL SYSTEM`): these account for **161** of the **359 (19.77%)** unmapped samples. This, however, leaves **198** samples which we would expect to map, spread across **16** tissues (**caudate nucleus, blood, placenta, bone, chorioamniotic membrane, ovary, lung, retroperitoneum, soft tissue, skin, connective tissue, thyroid, stomach, skeletal muscle, Buffy coat, and bone marrow**).

from	Charateristics [description]	Characteristics[Tissue]	Characteristics [Cell type]
FF:10379-105H1	caudate nucleus, adult, donor10258	caudate nucleus	CELL MIXTURE - tissue sample
FF:10422-106C8	Burkitt's lymphoma cell line:DAUDI	blood	b cell
FF:10558-107I9	osteosarcoma cell line:HS-Os-1	bone	osteoblast
FF:11794-124C3	CD4+CD25+CD45RA- memory regulatory T cells exp...	blood	T cell

Fig. 6.5 Four FANTOM5 samples that we would expect to be able to map using Ontolopy based on the information in the samples information file (selected columns shown here), but that do not map using Ontolopy with the FANTOM5 ontology.

Fig. 6.5 show four examples of such samples. The existence of such tissues, means that mapping via name as well as by ontology could prove useful.

6.5.1.3. Mapping by name

The `Uberon.sample_map_by_name` function simply looks up the strings provided (in this case those from the `Characteristics[Tissue]` column of the sample information file) and checks if any Uberon terms in the provided ontology has a matching name or synonym. The term name is preferred over synonyms, and where there are no exactly matching term names, but there are multiple possible synonyms (e.g. *bladder* is a synonym for *urinary bladder* and *bladder organ*), we decide by whether either of the terms are linked to the Foundational Model of Anatomy (FMA) ontology, as this is a human-specific ontology by using the `xref='FMA'` option. Since we are only looking for Uberon terms, it doesn't make any difference whether we use the "tissue only" or "including CL" versions of the Uberon ontology that we read in earlier, aside from a negligible difference in run time).

Ontolopy restricts the tissues mapped by name to `NARROW`, `EXACT`, and `BROAD` synonyms (other synonyms include "RELATED"). These synonyms usually includes what we want, but will miss some less closely related synonyms. Looking at the tissues that are unmapped by name can help us identify any that we might want to treat differently. For example, *cartilage* doesn't map to `UBERON:0002418` *cartilage tissue* as the synonym *cartilage* is `RELATED`. It's also useful to see that the formatting of the some names, e.g. "Fingernail (including nail plate, eponychium and hyponychium)" and "eye - vitreous humor" prevent the Ontolopy algorithm from recognising the names. We could map these to more standardised names with a dictionary and rerun the algorithm if we had no other option, but in this case we simply use the by-ontology mapping to map terms with unmappable tissue names.

FF ID	Charateristics [description]	Characteristics[Tissue]	Mapped by ontology	Mapped by name
FF:10040-101F4	frontal lobe, adult, pool1	frontal lobe	frontal cortex	frontal lobe
FF:10055-101H1	uterus, fetal, donor1	uterus	embryonic uterus	uterus
FF:10075-102A3	lung, right lower lobe, adult, donor1	lung	lower lobe of right lung	lung

Fig. 6.6 Comparisons of sample mappings by ontology (inlcluding CL) and by name (by tissue column of samples information file), illustrating the tendency of by-name mappings to be less precise.

As Table 6.1 shows, mapping by name function is quite slow, taking **8.99 seconds**. The mappings coverage is **1263 (69.55%)**.

This measure doesn't show that the mapping is constrained to be less precise than the mapping via ontology - of course this is particular to the data as it depends on how the tissues were labelled. In this case, FANTOM5 labelled the tissues fairly broadly, so we see examples like `FF:10075-102A3` (*lung, right lower lobe*) and `FF:14331-155F2` (*Fibroblast - Aortic Adventitial*) in Fig. 6.6 - this is particularly common when the samples are cell types.

In addition, there are samples that are not usefully mapped at all (or completely missing a mapping) using the by-name approach, that we can get by-ontology. One subset of these is the tissues which are labelled "ANATOMICAL SYSTEM": these map to the Uberon term *anatomical system*, but this is too general to be useful. Fig. 6.7 shows how these terms can be mapped to more localised terms

This also shows the one limitation of mapping by ontology, in that we might expect that samples such as `FF:11936-126A1` to be mapped to the more specific (and useful) `UBERON:0001997` *olfactory epithelium*. This is not the case since there is a "missing" annotation between `CL:0002167` *olfactory epithelial cell* and *olfactory epithelium*, since the definition of olfactory epithelial cell is still under discussion and development.

from	Charateristics [description]	Characteristics[Tissue]	Mapped by ontology	Mapped by name
FF:11966-126D4	Smooth muscle cells - airway, control, donor1	ANATOMICAL SYSTEM	respiratory system smooth muscle	anatomical system
FF:11941-126A6	Mast cell, expanded, donor8	ANATOMICAL SYSTEM	immune system	anatomical system
FF:11937-126A2	gamma delta positive T cells, donor1	ANATOMICAL SYSTEM	immune system	anatomical system
FF:11936-126A1	Olfactory epithelial cells, donor4	ANATOMICAL SYSTEM	epithelium	anatomical system
FF:11930-125I4	Mallassez-derived cells, donor3	ANATOMICAL SYSTEM	jaw region	anatomical system
FF:11927-125I1	Fibroblast - Gingival, donor9 (control)	ANATOMICAL SYSTEM	gingiva	anatomical system

Fig. 6.7 An example of *ANATOMICAL SYSTEM* tissue samples, with tissue-specific cells, which are unmapped by-name, but have useful by-ontology mappings.

There are also some benefits to the name based mapping. A quirk of the ontology-based mapping is that many cell types are identified as having being part of the *immune system*, however, this isn't a well-defined locality. It's possible for samples from different physical locations (e.g. liver, blood) to map to the *immune system* term. For the FANTOM5 data at least, the names better describe locations that these samples came from.

6.5.1.4. Combining mappings

To get the best of both mappings, we need to combine them using the `Uberon.get_overall_tissue_mappings` function. This function creates both an overall mapping and a list of disagreements. Where only one mapping covers a term, it is trivial to do this (the overall mapping uses the present mapping, and there are no disagreements). When both mappings are present and one term is an ancestor of another, we say there is no disagreement and choose the more specific mapping e.g. if mapping by ontology gives us *photoreceptor array*, but mapping by name gives us *eye*, then because *photoreceptor array* is `part_of` *eye* and *eye* `is_a` *sense organ*, we would use the overall mapping by ontology since *photoreceptor array* is the more specific term. When both mappings are present and there is no relationship between them, this is when we say there is a disagreement, and we can choose which mapping we give precedence to, by default it is the ontology mapping.

Before we create the overall mapping, we will remove the *immune system* by-ontology mappings, for the reasons discussed above.

The FANTOM5 data contains different categories of samples including tissues, time courses, immortal cell lines, fractionations and purturbations, and primary cells. Some of these categories might not map in the way that we might want them to because although they might be a cell type that is usually localised to a tissue, they are unusual since they represent unusual in-between developing tissues (e.g. stem cells) or cancerous immortal cell lines. This is likely to have led to uncertainties in the sample ontology file, so by restricting to primary cell and tissue samples, we might get a more accurate picture of the percentage of mappable samples that Ontolopy can reach.

Mapping name	Number (and percentage) of primary cell and tissue mapped samples	Number (and percentage) of primary cell and tissue unmapped samples
By ontology: using Uberon tissues only	**220 (29.57%)**	**524 (70.43%)**
By ontology: using Uberon tissues and CL cells	**700 (94.09%)**	**44 (5.91%)**
By name: using tissue column	**652 (87.63%)**	**92 (12.37%)**
Overall mapping (combining by-ontology with CL, and by-name mappings)	**739 (99.33%)**	**5 (0.67%)**

Table 6.2 Table showing the difference in coverage for different mappings, when restricting the samples to tissue and primary cell samples.

Table 6.2 shows that the missing mapping seen in Table 6.1 can be explained by the presence of sample types such as developing tissues and immortal cell lines (models for diseases), i.e. not healthy adult tissues. Again there was a benefit in combining mappings. The only remaining unmapped tissues were:

1. *unclassifiable* reference RNA samples (from different providers) - shown in Fig. 6.8, from mixed donors and cell types: this is reassuring as we would hope that these would not be mapped to a tissue.
2. Two *Buffy coat* sample of *reticulocytes*. We could map these by hand to the UBERON "blood" term.

sample_id	Charateristics [description]	Characteristics [Sex]	Characteristics [Age]	Characteristics[Tissue]
FF:10000-101A1	Clontech Human Universal Reference Total RNA, ...	mixed	NaN	unclassifiable
FF:10002-101A5	SABiosciences XpressRef Human Universal Total ...	mixed	NaN	unclassifiable
FF:10007-101B4	Universal RNA - Human Normal Tissues Biochain,...	mixed	NaN	unclassifiable
FF:11931-125I5	CD34 cells differentiated to erythrocyte linea...	NaN	NaN	Buffy coat
FF:11932-125I6	CD34 cells differentiated to erythrocyte linea...	NaN	NaN	Buffy coat

Fig. 6.8 Table showing the remaining samples which could not be automatically mapped to an Uberon tissue using Ontolopy. All three are reference RNA samples.

6.5.1.5. Finding inconsistencies

By comparing the results of both ontology and text based searches, Ontolopy can find inconsistencies between the two representations which sign post to issues with samples data and how it is presented, or in the ontologies that it is linked to (in this case Uberon and CL): I give an example of each type. I found this approach very useful, as it allowed me to feed back my discoveries to the maintainers of these ontologies and datasets in order to improve them, and has resulted in improvements to several of these resources.

There were two main ways in which inconsistencies were found:

1. Through looking at samples which are not mapped by one method or another.
2. By looking at the disagreements output which compares the mapping that Ontolopy finds using one file and method (text in sample information file), to that it finds using the other (terms in sample ontology file).

For the FANTOM5 data, disagreements between these mappings revealed problems in the biological ontologies and experiment metadata that were provided to the package in order to create the mappings. These discrepancies may be a lack of specificity, incompleteness in, or disagreement between FANTOM, CL, or Uberon annotations, either in creating ontologies or annotating tissues to samples. The process of mapping FANTOM to Uberon tissues found twenty-two such disagreements, of which FANTOM, Uberon, and CL where appropriate have been informed via GitHub issues, some of which have already sparked changes in the ontologies.

Four different types of example are described below, to give an idea of how multiple mappings may be used to improve annotation.

A full list of disagreements can be seen in Fig. 6.9. There were **32** disagreements/inconsistencies found using Ontolopy. These disagreements can affect multiple (replicate) samples, for a total of **96** samples.

sample_id	description	by name text	by ont text
FF:10277-104E7	optic nerve, donor1	neuron projection bundle connecting eye with b...	cranial nerve II
FF:11207-116A1	Endothelial Cells - Aortic, donor0	aorta	artery
FF:11216-116B1	Urothelial cells, donor0	urinary bladder	urothelium
FF:11219-116B4	Mesenchymal Stem Cells - Vertebral, donor1	spinal cord	vertebra
FF:11220-116B5	Sebocyte, donor1	zone of skin	skin sebaceous gland
FF:11234-116D1	Smooth Muscle Cells - Brain Vascular, donor1	brain	vasculature
FF:11242-116D9	Ciliary Epithelial Cells, donor1	camera-type eye	epithelium
FF:11248-116E6	Anulus Pulposus Cell, donor1	spinal cord	annulus fibrosus disci intervertebralis
FF:11252-116F1	Nucleus Pulposus Cell, donor1	spinal cord	nucleus pulposus
FF:11266-116G6	Endothelial Cells - Thoracic, donor1	internal thoracic artery	thoracic aorta
FF:11269-116G9	Fibroblast - Dermal, donor1	zone of skin	dermis
FF:11271-116H2	Hair Follicle Dermal Papilla Cells, donor1	hair follicle	dermal papilla
FF:11272-116H3	Keratinocyte - epidermal, donor1	zone of skin	skin epidermis
FF:11273-116H4	Mammary Epithelial Cell, donor1	breast	mammary gland
FF:11279-116I1	Preadipocyte - subcutaneous, donor1	adipose tissue	hypodermis
FF:11280-116I2	Preadipocyte - visceral, donor1	heart	connective tissue
FF:11291-117A4	Synoviocyte, donor1	synovial membrane of synovial tendon sheath	synovial membrane of synovial joint
FF:11393-118C7	Endothelial Cells - Lymphatic, donor3	capillary	lymphatic vessel
FF:11453-119A4	Bronchial Epithelial Cell, donor4	lung	bronchus
FF:11469-119C2	Preadipocyte - perirenal, donor1	kidney	perirenal fat
FF:11493-119E8	Meningeal Cells, donor1	meningeal cluster	blood-cerebrospinal fluid barrier
FF:11499-119F5	Perineurial Cells, donor1	spinal cord	perineurium
FF:11513-119H1	Smooth Muscle Cells - Tracheal, donor1	lung	trachea
FF:11518-119H6	Renal Mesangial Cells, donor1	kidney	connective tissue
FF:11535-120A5	Fibroblast - Villous Mesenchymal, donor1	trophoblast	placenta
FF:11590-120G6	Alveolar Epithelial Cells, donor2	lung	renal glomerulus
FF:11752-123G6	mesenchymal precursor cell - cardiac, donor1	heart	mesenchyme
FF:11758-123H3	mesenchymal precursor cell - ovarian cancer me...	ovary	connective tissue
FF:11842-124H6	mesenchymal precursor cell - ovarian cancer ri...	bone marrow	right ovary
FF:11933-125I7	Olfactory epithelial cells, donor1	anatomical system	epithelium
FF:12226-129F3	nasal epithelial cells, donor1	nasal cavity	epithelium
FF:12238-129G6	chorionic membrane cells, donor1	chorion membrane	egg chorion

Fig. 6.9 Table showing all types of disagreements found using Ontolopy, with example samples.

6.5.1.5.1. Finding samples that are missing annotations to tissues

When we look at the samples that we would expect to map by ontology, but that don't, after filtering for tissues and primary cells only, we see that there are just two types of samples:

1. One sample `FF:10379-105H1` which is missing `is_a: FF:0010164 ! human caudate nucleus - adult donor sample` in the FANTOM5 ontology file
2. 21 T-cell samples, all of which appear not to have been fully classified (i.e. contain the following line in the ontology file `comment: Changed from previous label. TODO: full classification`). I could map all of them to the term for *T-cell*, whereas someone with more knowledge of T-cells could more accurately map these samples to more specific cell types.

I can use Ontolopy to add these mappings to the merged ontology to improve the by ontology mapping if I needed to: this would help me to find additional mappings, for example, to *immune system* as well as *blood*.

6.5.1.5.2. Missing Uberon or CL annotation

Example: Missing annotation `Bronchus part_of some Lung`

One type of problem that can be revealed is a missing link in an ontology.

An example of this that was found using the FANTOM data set was that there was no formal relation in the Uberon ontology between *Bronchus* and *Lung*, despite the fact that the description text for *Bronchus* says "the upper conducting airways of the lung".

This was found because the sample `FF:11511-119G8` (*Bronchial Epithelial Cell, donor1*) is mapped by name to `UBERON:0002048` *Lung*, but by ontology to `UBERON:0002185` *Bronchus*. This was flagged as inconsistent because there are no relations in the Uberon ontology between these terms.

Similar missing annotations were discovered between *Aorta* and *Artery*, *Hair follicle* and *Dermal papilla*, and *Skeletal muscle myoblast* and *Skeletal muscle fiber*, and *Trophoblast* and *Placenta*.

6.5.1.5.3. Mislabelled sample

Sometimes samples are simply mislabelled, this can happen in any file type.

Example: `FF:11590-120G6` should be labelled *Alveolar Epithelial Cells* not *Renal Glomerular Endothelial Cells*

The FANTOM sample ontology file contains two samples named Renal Glomerular Endothelial Cells, donor2: `FF:11590-120G6` and `FF:11594-120H1`. One of these is a mislabelled sample, and it is actually an Alveolar Epithelial Cell sample. The mistake is only for the name in the FANTOM ontology file, but not the tissue annotation.

Example: `FF:11842-124H6` should be labelled *ovary* not *bone marrow* in the samples information file The tissue column of the samples information file lists sample `FF:11842-124H6` as a bone marrow sample, despite being an ovarian cancer sample.

6.5.1.5.4. Imprecise annotation to tissue

Example: *Nucleus pulpopus* as *Spinal cord*

Several FANTOM5 tissues are labelled by name colloquially, rather than precisely. For example, both *Nucleus pulpopus* and *Vertebra* are labelled *Spinal cord* (although the spinal cord itself is considered disjoint from these entities by definition, and in the Uberon ontology). It's for this reason that the ontology mapping is preferred over the labelled sample name in creating the overall FANTOM sample-to-tissue mapping.

Example2: `FF:11423-118G1 is_a` *dermal melanocyte*

Sometimes the text in the samples information file can help us to reach better mappings in the sample ontology file. For example sample `FF:11423-118G1` (and five other similar samples) are mapped to `CL:0000148` (*melanocyte*), which is a cell that can come from many different parts of the body (skin, heart, eyes, etc), so Ontolopy can only map this term to several tissues (some of which this cell will not have come from) and only if the child mapping functionality is used. However, since the sample was labelled as coming from the "skin", it's clear that this sample would have been better annotated to `CL:0002482` (*dermal melanocyte*).

6.5.1.6. Mapping overview

Using Ontolopy we can get a coverage of all samples that we would expect to map to a localised tissue (defining this as primary cell and tissue samples excluding reference RNA). These mappings correspond to **157** unique tissues.

6.5.2. Creating tissue-to-phenotype mappings

The approach to the creation of the tissue-to-phenotype mappings is different to that we just took for sample-to-tissue mappings in that we are only doing a by-ontology mapping, rather than also mapping by-name and then comparing. However, it is also a more complex example of a by-ontology mapping since we are asking more than one question to the ontology and adding them together. For all these questions, we start with the **157** tissues that we are interested in finding mappings for as source terms, and we use `opy.Relations`'s `mode='all'` option to find *all* of the Gene Ontology `targets=['GO']` terms that are related to them.

> **by-ontology mapping**
>
> Here we are only using an ontology based mapping, but if we had information in the samples information file about phenotype (e.g. disease), we could also use this to do an additional name based mapping if we wanted to.

We are interested broadly in tissues where a phenotype can take place, so this could be something on the level of proteins (*calcium signalling*), cells (*cell motility*), or tissue (*protein secretion*). This will affect what settings (particularly `allowed_relations`) we use when we make calls to `Relations`.

Only Gene Ontology *Biological Process* terms are related to phenotypes. The quickest way to retrieve only these is to ask for all `GO` terms and then filter them afterwards. After loading the GO basic ontology, we can easily retrieve a list of *Biological Process* terms.

6.5.2.1. Propagating relationships up the tree using `part_of`

Our first example of looking for relations between tissues and phenotypes will include the `part_of` relation. Since ontologies are often represented by DAGs, relationships are usually generally in one direction. While there is also the `has_part` relationship that we will look at shortly, `part_of` is preferred in the Uberon ontology with almost 10 times as many instances (15,486 compared to 1,703).

We first combine the GO ontology with the uberon ontology, which will simply help us to be able to look up the names of the GO terms to present the output in a more accessible format. This doesn't make a difference to the number of mappings, only to the `relation_text` field of the output (which will contain names instead of GO term IDs if available).

We then use the `opy.Relations` class with `mode='all'`, and `allowed_relations` including `is_a`, `part_of` (as mentioned), and some relationships which typically define relationships between tissues and phenotypes `is_model_for`, `capable_of`, `capable_of_part_of`, and the `GO` relation which is defined by Ontology to capture references to `GO` terms within definitions. This retrieves the mappings in **0.07 seconds**.

The `Relations` class returns a dataframe with the same format whether you use the default `mode` (finding `any` mapping that looks like the `targets`) or the `all` mode; both are indexed by source terms. For `all` mode, however, there can be multiple mappings for each source term, so the dataframe contains lists of mappings. This dataframe isn't too easy on the eyes (or analysis), so Ontolopy also has a helpful method called `format_all` which reformats the `Relations` output dataframe when the `all` mode is used into an easier-to-work-with multi-indexed dataframe. Example output of this can be seen in Fig. 6.10.

		relation_text
Tissue	**Phenotype**	
UBERON:0001255	**GO:0048731**	urinary bladder part of lower urinary tract part of renal system GO renal system development is a system development
	GO:0007275	urinary bladder part of lower urinary tract part of renal system GO renal system development is a system development part of multicellular organism development
	GO:0048856	urinary bladder part of lower urinary tract part of renal system GO renal system development is a system development is a anatomical structure development
	GO:0032502	urinary bladder part of lower urinary tract part of renal system GO renal system development is a system development is a anatomical structure development is a developmental process
	GO:0008015	urinary bladder part of lower urinary tract part of renal system GO renal system process involved in regulation of blood volume is a renal system process involved in regulation of systemic arterial blood pressure part of regulation of systemic arterial blood pressure is a regulation of blood pressure part of blood circulation
UBERON:0000955	**GO:0050890**	brain capable of cognition
	GO:0048856	brain is a organ part of anatomical system GO system development is a anatomical structure development
	GO:0007275	brain is a organ part of anatomical system GO system development part of multicellular organism development
	GO:0021551	brain part of central nervous system GO central nervous system morphogenesis

Fig. 6.10 Table showing a view of the output of `Relations.format_all` for the tissue-to-phenotype mapping.

As Fig. 6.10 shows, mappings contain a mixture of mappings to specific GO terms like *brain* and *cognition*, and very general phenotype terms, like *urinary bladder* and *anatomical structure development*.

As we've seen in other example use cases, it's possible to use the `exclude` option when retreiving the mapping, to exclude any terms that you might wish to avoid, for example very general terms if you have a list of these. Since we didn't know this, we found the 20 most frequently mapped GO terms (seen in Fig. 6.11), out of **260** mapped overall. From this list **10** tissues to remove were then manually identified: **developmental process, biological_process, anatomical structure development, multicellular organismal process, multicellular organism development, system development, system process, single-organism process, single-organism developmental process, and cellular process**.

	Frequency	**name**
GO:0032502	142	developmental process
GO:0008150	142	biological_process
GO:0048856	141	anatomical structure development
GO:0032501	141	multicellular organismal process
GO:0007275	141	multicellular organism development
GO:0048731	140	system development
GO:0003008	108	system process
GO:0044699	64	single-organism process
GO:0044767	63	single-organism developmental process
GO:0050877	49	nervous system process
GO:0009653	46	anatomical structure morphogenesis
GO:0007399	35	nervous system development
GO:0048513	35	animal organ development
GO:0007417	34	central nervous system development
GO:0021551	34	central nervous system morphogenesis
GO:0050890	30	cognition
GO:0003013	25	circulatory system process
GO:0007586	24	digestion
GO:0022600	24	digestive system process
GO:0008015	23	blood circulation

Fig. 6.11 Table showing the frequency that phenotype (`GO`) terms are mapped to the provided tissue terms, for the top 20 GO terms.

Fig. 6.13 (b) shows us that after removing very general terms, the majority of terms have 1-20 phenotypes mapped to them.

A small number have more, and a small number have no mappings. There are **24** terms in (b) which do not have a mapping except for the very general terms. These terms are: **adipose tissue, pancreas, blood, umbilical cord, throat, zone of skin, breast, skin of palm of manus, cerebrospinal fluid, anatomical system, epithelium, pelvic region of trunk, skin of body, retroperitoneal space, connective tissue, thoracic segment of trunk, neck, mediastinum, omentum, perirenal fat, amnion, chorion membrane, insect adult prothoracic segment, and insect adult mesothoracic segment**. Clearly there are phenotypes that affect these tissues (with the exception of the obsolete term), so the lack of mapping here may represent missing relationships or terms within the gene ontology. An important one for our data set is *blood* (since we have many such tissue samples): there are GO phenotype terms relating to *blood* such as *blood circulation* and *blood coagulation*, so why don't we get mappings to these terms?

6.5.2.2. Propagating "down" the tree: `has_part`

The problem above happens because the annotation to these phenotype terms is not carried out at the level of tissue (*blood*) but at the level of cell type (*blood cell*). In order to retrieve these terms, instead of propagating up the tree (to more general terms) we need to look down the tree (to more specific terms).

One way is to use `Relations` including relations in `allowed_relations` that denote having something as a part: `has_part` and `composed_primarily_of`. From here onwards, I'll use `has_part` as a shorthand for both of these terms. It doesn't make sense to run `Relations` with both `has_part` and `part_of`, since by running both up and down the tree, it could lead to technically true, but uninteresting and potentially misleading mappings. A simple fictional example of this would be mapping *little toe* and *big toe nail* by finding the relation *little toe* ***part of*** *toes* ***has part*** *big toe* ***has part*** *big toe nail*.

Including phenotypes that are only relevant for part of the tissue makes sense for tissue samples like *blood* where we have all parts of the blood in our sample (e.g. the sample will certainly contain *blood cell*s and *plasma*). However, they may make less sense for a tissue sample like *heart*, where we don't know if the sample came from the *right ventricle* or the *left ventricle* and there may be phenotype terms which are specific to a part of the anatomy we didn't sample from. With this in mind, our choices are:

1. don't include `has_part` relations, and miss phenotype mappings that are made at the level of constituent parts
2. include `has_part` relations, but be aware that some samples may map to phenotypes that they are not capable of if the sample-to-tissue mapping is not specific enough.

In our case, option (2) is preferable, particularly because the FANTOM5 dataset contains many blood samples, and otherwise we would be missing phenotype mappings entirely for these samples. Running this is otherwise very similar.

has_part and part_of

Ontological relations have strict definitions which help allow us to reason based on these relations, for example by `A has_part B`, we mean that A always has B as a part, while `B part_of A` means that whenever B exists it is part of A.

This means that `has_part` and `part_of` are not inverses[187], i.e. if `A part_of B`, that does not necessarily mean that `B has_part A`. For example, wherever human ovaries exist, they are `part_of` humans, but whever humans exist, they don't necessarily `has_part` human ovaries.

There are also other terms which denote similar relations, such as `composed_primarily_of`.

In Table 6.3, which compares the number of mapped tissues and phenotypes for different tissue-to-phenotype mapping methods, we can see that this method maps more phenotypes, but for less tissues.

Although less tissues have been mapped overall, we can tell they do capture previously unmapped tissues since the overall number of unmapped tissues (after removal of too-general terms) reduces from **24** with propagating up only to **10** which aren't mapped by either method. While propagating down therefore improves the overall mapping coverage, looking at the tissues which remain unmapped gives us a clue as to what further improvements we can make.

The terms which remain unmapped are **connective tissue, epithelium, adipose tissue, cerebrospinal fluid, mediastinum, umbilical cord, perirenal fat, retroperitoneal space, omentum, and anatomical system**. The give-away term is *skin of body*, since looking at subfigure `(d)` in Fig. 6.13, by hovering over the top most well-mapped tissue we can see that it is *stratum basale of epidermis*, which is part of the *epidermis* which is in turn part of the *skin of body*. The reason skin of body doesn't have a mapping is because while the *epidermis* is `part_of` the *skin of body*, the *skin of body* does not have the `has_part` relation to *epidermis*.

6.5.2.3. Propagating down the tree: inverse of `part_of`

In addition to the `has_part` approach, we could use the inverse of the `part_of` relations, however the definitions of these terms mean that the inverse of `part_of` means something like *can have part*. This would mean that in addition to the risk of potentially including mapping to more specific parts of the body that weren't in our sample (as we discussed for the `has_part` approach), we might sometimes include mappings to tissues that were not even present in the species or gender from which the sample came. The species problem is the much more pressing concern since Uberon is a multi-species ontology containing many non-human-specific terms and therefore we could end up mapping human samples to terms like `GO:0035844` *cloaca development*.

One solution to this in Ontolopy is to define relations that look something like `A can_have_human_part B` from the information in the ontology files, by using the inverse of `part_of` relations only where there is an external reference (`xref`) to the FMA human anatomy ontology. We need to do this semi-manually as Ontolopy does not currently contain tools for automatically defining new relations. Once we've created this new relationship, we can ask for relations including it in the same query as `has_part`. One downside of this approach is that relations found using this kind of self-defined relation will not be able to use the simple reasoning that Ontolopy is capable of (i.e. collapsing relations by using definitions like `is_a·part_of` == `part_of`, since such equivalences are not defined).

> **Sex-specific phenotype mappings**
>
> We could also create a relation like `A can_have_part_in_female B` (and an analagous term for male) when `B part_of A` and `B part_of UBERON:0003100` *female organism*. We could then cross-reference the sex of our samples from the sample information file to ensure that we don't create mappings between e.g. male-only samples and ovaries. This isn't done here, since it will only affect a very small number of mappings (given that many of the samples are mixed/unknown sexes, that there are only a small number of sexual dimorphic tissues, and that these are generally mapped at the level of specific sexes already), and wouldn't illustrate a different aspect of using Ontolopy. Such mappings, if they exist, are simply not included. If they are excluded, it means that we simply do not map non sex-specific tissues like *gonad* to either testes or ovary-related phenotypes, so we might be missing such mappings.

We could also do the same for sex-specific tissue-phenotype mappings, should we want to. This could be useful, depending on the experiment data in question and the resulting sample-tissue mappings we'd previously attained.

This resulting mapping contains **4968** additional tissue-phenotype mappings (not found in either the `has_part` or the `part_of` approach). Some examples of these additional tissue-phenotype mappings that were found using this `can_have_human_part` approach are given in Fig. 6.12. Overall, this mapping covers **133 (85.26%)** tissues and **449** phenotypes.

The yet unmapped tissues (by any method) are now: **connective tissue, epithelium, cerebrospinal fluid, adipose tissue, mediastinum, umbilical cord, perirenal fat, retroperitoneal space, and omentum**. While this list still contains tissues that we would expect to map to GOBP phenotypes, the lack of these terms in our searches now means that they are simply missing annotations. For example we have no mapping for *adipose tissue* despite the fact that GOBP terms exists for *adipose tissue development* and *fat cell proliferation*, but there is no cross-ontology mapping of these term in the current version of the ontology, so there is no way Ontolopy could pick them up.

Tissue	Phenotype	relation_text
UBERON:0002022	**GO:0048858**	insula is a cerebral hemisphere gray matter can have human part cerebral cortex can have human part hippocampal formation can have human part hippocampus alveus is a central nervous system white matter layer composed primarily of white matter can have human part gracile fasciculus GO gracilis tract morphogenesis is a central nervous system projection neuron axonogenesis is a central nervous system neuron axonogenesis is a axonogenesis is a neuron projection morphogenesis is a cell projection morphogenesis
UBERON:0016525	**GO:0019226**	frontal lobe can have human part anterior segment of paracentral lobule is a regional part of brain composed primarily of neural tissue has part neuron capable of transmission of nerve impulse
UBERON:0000473	**GO:2000147**	testis can have human part seminal vesicle can have human part duct of seminal vesicle channel for seminal vesicle fluid is a seminal fluid capable of part of positive regulation of flagellated sperm motility is a positive regulation of cilium-dependent cell motility is a positive regulation of cell motility
UBERON:0011595	**GO:0048870**	jaw region can have human part tooth bud can have human part odontogenic papilla is a developing mesenchymal condensation composed primarily of mesenchyme condensation cell is a mesenchymal cell is a motile cell capable of cell motility
	GO:0006897	jaw region is a organism subdivision has part external soft tissue zone has part musculature can have human part muscle tissue can have human part endomysium is a reticular tissue can have human part reticuloendothelial system composed primarily of phagocyte capable of phagocytosis is a endocytosis
UBERON:0001255	**GO:0060562**	urinary bladder is a viscus is a trunk region element is a organ can have human part vasculature of organ is a vasculature can have human part capillary bed is a epithelial plexus is a epithelial tube GO epithelial tube morphogenesis

Fig. 6.12 An example of six of the **4968** additional Uberon-GOBP mappings found using `can_have_human_part`.

6.5.2.4. Combining previous mappings

To create the final tissue-phenotype mapping, we combine the propagating up (`part_of`) mapping with the larger propagating down (`can_have_human_part`) mapping, by simply appending the new lines of the DataFrame.

> **'can_have_human_part' query completely contains 'has_part' query.**
>
> Because the `allowed_relations` for the mappings of the `can_have_human_part` query completely contain those for the `has_part` query, so do the found relations. This means that we only need to combine the `part_of` and `can_have_human_part` mappings to get the most complete set.

As we can see in Table 6.3, the overall combined mapping covers **147 (94.23%)** of the Uberon tissues searched for and maps to **510** unique GOBP tissues. Since this is greater than any other individual mapping, we can see that it is necessary to combine different mapping types to get high (>90%) coverage of tissues.

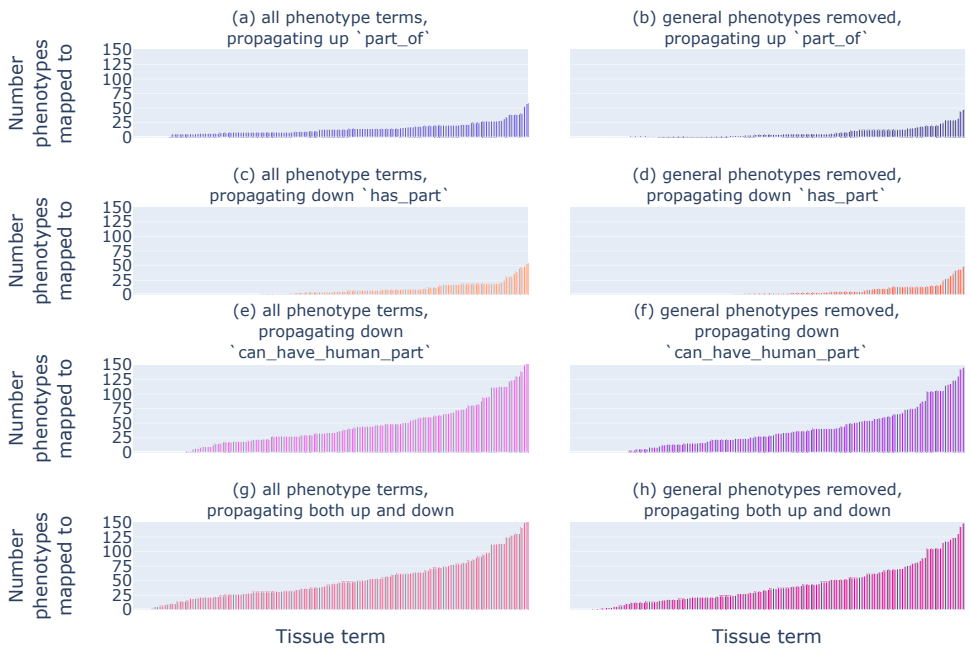


Fig. 6.13 Bar charts showing the number of tissue-phenotype mappings (number of phenotypes mapped to each tissue - roll mouse over to see tissue names) for (a) all phenotype terms propagating down, (b) general phenotypes removed propagating down, (c) all phenotype terms propagating up, (d) general phenotypes removed propagating down, (e) all phenotype terms propagating both up and down, and (f) general phenotypes removed propagating both up and down.

There are **9** unmapped tissues (which map to zero phenotypes), and all other tissues map to between 2 and 131 phenotype terms, (as we can see in figure Fig. 6.13). The number of mappings varies smoothly in this range with more general tissues and organs broadly appearing to have higher numbers of mappings than very specific tissues. We can also see in Fig. 6.13 that the `can_have_human_part` mapping makes up the majority of the mappings in the final combined mapping.

Mapping name	Tissue coverage: number (percent) tissues mapped (this mapping only)	Number of unique phenotype mapped to (by this mapping only)	Number tissues remaining unmapped (by this or any previous mapping)	Time to retrieve mapping
Propagating up	**132 (84.62%)**	**250**	**14**	**0.07 seconds**
Propagating down using `has_part`	**92 (58.97%)**	**248**	**10**	**0.07 seconds**
Propagating down using `can_have_human_part`	**133 (85.26%)**	**449**	**9**	**0.45 seconds**
Combined (all of the above)	**147 (94.23%)**	**510**	**N/A**	**N/A**

Table 6.3 Table comparing the difference in coverage (mappable tissues), phenotypes, time taken, and unmapped tissues for different mapping techniques.

6.5.3. Creating sample-to-tissue-phenotype mappings

Once we have both the sample-to-tissue and tissue-to-phenotype mappings, we can combine them to get the sample-to-tissue-phenotype mappings: mappings between samples and phenotypes that occur in the tissue type of that sample. There isn't a built-in Ontology function to do this, but since Ontology objects are built on top of Pandas DataFrames, they are fairly easy to work with.

Since we chose to map primary cell and tissue samples only, there are many samples which are not mapped. Null mappings are included in the output, and where mapping by name is used, it is recorded as a `mapped_by_name_to` relationship in the relation path, e.g. `FF:11453-119A4.mapped_by_name_to~UBERON:0002048.is_a~UBERON:0000171.capable_of~GO:0007585` or in text ***Bronchial Epithelial Cell, donor4*** *mapped by name to* ***lung*** *is a* ***respiration organ*** *capable of* ***respiratory gaseous exchange by respiratory system***. Since relation strings can now contain FANTOM5, CL, UBERON and GO terms, I first merge the GO ontology into the merged FANTOM5 and Uberon ontologies, so that the names of all terms can be found for the relation text.

6.5.3.1. Final mapping

There are **65953** rows of the sample-to-tissue mapping DataFrame in total. This includes some `NaN` values, so it contains **65608** mappings from sample to phenotype; equivalent to a sample coverage of **92.34%** of filtered (*tissue* and *primary cell*) samples or **81.00%** of all samples. It also includes an additional **181** mappings from sample to tissue (but not to phenotype), and **164** samples with no mapping to tissue or phenotype. Fig. 6.14 shows why almost 10% of samples are unmapped in more detail: many samples map to the same unmapped tissues, particularly adipose tissue, epithelium, or connective tissue.

Uberon ID	Uberon Name	Number samples mapped to tissue
NaN	Unmapped to tissue	164
UBERON:0001013	adipose tissue	72
UBERON:0002331	umbilical cord	8
UBERON:0001359	cerebrospinal fluid	1
UBERON:0000483	epithelium	62
UBERON:0003693	retroperitoneal space	2
UBERON:0002384	connective tissue	28
UBERON:0003728	mediastinum	1
UBERON:0003688	omentum	6
UBERON:0005406	perirenal fat	1

Fig. 6.14 Table showing how many samples are mapped to each unmappable tissue, showing why the coverage of samples isn't higher.

With Ontolopy, this complex task is relatively quick: it took **45 seconds** to run this whole notebook on a laptop without any parallelisation. Also recall that this section is merely an example of an application of Ontolopy: this same process could be done for other datasets that provide an ontology and/or a sample information file.

6.6. Discussion

This chapter is simply supposed to present Ontolopy as a usable tool for finding relationships in OBO ontologies, from which useful outputs can be obtained.

6.6.1. Usefulness

Ontolopy fills a need for quickly searching OBO files for relationships between ontology terms, and as the examples (both simple and complex) show, it fulfils this role well: it works quickly and finds the relationships that you would expect to find (when you know what to ask for).

By building on top of the extremely well-used data analysis tool of Pandas, when Ontolopy doesn't have a function written to do something (for example defining new relations, or combining mappings), users can fall back on the functionality of Pandas to create what they need with Ontolopy's outputs. By not (yet) integrating well with other OBO tools in development, however, it does miss potential impact.

Ontolopy is only as useful as the ontologies that it can query, so it has all of the limitations of those tools: they're missing some links because they are constantly being updated as our knowledge increases. At the same time, it is useful *because* it builds on these resources: these resources are created by biological curators with heaps of experience working with academic and medical communities. Ontolopy has already proved useful at least in providing a valuable way of feeding back into experimental data and ontologies. By checking for inconsistencies between multiple ways of labelling data, multiple issues in these resources and data sets have been identified and some of these revisions have been accepted.

6.6.2. Usability

One key feature of Ontolopy's usability is that it is well-documented. At the time of writing, it is much more so than other alternatives for working with OBO files. The documentation is versioned and contains well-worked examples and a descriptive API.

It is also quick and easy to install, lightweight, and has a small number of dependencies (the upside of the lack of integration with other tools).

Ontolopy runs quickly for a wide variety of tasks. As we saw in the examples, Ontolopy runs quickly for most uses involving operations on or queries to ontology objects (typically less than half a second). The time taken depends on the size of the ontology, the number of the chosen relations, and the popularity of those relations within the ontology. However, making a query with a large number of relations to check can inflate how long a query takes to run.

The name mapping (`Uberon.map_by_name`), however is the exception to this, which runs fairly slowly (on the order of seconds). There are more interesting text-mining techniques that could be integrated into Ontolopy if gains in speed where made here, for example using fuzzy-text matching to catch typos in sample information files (which are often present as they are often created by hand).

6.6.3. Limitations

Ontolopy is a small and lightweight package, so it hasn't got as much functionality as some larger tools, as well as having some limitations due to it's reliance on underlying ontologies.

6.6.3.1. You still need to understand the structure of the ontology

While Ontolopy makes it easy to query biological ontologies in Python, it doesn't prevent the user from needing to understand the structure of the ontology (what kind of relations it contains and what these mean) to be able to ask meaningful queries. Ontolopy will allow you to ask for nonsense relations, e.g. combining any relations which may give misleading responses if you are only looking at what it is mapping to and from (rather than the path that the mapping represents).

6.6.3.2. "Missing" functionality

There is plenty of non-existent functionality for Ontolopy that could be useful, namely:

- Text search functionality.

- Functions to facilitate more complex queries. I talk about both of these in Ontolopy future work. While I think including both of these pieces of functionality would be helpful to Ontolopy, it is completely beyond the scope of the package to re-implement SPARQL or semantic reasoners.

6.6.3.3. Improving choosing from multiple synonym options

The `Uberon.sample_map_by_name` function simply looks up the strings provided and looks for important external references to decide between synonyms. If this information is not provided or doesn't help us to make the choice, we currently just choose the first term that we found, ignoring information about synonym, or which synonym-having term is more specific.

6.7. Future Work

I keep an updated roadmap for Ontolopy on the documentation website and a detailed list of issues on GitHub.

6.7.1. v2.0.0

The main priority for Ontolopy is reaching a stable version for release, user-testing, and publishing the work.

You can see the full list of features for the v2.0.0 Milestone on GitHub, but to summarise:

- Finish tutorials for all functions/methods
- Reach 80% test coverage
- User-testing
- Benchmark speed against OWL reasoning

Classes of synonyms

There are different classes of synonyms defined for synonyms. `EXACT` synonyms mean that the meaning of the synonym term would be identical to the term's name, for example *mononuclear cell* has the exact synonym *mononuclear leukocyte*. On the other hand, `NARROW` synonyms have more specific definitions than the name itself (in some cases they might eventually become a subclass of the original term), for example *mononuclear cell* has the narrow synonym *peripheral blood mononuclear cell*. There are also *BROAD* and *RELATED* synonyms.

6.7.2. Other potential improvements to Ontolopy

In addition to the barebones necessities for v2.0.0 above, there are a number of more ambitious pieces of functionality which would improve its' usefulness.

6.7.2.1. Text-search and fuzzy-matching

The `Uberon.map_by_name` function currently implements a slow search of names in ontologies, but this search could be both much quicker and more general (e.g. able to use wild cards, or search in any field to create mappings), i.e. similar in fuctionality to OwlReady2[248]'s **`search`** function. Fuzzy-matching then expand this functionality to help capture misspelled information.

6.7.2.2. Functionality for more complex queries

Ontolopy doesn't contain tools for making complex queries. For example, if we want to find out which samples are made of precursor cells, we have to find *in vivo* samples which are or are derived from stem cell samples. In this particular case, the difficulty is partly because `derives_from` means "extracted from", or "extracted from and then had lots of things done to it", which can change the meaning.

If we want to do this at the moment, we have to make two queries to Ontolopy separately and then combine them (as in this example), which is clearly not very user-friendly.

6.7.2.3. `opy.Go`

The `opy.Uberon` class adds functionality specifically for the Uberon ontology, helping users to map between samples (or other entities) and tissues for their specific area of interest. We could imagine similar functionality for the Gene Ontology, and perhaps integration with GOATools[236].

6.7.2.4. Integration with Pronto

Test integration with Pronto to investigate how it would effect how the speed of the current implementation.

6.7.2.5. Ontology validity

Since it is possible with Ontolopy to add new terms and relationships, to merge OBO files, and to build ontologies from scratch, it might be useful for Ontolopy to have some functionality for checking the validity of the OBO object automatically. This could include for example checking for cyclic relationships, or checking if there is any missing information such as ontology version numbers, or required attributes for terms. This could use some of the same approaches as ROBOT.

6.7.3. Miscellaneous

- **NLP similarity measures**: An earlier version of Ontolopy included text-mining to find similar terms using a TF-IDF like measure to create a similarity measure based on co-occurance, e.g. "mental" and "brain" have high similarity since they often appear in the same document (documents being term descriptions). This functionality is not currently in Ontolopy, and doesn't really align with the core functionality of the module, but it could be released separately to find potential missing ontology links.
- **Versioned docs sphinx extension:** Although it is not really an output of Ontolopy, I also hope to be able to release the sphinx extension that I used to create versioned documentation soon.

7. Combining RNA-seq datasets

In this chapter, I present work done in combining four tissue-specific gene expression data sets into one harmonised data set. The resulting data set has been documented and made available to the research community - and will be used to further improve Filip.

This chapter includes the methodology for choosing suitable data sets and combining meta-data into a harmonised form using Ontolopy, and explains why this is useful.

> 💡 **Contributions to research outputs in this chapter**
>
> The contributions in this chapter include:
>
> - Creation of publicly available combined gene expression meta-data.

7.1. Introduction

7.1.1. Motivation

There is general agreement that integrating omics datasets is one of the primary challenges to overcome if we wish to harness the full information contained within them[249]. Falling costs and rapid advances in sequencing technologies have resulted in what many have described as a deluge of omics data[250]. And this includes huge amount of gene expression data, as demonstrated by the 3,564 studies and 112,225 assays currently available through the European Molecular Biology Laboratory's (EMBL) Gene Expression Atlas (GxA) website [69].

Each individual measure of expression is only a snapshot of what a gene can do. It only tells us about the transcription of proteins at that one time, in that one sample. Gene expression can also vary by tissue and cell type, individual organism[251], age[252,253], sex[254,255], time of day[251,256], and spatial location within a tissue or culture[257]. There are also interactions between these different sources of variation, for example certain genes may only exhibit differential expression based on time of day in certain cell types. If we want a full understanding of what a gene does, we must understand how it's expressed in a variety of scenarios, for example, in different tissues, from different people, at different times of day, and across many repeats.

My primary motivation in the creation of this data set was to aid in protein function prediction, since as we saw in chapter 4, when we use structural information to predict function, we don't have enough information about ways in which that structure might be prevented from functioning in certain tissues. Many phenotypes are naturally linked to certain locations. Humans have many diseases and features which are particular to certain locations on the body and certain tissues, be it blood, brain, skin, or lung. So to answer the question of whether a protein is produced in a context relevant to a phenotype, we not only need gene expression information, but *tissue-specific* gene expression information: is this gene ever expressed in the heart?

For our need in mapping tissue-specific gene expression data to phenotypes, preliminary work showed that one gene expression experiment was not sufficient: it would not give us enough coverage of phenotypes to validate an improvement in protein function prediction in CAFA.

For a typical (human) next-generation sequencing transcriptomics experiment, data is collected for over 20,000 genes, but generally far fewer samples, and very few replicates of a certain kind of sample (e.g. tissue). For context, the largest experiment in the Ensembl Gene Expression Atlas (GxA) by quite a margin is currently the Genotype-Tissue Expression (GTEx) Project with 18736 samples. This is simply because it is still too expensive for one experiment to measure enough samples to give us a comprehensive understanding of how genes can behave. So, when it comes to gene expression, we have "big data" in the sense that the data is large and we need to take care to access and compute on it efficiently (as we are measuring so many genes), but in any one experiment we don't have great coverage of sample types.

Combining expression data from multiple different experiments is perhaps the obvious tonic to this problem, since it has the potential to create a data set containing a more representative view of gene expression. However, it is not as straightforward as loading in multiple data sets. There are specific challenges relating to data management, statistics, and harmonisation of meta-data for interoperability. However it is possible, and has already been done for two experiments[258].

Although my aim in creating this data set was specifically to improve phenotype and protein function prediction, a larger gene expression data set has further uses outside of this. Additional repeated measurements, and a larger spread of samples would allow researchers to ask more questions and have a larger statistical power. For example, one possible use is identification of housekeeping genes, or building models of gene regulatory networks.

7.1.2. Challenges in combining gene expression data sets

Challenges in combining gene expression data arise from the myriad of possible differences in experimental and analysis protocols between gene expression experiments.

7.1.2.1. Harmonising meta-data

An important feature of any gene expression data set is the quality of the meta-data, by which I mean everything except the measures of gene expression, particularly including additional information about samples and protocols. For example, data about samples can be recorded at different levels of specificity. This was a particular challenge for tissue type labels where some samples are simply labelled *brain*, while others are labelled *medulla oblongata*, and yet others are identified by cell type.

Harmonising this cell and tissue meta-data was the challenges of combining the data sets, which was done using the Uberon cross-species anatomy ontology[109], and the Cell Ontology[110] (CL), which is integrated with Uberon. Samples were primarily assigned Uberon term identifiers by searching for matching text between sample information files and CL or Uberon term names or descriptions. Where existing terms did not turn up a match, samples were assigned an Uberon term by hand. Then using the Uberon ontology, tissues could be understood in relation to each other, being mapped to tissues and more general tissue groups.

7.1.2.2. Batch effects

Combining gene expression data itself, is also not trivial: a major problem is their well known susceptibility to batch effects (differences in measurements due to technical artefacts of sequencing batch)[259]. When combining and comparing gene expression data from two (or more) experiments, it's not obvious how much of our signal comes from real biological differences in transcription, and how much comes from unwanted variation associated with the batch it was sequenced in. These "batch effects" result from unknown variation during the process of sequencing for example the date, time, or location of sequencing[260], or the technician doing the work.

To complicate matters, some batch effects may be due to factors that might be expected to genuinely influence expression of genes, such as temperature, time of year, humidity, diet, individual, age, etc. Covariates such as these are often unrecorded and/or not reported, so it is not easy to distinguish these from those due to protocol differences, such as reagents, personnel doing the sequencing, hardware, processing pipeline, etc. For this reason, the problem of batch effects is closely related to the problem of recording sample metadata.

Batch effects can often confound and obscure the biological differences of interest between samples (e.g. tumour versus healthy tissue). At best, batch effects add random variation to expression measurements, which obscure signals. Often they can also add systematic differences that can lead to incorrect biological conclusions[259]. They are a problem for analysing the output of an individual experiment where there are multiple sequencing batches, but pose a particular problem in combining data from different experiments, as there is almost certainly more variations between analysis pipelines.

Batch-effect correction:

Batch effects may affect only specific subsets of genes, and may affect different genes in different ways[259]. This means that normalisation (e.g. TPM, FKPM) will not account for batch. However, when it is known, date of sequence processing is often used as a surrogate for batch, enabling researchers to check for, and then remove, batch effects if necessary.

There are a number of batch correction analyses which attempt to remove batch effects from RNA-seq data, for example ComBat[261] and Surrogate Variable Analysis (SVA)[262] Batch correction can be very useful for understanding baseline gene expression, but can lead to inflated p-values for downstream analysis (notably for differential gene expression, using ComBat[261]), where a more sensible approach is to include batch as a confounder for statistical tests.

ComBat:

ComBat[261] is a popular batch effect removal procedure, which was first developed for use with microarray data, but continues to be a popular choice for RNA-seq data. Generally, it is a well-trusted method for both of these types of gene expression data[263], although there is some evidence that it may "over-correct" batches for some RNA-seq data[264].

ComBat is an Empirical Bayes method, meaning that the prior distribution is estimated from the data. It is designed to "borrow/share information" between genes in order to get a better estimate of batch effects, and assumes that batch effects affect many genes in similar ways.

PCA to visualise batch effect removal:

Principal Components Analysis (PCA) is often used to visually inspect experimental results for batch effects; when biologically alike samples cluster together rather than those from like-batches, batch effects are often ignored. Cell type is one of the better understood influences on gene expression. We know that the same DNA is in every cell, and yet the morphology and function of each cell is determined by its cell type, due to its gene expression. We can expect largely similar patterns of gene expression in similar cell types, which means that when we know cell type of samples, this information can be used to aid in visually checking the results of batch correction using PCA.

7.2. Data Acquisition

This section describes the constituent data sets that were chosen for combination, and how they were chosen and acquired.

7.2.1. Criteria for choosing datasets

Datasets were chosen from the EBI's Gene Expression Atlas (GxA)[69]: the European Bioinformatics Institutes' Open Source gene and protein expression database, and the largest of its type. At the time of writing, it contains over 3,000 gene expression and protein abundance experiments across many organisms, organism parts (tissues), diseases, and sequencing technologies. There is a separate database for scRNA-seq experiments.

A major benefit of the GxA is that raw data using the same sequencing technology is re-analysed by GxA using the same data analysis pipeline (iRAP[265] for RNA-Seq). In addition to ensuring the quality of each data set included, and running it through the same pipeline, the GxA adds additional metadata for the experiments by using the literature to biologically and technically annotate each sample.

Data sets were chosen based on the following requirements.

1. **Inclusion in the GxA**.
2. **Experiments must be measuring baseline, rather than differential gene expression**.
3. **Samples must be sequenced using Next Generation Sequencing**, i.e. including RNA-Seq and CAGE, and excluding microarrays.
4. **Data sets must contain a breadth of tissues and genes**, i.e. experiments must include "organism part" as an experimental factor (otherwise tissue would not be recorded) and must have at least 80 assays (samples).
5. **Samples must not be disease-focused**. In practice, excluding cancer datasets was enough to exclude disease-focused datasets.

Choices (1)-(3) were made to ensure the data underwent the most similar possible analysis pipeline. Number (4) was necessary to aid batch correction by facilitating the most balanced data set design in terms of batch (experiment) to group (tissue), and in order to have good coverage of genes and tissues, which is necessary for downstream use. Choice number (5) was also made primarily to aid batch correction: since many phenotypes occur only in particular tissues, there is not a breadth of tissue measurements for most diseases.

As described in the introduction chapter, there are many ways to measure which proteins are being created. Here, I justify my choices of measures to include in the combined data set.

7.2.1.1. Gene expression vs protein abundance

As I mentioned earlier, gene expression is only weakly correlated to protein abundance. So, what are the potential reasons for this, and which makes a better measure of the process of translation?

One reason is that there is something preventing the mRNA from being translated, such as slow codons, the temperature, ribosome occupancy, or regulatory RNAs and proteins[266]. In these cases, the DNA is transcribed into mRNA, but the protein is never produced, meaning that using gene expression data as a measure of how much protein is produced would be overestimating the protein abundance. If these factors were a large contribution to the weak correlation, it could provide better results to use protein abundance data instead of mRNA abundance data to make predictions about how proteins are affecting human phenotypes. On the other hand, it could equally be possible that proteins are being produced, but not measured by protein abundance techniques. Protein half-lives range over orders of magnitude from seconds to days[266,267]. In this case, gene expression data may be a more reliable measure of protein production than protein abundance, since proteins may degrade before being measured. In yeast, protein degradation was shown to be the largest contribution to the protein-mRNA correlation compared to codon and amino acid usage (the two other factors estimated in the study), and more influential than those other two factors combined[268].

In summary, there's no perfect measure of translation, but since gene expression data is more readily available, and protein degradation appears to account for most of the differences between correlations, gene expression data presents the best proxy for translation for the downstream uses discussed here.

7.2.1.2. Gene expression vs Transcript expression

It's likely that transcript expression data would provide more insight than gene expression data if it were available, since it is likely that there are tissue-specific transcripts which do not correspond to tissue-specific genes, e.g. where different transcripts from the same gene are expressed in different tissues. Transcript expression data, however, is harder to come by and this approach relies on a wealth of available data. Furthermore, transcript expression data can be straightforwardly converted to gene expression data (by summing over the transcripts), while

the conversion of gene to transcript expression data is decidedly less accurate. When transcript-expression (CAGE) measurements are aggregated at the gene/protein level, measures of tissue-specificity have been found to largely (75-93%) match up with measures of tissue-specificity resulting from gene-expression measurements, as found in a comparison between the HPA and FANTOM5 experiments[269].

For these reasons, I have taken a gene-centric approach here. It may be important, however, to consider whether a gene has multiple transcripts in downstream analysis, for example, if including tissue-specific gene expression information when predicting the function of a protein-coding SNP (since it may not be in the relevant transcript).

7.2.1.2.1. Tissue-specificity versus cell-specificity

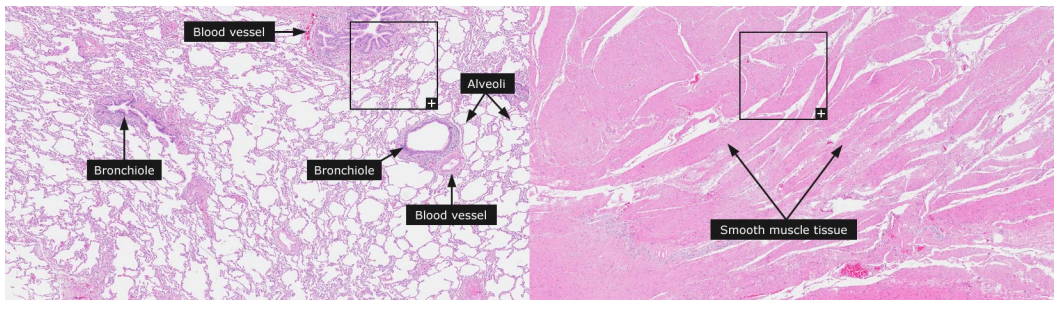


Fig. 7.1 Images of smooth muscle tissue from the stomach wall (left) and lung tissue (right), examples of homogeneous and heterogeneous tissue types respectively, taken from the Human Tissue Atlas website[270].

Tissues can be made up of various cell types. Some tissue types (e.g. smooth muscle) are quite homogeneous, comprising of predominantly one cell type and/or lacking structural features. Other tissues (e.g. lung) are heterogeneous, consisting of multiple cell types and features. Fig. 7.1 shows the different structure of two example tissue types. The bronchioles of the lung alone consist of six different cell types (basal cells, neuroendocrine cells, ciliated cells, serous cells, Clara cells and goblet cells), while smooth muscle tissue consists almost exclusively of tightly packed smooth muscle cells. The varying proportions of constituent cell types in heterogeneous tissues can influence tissue function.

While we may prefer to look at the gene expression of a cell type, we currently have much less scRNA-seq data than bulk RNA-seq data. Bulk RNA-seq also gives us the ability to measure the gene expression of tissues as they appear in humans. The average supply of a protein to a tissue (averaged over multiple cell types) may well influence a tissue's phenotype, in these circumstances gene expression at the level of a tissue may give us information that we can't retrieve from cell-line cells alone.

7.2.1.3. Inclusion of CAGE data

CAGE is transcript expression, rather than gene expression, and there are likely to be different transcripts measured by CAGE than by RNA-Seq. As mentioned above, however, it is possible to calculate gene expression from transcript expression. It's also possible to map between CAGE transcription start sites and existing transcript IDs that may be featured in RNA-Seq arrays. When this is done, it has been observed that the results of CAGE are comparable to those of RNA-seq[271], so the inclusion of CAGE data in a combined data set is reasonable.

7.2.1.4. Excluding disease-focused experiments

The decision to exclude disease-focused experiments was made primarily to reduce the complexity of the analysis and the resulting data set. The data set can now be interpreted as representing gene expression of healthy tissues. This was also a practical choice since most disease data sets (with the exception of cancer datasets) tended to have a narrow breadth of tissues, which would interfere with the batch correction methodology. For example, experiments interested in heart disease would naturally contain measurements of healthy and non-healthy heart tissues, and not other tissues, so would be difficult to combine with existing data sets due to the "missing" data. This would not have been a problem for cancer experiments, however cancer samples gene expression is known to be tissue-non-specific[76,215].

7.2.2. Method of searching

It would have been preferable to interrogate the GxA for datasets using the ExpressionAtlas R package, or the AtlasExpress API which it is built on. However, it was necessary to do so via the website since searches to the AtlasExpress API cannot differentiate between baseline and differential gene expression.

For reproducibility, this was done by downloading the json file used by the GxA experiment browser webservice and by choosing experiments with:

- baseline (rather than differential) expression measurements
- homo sapiens species
- RNA-Seq mRNA technology
- *organism part* as an experimental factor
- at least 80 assays.
- no mention of "cancer" in the description

Fig. 7.2 Our requirements for a data set reduce the number of eligible data sets to four.

As Fig. 7.2 shows, at the time of writing, there are over 3000 experiments in the GxA, and of these 27 are human baseline RNA-Seq experiments. Of these there are 4 which offer a good coverage of non-disease organism parts.

7.2.3. Eligible data sets

The four eligible data sets are summarised in Table 7.1, and they are described in further detail below.

shortName	experimentAccession	experimentDescription	numberOfAssays	experimentalFactors
HPA	E-MTAB-2836	RNA-seq of coding RNA from tissue samples of 122 human individuals representing 32 different tissues	200	organism part
FANTOM5	E-MTAB-3358	RNA-Seq CAGE (Cap Analysis of Gene Expression) analysis of human tissues in RIKEN FANTOM5 project	96	developmental stage, organism part
GTEx	E-MTAB-5214	RNA-seq from 53 human tissue samples from the Genotype-Tissue Expression (GTEx) Project	18736	organism part
HDBR	E-MTAB-4840	RNA-seq of coding RNA: Human Developmental Biology Resource (HDBR) expression resource of prenatal human brain development	613	developmental stage, organism part

Table 7.1 Table showing the GxA datasets that meet the criteria

7.2.3.1. FANTOM5

opl used the FANTOM5 data, as described in Section 5.3.1.

7.2.3.2. Human Protein Atlas

The Human Protein Atlas (HPA) project[272,273] aims to map all human proteins in cells (including subcellular locations), tissues and organs. The HPA project's data is not limited to the gene expression data that can be found in GxA, but that is the only part of the data that is used here. The gene expression data that was used (E-MTAB-2836 in GxA) excludes cell lines and includes tissue samples of 122 individuals and 32 different non-diseased tissue types.

7.2.3.3. Genotype Tissue Expression

The Genotype Tissue Expression (GTEx) project[274] was developed specifically for the purpose of studying tissue-specific gene expression in humans and gene expression data from over 18,000 samples, including 53 non-diseased tissue types and 550 individuals (ranging in age from 20s to 70s).

7.2.3.4. Human Developmental Biology Resource

The Human Developmental Biology Resource (HDBR) Expression data[275] is slightly different from the other data sets in that contains a much narrower range of sample types. All HDBR samples are human brain samples at different stages of development, ranging from 3 to 20 weeks after conception.

7.2.4. Data acquisition

Data was obtained, where possible via the *ExpressionAtlas* R package[276], which gives gene expression counts identified by ENSG IDs, metadata (containing pipeline, filtering, mapping and quantification information), and details of experimental design (containing for example organism part name, individual demographics, and replicate information, depending on the experiment).

For the FANTOM experiment, counts for transcript expression were downloaded directly from the FANTOM website.
The downloaded FANTOM5 file has already undergone some quality control by FANTOM, it is limited to peaks which meet a "robust" threshold (>10 read counts and 1TPM for at least one sample). The data acquisition code is not executed in this notebook as it is slow to download all the files, but the R script to do so can be downloaded **here**.

7.3. Data Wrangling

Before the data sets could be combined, substantial data wrangling was necessary. The details of these processes - obtaining, checking, mapping identifiers, and excluding irrelevant data - are described in this section. Ontology was developed and used to do much of this mapping, and parts of the wrangling mentioned here form examples in the Ontology chapter.

The steps required to obtain consistently formatted and labelled data can be described as follows:

- Obtaining the raw expression per gene for healthy human tissues
  - Data acquisition
  - (Where required) Mapping from transcript to gene
  - (Where required) Filtering out disease samples
  - (Where required) Filtering out non-human samples
- Mapping from sample name to UBERON tissue using Ontolopy.
- Mapping from UBERON tissues to tissue groups using Ontolopy.
- Aggregating metadata

7.3.1. Obtaining raw expression per gene for healthy human tissues

As mentioned in Data Acquisition, for the HPA, GTeX and HDBR experiments, count data were available through the *ExpressionAtlas* R package[276], and the FANTOM dataset was downloaded directly.

7.3.1.1. Mapping from transcript to gene

This step was only required for the FANTOM dataset.

FANTOM provides mappings to gene IDs based on proximity of genes to peaks according to Ensembl. Gene expression was then calculated by summing over transcripts mapped to genes. The transcripts were already mapped to HGNC gene identifiers in the downloaded FANTOM file and Ensembl's Biomart was used to obtain a **`mapping from HGNC gene identifiers to ENSG gene identifiers`**, in order to match the gene expression atlas format.

Any transcripts which mapped to multiple genes were discarded, as were any HGNC ids which did not map to ENSG ids.

7.3.1.2. Filtering out disease samples

The HDBR and HPA experiments contained only healthy samples.

GTEx Although GTEx contained clinical data, no disease-related phenotypes were removed from the data set, since the `disease` column contains only values of "normal" and the only clinical variables (as described in the `clinical_variables` column) in the dataset were sun exposure or lack thereof for skin tissues. I judged these to be within the normal range of environments that we would expect skin to be subjected to.

FANTOM The FANTOM sample ontology was used to remove samples which are models for diseases. Samples which are disease models are identified using the `is_model_for` relationship and these relationships are propagated to the children terms based on the `is_a` relationship. For example, `FF:11558-120D1` (Fibroblast - skin spinal muscular atrophy, donor2) would be removed from the set of samples, since: `FF:11558-120D1` (Fibroblast - skin spinal muscular atrophy, donor2) `is_a FF:0000251` (human fibroblast - skin spinal muscular atrophy sample) `is_model_for DOID:12377` (spinal muscular atrophy).

7.3.1.3. Filtering out non-human samples

The GTEx, HDBR, and HPA experiments contained only human samples.

FANTOM The FANTOM5 data set also contains non-human (mouse) samples. The FANTOM sample ontology (which was downloaded from here) was used to look-up which FANTOM samples are human samples, i.e. have an `is_a` relationship to the term `FF:0000210` (human sample) directly or indirectly.

7.3.2. Mapping to UBERON

Mapping from samples to Uberon tissue required the development of a small Python package Ontolopy. To create input to this package, informal tissue names (e.g. blood, kidney) were taken from the experimental design files (or the human sample information file for FANTOM) to create a map of samples to informal tissue names. For FANTOM, the FANTOM ontology could also be used to create a more fine-grained mapping of samples to tissues based on FANTOM sample identifiers and/or cell type (CL) identifiers.

HPA The HPA samples were mapped using exact matches to Uberon names. Three types of sample did not have exact matches: *transformed skin fibroblast*, *suprapubic skin*, and *ebv-transformed lymphocyte*. I manually mapped *suprapubic skin* to `UBERON:0001415` *Skin of pelvis*, and excluded the other two (corresponding to excluding 869 samples).

HDBR For HDBR, tissue names from the "organism part' column of the column data file were matched to Uberon names and synonyms from the Uberon extended ontology. The 96 unmatched terms corresponding to mixed brain tissues and brain fragments were defaulted to the more general Uberon Brain term.

FANTOM Since an experimental design file could not be obtained for FANTOM via GxA, additional sample information was obtained via the FANTOM5 website, namely the human sample information file and the FANTOM5 ontology.

FANTOM also contains time courses of cell differentiation (cells changing from one type to another) as well measures of perturbed cells. Since these samples do not have a well-defined locality in the body given by cell or tissue type, they were not used in the combined dataset. Such samples were filtered out using the human sample information file.

Since the FANTOM data had both an ontology file and the human sample information file, both were used to map to Uberon. The disagreements between the two mappings revealed some inconsistencies with the data set: these are described in the previous section, as they demonstrate a potential use case for Ontolopy.

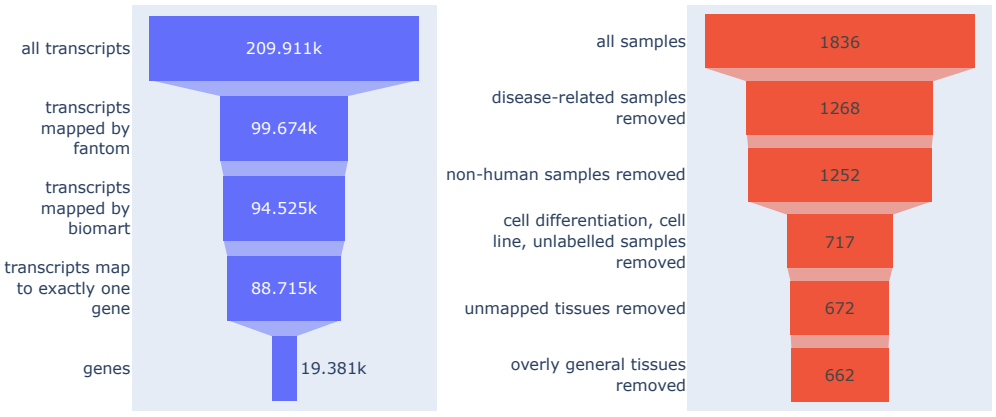


Fig. 7.3 Funnel plot showing the data cleaning pipeline for FANTOM transcripts/genes (left) and samples (right), along with the number which remained after each stage of data cleaning.

The amount of data that flows through the processing pipeline for the FANTOM5 dataset can be seen in Fig. 7.3.

7.3.3. Aggregating Metadata

To create consistent metadata for the samples (e.g. age, developmental stage, replicate status, etc), information was extracted from multiple sources (including GxA and additional data from each experiment), and sometimes manually curated or corrected.

HPA, HBDR, and GTEx: Metadata about the experiments was collected from multiple sources, primarily the column data files accessed via ExpressionAtlas. This metadata was used to describe the experimental design for ComBat. The metadata collected includes (where available), sample identifier, individual identifier, age (exact), age (range), developmental stage, tissue type (as Uberon term), sex, experiment, biological replicate identifier and technical replicate identifier.

Both age variables are given in years and may include negative values (e.g. for a developing fetus). The age (range) variable contains uneven ranges, since this allows there to be an age-related factor that is compatible across the experiments. These values had to be converted to common units, since they were incompatible between experiments, and age-related terms were missing in GxA for GTEx and HPA. For GTEx it was possible to acquire this information via its own website.

FANTOM: The metadata aggregation for the FANTOM dataset is described in detail in Section 6.5.

7.3.3.1. Tissue groups

While the data is in general mapped to the most specific Uberon terms possible, 10 broader tissue groups (e.g. “brain”, “connective tissue”) were identified by hand and the individual samples were mapped to these groups using Ontolopy’s `Relations()` function. This level of specificity is useful for comparing between experiments, since many experiments describe some tissues more specifically than others. For example, there are many FANTOM5 tissues labelled “brain”, but many HDBR experiments are labelled as more specific parts of the brain.

7.3.4. Final Experimental Design

Experiment	brain	central nervous system	connective tissue	respiratory system	cardiovascular system	digestive system	skin of body	reproductive system	renal system	muscle tissue	Samples per experiment
FANTOM5	75	30	28	28	203	50	5	37	37	0	528
GTEx	1879	253	757	721	3328	2483	1302	1246	94	1493	13556
HDBR	356	222	0	0	0	0	0	0	0	0	578
HPA	0	3	7	8	9	56	6	42	10	9	158
Total	2310	508	792	757	3540	2589	1313	1325	141	1502	14820

Fig. 7.4 A table showing the number of samples in each category, by tissue group and experiment. Note that the design is not balanced: there are some categories that do not overlap at all.

Fig. 7.4 shows the experimental design of the combined data set. Since it is not balanced, it is not likely to be suitable for batch-correction algorithms such as ComBat or ComBat-Seq.

7.4. Results and discussion

By harmonising the metadata of the four gene expression experiments, I have made it possible to query these four large data sets together, and I show an example of this. I have made the harmonised metadata for these experiments available for download through the Open Science Framework, here.

The combined data set represents 122 healthy tissues (all of which map to Uberon terms), over almost 20,000 samples, all which have consistent labelled sample information (age, development stage, sex). This wider variety of information can be used to increase coverage when gene expression data is needed for input to algorithms, which is done in the Filip chapter.

7.4.1. Example: Tissue-specific expression comparison

To illustrate the benefit of combining datasets, I will demonstrate that even the largest and most comprehensive gene expression experiments do not show all genes that are capable of expression being expressed.

I looked at the tissue group *brain*, since all experiments have tissues in this group (see Fig. 7.6). These samples represent **38** different brain tissues, Fig. 7.5 shows the most prevalent subtissue types.

	Frequency in samples
cerebral cortex	383
cerebellum	314
caudate nucleus	267
pituitary gland	249
nucleus accumbens	244

Fig. 7.5 The five most common subtissues making up the brain tissue group.

	Brain tissues
Experiment	
FANTOM5	98
GTEx	2840
HDBR	480
HPA	3

Fig. 7.6 The breakdown of samples per source experiment in the brain tissue group.

To illustrate the fact that the different sources do not agree on whether or not genes are expressed, I first chose a random subset of 1000 genes from the combined dataset, then normalised the counts into TPM (Tags Per Million) (using average transcript length from BioMart[277]). I then identified a TPM cutoff per experiment to reduce noise by graphing pairs of *cerebral cortex* samples in each experiment, looking for a threshold where like samples are not as similar as we would imagine.

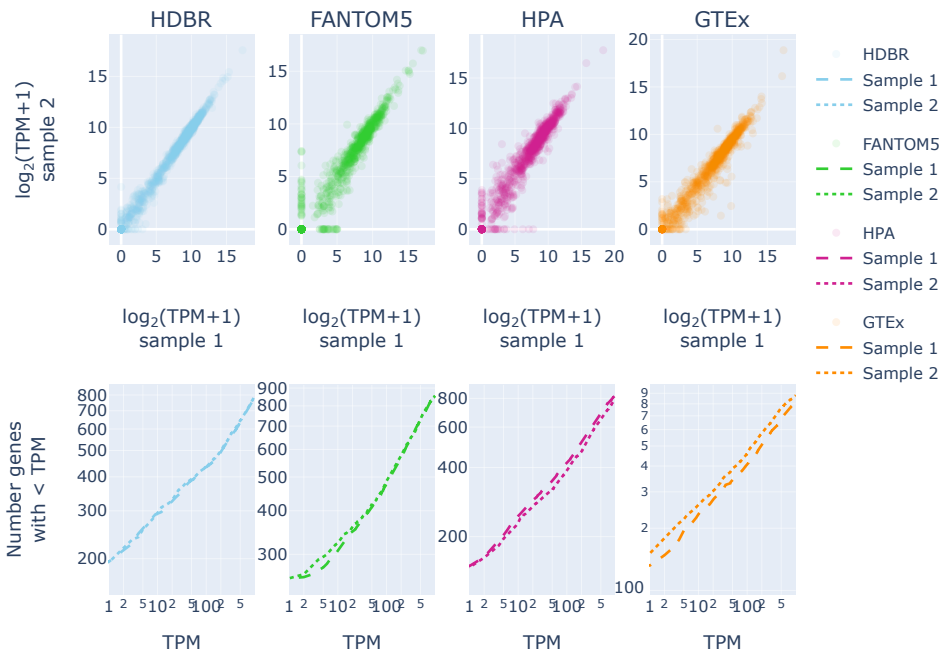


Fig. 7.7 Scatter plot (above) and cumulative histograms (bottom) showing two similar *cerebral cortex* samples from each experiment.

Experiment	TPM cut-off
HDBR	5
FANTOM5	50
HPA	10
GTEx	25

Table 7.2 Table showing the cut-offs chosen for each experiment.

In Fig. 7.7, I am looking for thresholds above which the samples correlate more strongly, as well as nonlinear behaviour in the low TPMs in the bottom plots, as described here[278]. I chose thresholds as shown in Table 7.2.

I then define "unexpressed" genes as genes which on average across samples in an experiment have a lower mean than this noise threshold, and calculated the genes that were unexpressed in brain samples from each experiment. Calculating the inter-rater reliability using Cohen's Kappa (which adjusts for the probability of randomly rating samples the same way) between experiments reveals that there is moderate agreement between samples when using the per-experiment cut-offs chosen (see Fig. 7.8).

	HDBR	FANTOM5	HPA	GTEx
HDBR	1.000000			
FANTOM5	0.575297	1.0		
HPA	0.730815	0.720104	1.0	
GTEx	0.763625	0.701627	0.875393	1.0

Fig. 7.8 Inter-rater reliability (Cohen's Kappa) for unexpressed genes. The score can vary between -1 and 1, with scores below 0 representing random variation and 1 representing perfect agreement.

Although the different experiments do have moderate agreement, there is also a lot to be gained by combining them. Fig. 7.9 shows the overlap between unexpressed genes for brain, found in each experiment.

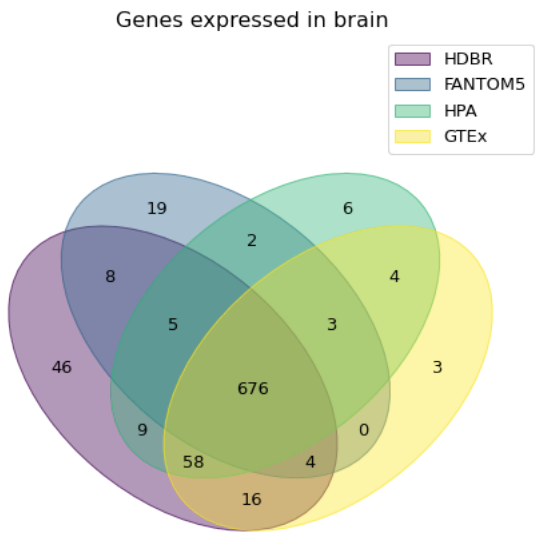


Fig. 7.9 Venn diagram showing the number of unique genes found in each experiment.

7.4.2. Batch effects

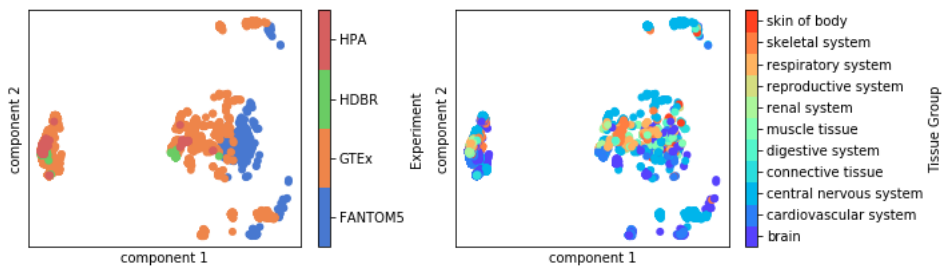


Fig. 7.10 PCA (Principal Components Analysis plot showing batch effects present in the combined dataset.

Batch effects are clearly present in the combined data set (see Fig. 7.10). I attempted batch effect correction using ComBat, however the PCA of the resulting data set did not cluster clearly by tissue type after batch effect removal which is a sign of "over correction". ComBat requires a balanced experimental design which, as we have seen, is lacking in this combined dataset, so it is likely that is the reason for it's unsuitability.

This means that the data set can not be used as-is for the purpose of measuring baseline expression (e.g. identifying housekeeping genes or measuring baseline tissue-specific gene expression). I explain some ideas for making the combined data set suitable for these types of analyses in future work.

However, by overcoming the data cleaning and standardisation necessary to have all datasets in the same format with the same sample metadata, the data can be used for analyses where batch and other sample metadata is used as covariates (e.g. differential expression of tissues). In its current iteration, it is also suitable for use in improving Filip, where the data set only needs to distinguish between presence and absence (as in the example above, this problem can be side-stepped by choosing a cut-off per experiment).

7.4.3. Combining omics data sets is an opportunity to improve existing resources

While a great deal of careful work has clearly been spent on making the datasets used in this analysis available and useful to researchers such as myself, there were still many barriers to their use in this circumstance. This ranged from mislabelled samples, to missing information, to having to seek data about the same experiment from multiple different sources (as we saw in Section 6.5). It is reassuring that the data issues that were discovered had clear pathways for reporting, and that some of them have already resulted in changes to the files used. In particular, I think it's important that key information that we know affects gene expression such as age, developmental stage, and sex are made available with the data set and preferably in a standardised format across experiments.

7.4.4. Future Work

The main piece of future work that I anticipate doing is using the combined dataset to improve the coverage of the Filip prediction filter.

7.4.4.1. Mapping improvements

There are also some mapping improvements which might improve the quality of the data set as a resource for other people.

Multiple membership of tissues and cells It is sometimes appropriate for samples to map to two apparently distinct Uberon terms. For example, leukocytes are known to be part of the immune system, but are found in the blood. In the FANTOM mapping, they would be mapped by name to blood, but by ontology to immune system. In this case, we could imagine mapping to two Uberon terms rather than defaulting to where the cells were collected, since researchers interested in blood or the immune system would both like to access the information.

In addition, it would be preferable to map simultaneously to tissue and cell type, since this enables researchers to, for example, make queries about expression about the same cell types in different tissue locations, query the data set against scRNA-seq data, or simply find cell as well as tissue specific information. This could be achieved partly with relative ease by using the ontological mapping between CL and Uberon. Improvement of the CL-Uberon mapping would then allow for a complete understanding of which cell types are in a tissue, but not their relative abundances.

Cell type deconvolution: In order to understand the relative abundances of cell types in each sample, a cell type deconvolution programme (e.g. CIBERSORT[279], BSEQ-sc[280], or MuSiC[281]) could be used. These algorithms estimate percentages of cell types making up a tissue. This would require the input of a large scRNA-seq data set as input, and there doesn't yet exist enough diversity to deconvolve all tissue types. As well as improving the mapping, this is likely to improve the quality and variety of batch effect correction methods available.

7.4.4.2. Batch effect removal

Many popular batch-effect removal techniques (e.g. ComBat and ComBat-Seq[282]) require a balanced experimental design, which this combined dataset does not have. It is not clear, however, to what extent this may affect their performance. Some alternative methods are not as sensitive to this requirement, e.g. Mutual Nearest Neighbour (MNN)[283], which was developed for scRNA-seq data. No batch-effect removal method is designed specifically for this kind of scenario, so it would be sensible to do a simulation study to test their suitability; some preliminary work towards this goal can be found in the appendix.

7.4.4.3. Tissue-specific vs cell specific

As the number of scRNA-seq experiments increases, including them in a combined dataset of tissue-specific expression will become more statistically viable. A prerequisite of including scRNA-seq data would be the use of an alternative batch effect removal algorithm that is suitable for single cell data (e.g. MNN). It would be interesting to compare how the expression of cells which can exist in multiple tissue types differs across those different tissue types, and to investigate whether some gene expression is truly tissue-specific rather than cell-type specific.

8. Concluding remarks

Working on Snowflake gave me a bird's eye view of our model of the connection between genotype and phenotype: and the data sets we have about that connection. It is (obviously) regrettable that we could not conclusively test it as a phenotype predictor across phenotypes. I think that the work I've done in developing Snowflake as a tool for outlier detection for unusual combinations of variants could still prove useful in the future, but we would first need to access a data set with many phenotypes. However, in Filip, I have found a small way in which to improve phenotype predictions across the genome, with a mechanistic reason behind it, and I hope to continue to improve this.

In my attempts to make explanative genome-wide predictions about protein function, I continuously bumped up against the limits of what is possible with the data that we currently have. These resources are absolutely vital to the efforts of computational biology, and are amazing feats of research, engineering, and collaboration, but there are some limits at present in using them for "big-picture" biology. As such, some of the most satisfying work has been to contribute back to some of these resources. Through linking them, and finding inconsistencies, I have in some small way been part of science's self-correcting mechanism, and hope that this brings us a little closer to their use for genome-wide explanatory predictions.

Appendix

Simulating RNA-Seq data to test batch-correction across experiments

Simulated data can be used to test that methodologies are applicable to new data types. Since simulated data has a well-defined ground truth, we can test the performance and accuracy of a methodology using it. As long as the real data is similar to the simulated data, we can assume that methodologies will perform similarly on the real data.

In order to test whether it is feasible to use batch correction to adjust the RNA-Seq experiments chosen (considering the unbalanced design), I want to create a simulated data set of tissue-specific batch-affected gene expression data. This appendix contains some preliminary work towards this goal, in estimating parameters from the combined data set that will be useful in creating the data set.

Parameters for simulation

In order to create simulated data that is similar to the real thing, decisions must be made about how to parameterise the distribution of counts per sample and how these relate to tissue specific effects and batch effects.

The `polyester` R package[284] can be used to simulate RNA-seq count data with the same design of tissues, samples, and experiments as in the combined data set, particularly the `create_read_numbers` function, which requires a model matrix that specifies the experimental design and a matrix of coefficients β that specify the sample-specific effects.

Count parameters

For gene expression count data, a zero-truncated negative binomial distribution is commonly used to represent the underlying gene expression counts because the distribution is always positive, does not assume mean and variance are equal, and can be tuned to have many zero counts as we see in real data.

The `get_params` function from `polyester` handles the paramerisation of the zero-inflated negative binomial that it uses to simulate count data, using an example data set as input. I used a cleaned version of the FANTOM5 data as input which was restricted only to genes that are common between all experiments (HDBR, HPA, FANTOM and GTEx), removing all zero rows, and set NaN counts to 0. Parameters calculated by polyester include means per gene and size, and probability of a zero count per gene.

Experimental design of simulated data

I already have "model-matrix", specifying the experimental design of the combined data set, in terms of batch and the 10 more general tissue groups that I mapped samples to using Ontolopy. These more general tissue groups contain the same specificity (and some of the exact same) terms (e.g. brain) that are in the Human Protein Atlas, which was used to parameterise the simulated count data, which is why this model matrix is a better choice than the more specific 129 Uberon terms that the samples also map to.

Estimating co-expression between genes

The `polyester` package does not include gene co-expression (a.k.a. co-occurrance): the correlation between genes of expression values, which is due to genes working together in the same networks, although some other packages do have this functionality.

In order to introduce this correlation to some extent, I used FANTOM data to estimate the correlation between gene expression and used this correlation matrix to create the tissue-specific effects over genes.

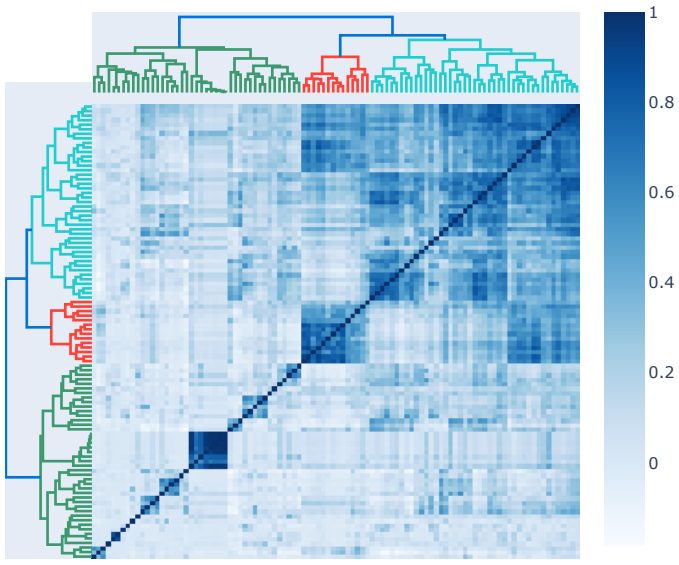


Fig. 1 Heatmap showing the correlation coefficients between 100 randomly sampled genes from the FANTOM5 data set.

The correlations between genes, which could be used to create the coefficient matrix β are shown in Fig. 1.

Distribution of fold-changes for tissue specific genes

The expected log2-fold change due to tissue-specific effects per gene and per sample (matrix β) must be pre-decided in order to simulate the data set. The size of the effect and number genes affected were estimated using data from the Human Protein Atlas (HPA) - available here - which contains for each tissue-specific gene, the transcripts per million (TPM) for tissues that were found to be tissue-enriched (at least a 5 fold change, compared to all other tissues), group-enriched (at least a 5 fold change between the group of 2-7 tissues compared to all other tissues) or tissue enhanced (at least a 5 fold change between the tissue and the average of all other tissues), and the transcripts per million of the most highly expressed tissues that were not. Taken together (tissue-enriched, group-enriched and tissue-enhanced), we here refer to these genes/tissues as tissue-specific. An excerpt of the file can be seen in Fig. 2.

Gene	RNA tissue category	RNA TS	RNA TS TPM	TPM max in non-specific
TSPAN6	Mixed	NaN	NaN	fallopian tube: 102.1
TNMD	Tissue enhanced	NaN	adipose tissue: 10.1;seminal vesicle: 33.2	breast: 4.3
DPM1	Expressed in all	NaN	NaN	thyroid gland: 80.5
SCYL3	Expressed in all	NaN	NaN	parathyroid gland: 24.6
C1orf112	Tissue enhanced	NaN	parathyroid gland: 25.0	testis: 18.6

Fig. 2 The Human protein atlas provides a csv file of TPM values for tissues with >5 fold change. This table was used to parameterise the matrix of coefficients β.

Since the HPA data does not include fold-changes of less than 5, I had no information about these changes, and decided to model the distribution of unaffected genes separately to the affected genes.

Estimating parameters of lognormal distribution of log2-fold change per gene: For any of these tissue-specific genes/tissues, the log2-fold change per tissue per gene was calculated. I first checked that each of the tissues had some tissue-specific genes according to the HPA data; this was the case.

I then extracted the multipliers from the data, and converted them to log2-fold format (expected by `polyester`). Since the distribution was long-tailed, I compared the distribution to an exponential, log-normal and power-law distribution using the python `powerlaw` package[285]. Comparative tests showed that lognormal was the best fit (with extremely low p-values, see code output below); Fig. 3 visualises this.

```
lognormal distribution is a better fit than power_law distribution with p-
value=0.0
lognormal distribution is a better fit than power_law distribution with p-
value=1.8479647159522614e-182
```

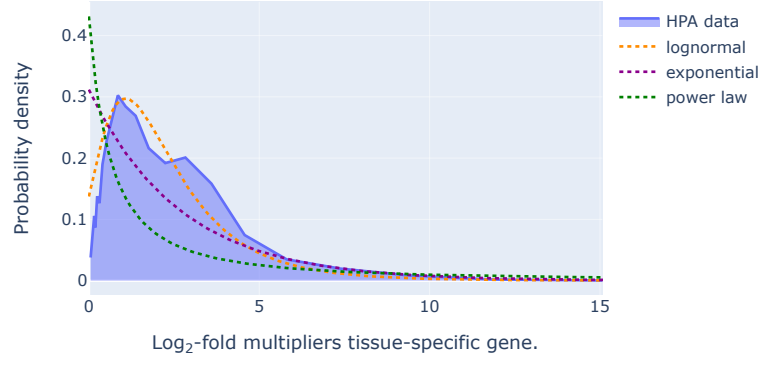


Fig. 3 The distribution of tissue-specific fold-change over all tissue-specific gene-sample pairs from HPA, fitted to lognormal, powerlaw, and exponential distributions, showing the lognormal as the best fit.

The log-normal distribution was the best fit to the data, see Fig. 3). The parameters fitting the log2-fold changes to the log-normal distribution were estimated as μ =**1.01** σ =**0.55**. Visual inspection of Fig. 3 reveals that the data simulated from these parameters appears to fit the data reasonably well, although it may be better parameterised by two overlapping distributions.

Number of tissue-specific genes per tissue: The number of tissue-specific genes per tissue was also calculated from the HPA data. Again, the data was most similar to a lognormal, still with very small p-values (see code output below), but the fit (see Fig. 4) was less convincing, probably due to the small number of tissues: 37. The distribution was parameterised with μ =**5.44** σ =**0.70**.

```
lognormal distribution is a better fit than power_law distribution with p-
value=2.872370348407609e-22
lognormal distribution is a better fit than power_law distribution with p-
value=2.872370348407609e-22
```

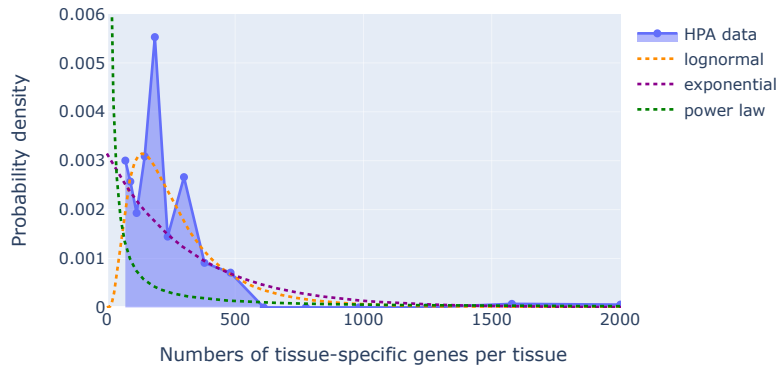


Fig. 4 The distribution of the number of tissue-specific genes per tissue from HPA, fitted to lognormal, powerlaw, and exponential distributions, showing the lognormal as the best fit.

Simulating tissue-specific RNA-Seq counts

Counts can then be simulated using `polyester` (using **`this script`**) or an alternative tool.

The simulated data set is given by: $C_{ijk} \propto NegativeBinomal(mean = \mu_{jk}, size = r_{jk})$ for replicate i, gene j, and sample k, where:

- the means are given by $\mu_{jk} = \mu'_j + \beta_{jk} \cdot mod$
- μ'_j are the estimated base means per gene
- β_{jk} are the generated matrix of log-fold changes in matrix format, including both batch and tissue effects (`coeffs_batch.csv`)
- mod is the model design matrix.
- the dispersion parameter (size), r_{jk} is calculated based on μ_{jk} and the fit between mean and size (estimated from the FANTOM5 data).

Next Steps

Next steps will be to perform batch-correction on these simulated data sets, e.g. ComBat, ComBat-Seq, and Mutual Nearest Neighbours, and performing differential expression analyses, using the input β matrix to test for ground truths.

Bibliography

References are shown at the bottom of the page on which they are cited. A full bibliography (of the entire Jupyter Book) appears below:

1 Executable Books Community. Jupyter book. February 2020. URL: https://zenodo.org/record/4539666.

2 Jan Zaucha, Jonathan Stahlhacke, Matt E Oates, Natalie Thurlby, Owen J L Rackham, Hai Fang, Ben Smithers, and Julian Gough. A proteome quality index. *Environ. Microbiol.*, 17(1):4–9, 2015. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1462-2920.12622.

3 Matt E Oates, Jonathan Stahlhacke, Dimitrios V Vavoulis, Ben Smithers, Owen J L Rackham, Adam J Sardar, Jan Zaucha, Natalie Thurlby, Hai Fang, and Julian Gough. The SUPERFAMILY 1.75 database in 2014: a doubling of data. *Nucleic Acids Res.*, 43(Database issue):D227–33, January 2015.

4 Julian Gough, Jan Zaucha, and Natalie Thurlby. Determining phenotype from genotype. July 2017. URL: https://patents.google.com/patent/US20200176085A1/en.

5 Naihui Zhou, Yuxiang Jiang, Timothy R Bergquist, Alexandra J Lee, Balint Z Kacsoh, Alex W Crocker, Kimberley A Lewis, George Georghiou, Huy N Nguyen, Md Nafiz Hamid, Larry Davis, Tunca Dogan, Volkan Atalay, Ahmet S Rifaioglu, Alperen Dalkiran, Rengul Cetin-Atalay, Chengxin Zhang, Rebecca L Hurto, Peter L Freddolino, Yang Zhang, Prajwal Bhat, Fran Supek, José M Fernández, Branislava Gemovic, Vladimir R Perovic, Radoslav S Davidović, Neven Sumonja, Nevena Veljkovic, Ehsaneddin Asgari, Mohammad R K Mofrad, Giuseppe Profiti, Castrense Savojardo, Pier Luigi Martelli, Rita Casadio, Florian Boecker, Indika Kahanda, Natalie Thurlby, Alice C McHardy, Alexandre Renaux, Rabie Saidi, Julian Gough, Alex A Freitas, Magdalena Antczak, Fabio Fabris, Mark N Wass, Jie Hou, Jianlin Cheng, Zheng Wang, Alfonso E Romero, Alberto Paccanaro, Haixuan Yang, Tatyana Goldberg, Chenguang Zhao, Liisa Holm, Petri Törönen, Alan J Medlar, Elaine Zosa, Itamar Borukhov, Ilya Novikov, Angela Wilkins, Olivier Lichtarge, Po-Han Chi, Wei-Cheng Tseng, Michal Linial, Peter W Rose, Christophe Dessimoz, Vedrana Vidulin, Saso Dzeroski, Ian Sillitoe, Sayoni Das, Jonathan Gill Lees, David T Jones, Cen Wan, Domenico Cozzetto, Rui Fa, Mateo Torres, Alex Wiarwick Vesztrocy, Jose Manuel Rodriguez, Michael L Tress, Marco Frasca, Marco Notaro, Giuliano Grossi, Alessandro Petrini, Matteo Re, Giorgio Valentini, Marco Mesiti, Daniel B Roche, Jonas Reeb, David W Ritchie, Sabeur Aridhi, Seyed Ziaeddin Alborzi, Marie-Dominique Devignes, Koo Da Chen Emily, Richard Bonneau, Vladimir Gligorijević, Meet Barot, Hai Fang, Stefano Toppo, Enrico Lavezzo, Marco Falda, Michele Berselli, Silvio C E Tosatto, Marco Carraro, Damiano Piovesan, Hafeez Ur Rehman, Qizhong Mao, Shanshan Zhang, Slobodan Vucetic, Gage S Black, Dane Jo, Dallas J Larsen, Ashton R Omdahl, Luke W Sagers, Erica Suh, Jonathan B Dayton, Liam J McGuffin, Danielle A Brackenridge, Patricia C Babbitt, Jeffrey M Yunes, Paolo Fontana, Feng Zhang, Shanfeng Zhu, Ronghui You, Zihan Zhang, Suyang Dai, Shuwei Yao, Weidong Tian, Renzhi Cao, Caleb Chandler, Miguel Amezola, Devon Johnson, Jia-Ming Chang, Wen-Hung Liao, Yi-Wei Liu, Stefano Pascarelli, Yotam Frank, Robert Hoehndorf, Maxat Kulmanov, Imane Boudellioua, Gianfranco Politano, Stefano Di Carlo, Alfredo Benso, Kai Hakala, Filip Ginter, Farrokh Mehryary, Suwisa Kaewphan, Jari Björne, Hans Moen, Martti E E Tolvanen, Tapio Salakoski, Daisuke Kihara, Aashish Jain, Tomislav Šmuc, Adrian Altenhoff, Asa Ben-Hur, Burkhard Rost, Steven E Brenner, Christine A Orengo, Constance J Jeffery, Giovanni Bosco, Deborah A Hogan, Maria J Martin, Claire O'Donovan, Sean D Mooney, Casey S Greene, Predrag Radivojac, and Iddo Friedberg. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *bioRxiv*, pages 653105, May 2019.

6 Welcome to the krita 4.4 manual! — krita manual 4.4.0 documentation. https://docs.krita.org/en/index.html. Accessed: 2021-2-22. URL: https://docs.krita.org/en/index.html.

7 Eric Bonabeau. Agent-based modeling: methods and techniques for simulating human systems. *Proc. Natl. Acad. Sci. U. S. A.*, 99 Suppl 3:7280–7287, May 2002. URL: http://dx.doi.org/10.1073/pnas.082080899.

8 A M Turing. The chemical basis of morphogenesis. 1953. *Bull. Math. Biol.*, 52(1-2):153–97; discussion 119–52, 1990. URL: http://dx.doi.org/10.1007/BF02459572.

9 Kristie Whitaker and Olivia Guest. # bropenscience is broken science: kirstie whitaker and olivia guest ask how open `open science'really is. *Psychologist*, 33:34–37, 2020. URL: https://pure.mpg.de/rest/items/item_3286863/component/file_3286864/content.

10 Steven Rose. Darwin, race and gender. *EMBO Rep.*, 10(4):297–298, April 2009. URL: http://dx.doi.org/10.1038/embor.2009.40.

11 Conway Zirkle. The inheritance of acquired characters and the provisional hypothesis of pangenesis. *Am. Nat.*, 69(724):417–445, September 1935.

12 Anthony Grafton and Nancy G Siraisi. Natural particulars: nature and the disciplines in renaissance europe. In *acls humanities e-book*. MPublishing, University of Michigan Library, 1999.

13 Charles Darwin. *On the Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life*. International Book Company, 1913. URL: http://lnmcp.mf.uni-lj.si/ISN/Darwin_C.doc.

14 Daniel J Fairbanks. Mendel and darwin: untangling a persistent enigma. *Heredity*, 124(2):263–273, February 2020. URL: http://dx.doi.org/10.1038/s41437-019-0289-9.

15 Gregor Mendel. Experiments in plant hybridization (1865). *Verhandlungen des naturforschenden Vereins Brünn) Available online*, 1996. URL: http://old.esp.org/foundations/genetics/classical/gm-65-a.pdf.

16 W Johannsen. The genotype conception of heredity. *Am. Nat.*, 45(531):129–159, March 1911. URL: https://doi.org/10.1086/279202.

17 Richard J Evans. RA fisher and the science of hatred. *New Stateman*, June 2020. URL: https://www.newstatesman.com/international/science-tech/2020/07/ra-fisher-and-science-hatred.

18 J D Watson and F H Crick. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, April 1953. URL: http://dx.doi.org/10.1038/171737a0.

19 J C Venter, M D Adams, E W Myers, P W Li, R J Mural, G G Sutton, H O Smith, M Yandell, C A Evans, R A Holt, J D Gocayne, P Amanatides, R M Ballew, D H Huson, J R Wortman, Q Zhang, C D Kodira, X H Zheng, L Chen, M Skupski, G Subramanian, P D Thomas, J Zhang, G L Gabor Miklos, C Nelson, S Broder, A G Clark, J Nadeau, V A McKusick, N Zinder, A J Levine, R J Roberts, M Simon, C Slayman, M Hunkapiller, R Bolanos, A Delcher, I Dew, D Fasulo, M Flanigan, L Florea, A Halpern, S Hannenhalli, S Kravitz, S Levy, C Mobarry, K Reinert, K Remington, J Abu-Threideh, E Beasley, K Biddick, V Bonazzi, R Brandon, M Cargill, I Chandramouliswaran, R Charlab, K Chaturvedi, Z Deng, V Di Francesco, P Dunn, K Eilbeck, C Evangelista, A E Gabrielian, W Gan, W Ge, F Gong, Z Gu, P Guan, T J Heiman, M E Higgins, R R Ji, Z Ke, K A Ketchum, Z Lai, Y Lei, Z Li, J Li, Y Liang, X Lin, F Lu, G V Merkulov, N Milshina, H M Moore, A K Naik, V A Narayan, B Neelam, D Nusskern, D B Rusch, S Salzberg, W Shao, B Shue, J Sun, Z Wang, A Wang, X Wang, J Wang, M Wei, R Wides, C Xiao, C Yan, A Yao, J Ye, M Zhan, W Zhang, H Zhang, Q Zhao, L Zheng, F Zhong, W Zhong, S Zhu, S Zhao, D Gilbert, S Baumhueter, G Spier, C Carter, A Cravchik, T Woodage, F Ali, H An, A Awe, D Baldwin, H Baden, M Barnstead, I Barrow, K Beeson, D Busam, A Carver, A Center, M L Cheng, L Curry, S Danaher, L Davenport, R Desilets, S Dietz, K Dodson, L Doup, S Ferriera, N Garg, A Gluecksmann, B Hart, J Haynes, C Haynes, C Heiner, S Hladun, D Hostin, J Houck, T Howland, C Ibegwam, J Johnson, F Kalush, L Kline, S Koduru, A Love, F Mann, D May, S McCawley, T McIntosh, I McMullen, M Moy, L Moy, B Murphy, K Nelson, C Pfannkoch, E Pratts, V Puri, H Qureshi,

M Reardon, R Rodriguez, Y H Rogers, D Romblad, B Ruhfel, R Scott, C Sitter, M Smallwood, E Stewart, R Strong, E Suh, R Thomas, N N Tint, S Tse, C Vech, G Wang, J Wetter, S Williams, M Williams, S Windsor, E Winn-Deen, K Wolfe, J Zaveri, K Zaveri, J F Abril, R Guigó, M J Campbell, K V Sjolander, B Karlak, A Kejariwal, H Mi, B Lazareva, T Hatton, A Narechania, K Diemer, A Muruganujan, N Guo, S Sato, V Bafna, S Istrail, R Lippert, R Schwartz, B Walenz, S Yooseph, D Allen, A Basu, J Baxendale, L Blick, M Caminha, J Carnes-Stine, P Caulk, Y H Chiang, M Coyne, C Dahlke, A Mays, M Dombroski, M Donnelly, D Ely, S Esparham, C Fosler, H Gire, S Glanowski, K Glasser, A Glodek, M Gorokhov, K Graham, B Gropman, M Harris, J Heil, S Henderson, J Hoover, D Jennings, C Jordan, J Jordan, J Kasha, L Kagan, C Kraft, A Levitsky, M Lewis, X Liu, J Lopez, D Ma, W Majoros, J McDaniel, S Murphy, M Newman, T Nguyen, N Nguyen, M Nodell, S Pan, J Peck, M Peterson, W Rowe, R Sanders, J Scott, M Simpson, T Smith, A Sprague, T Stockwell, R Turner, E Venter, M Wang, M Wen, D Wu, M Wu, A Xia, A Zandieh, and X Zhu. The sequence of the human genome. *Science*, 291(5507):1304–1351, February 2001.

20 James Meek. Decoding DNA. *The Guardian*, June 2000. URL: http://www.theguardian.com/science/2000/jun/27/genetics.uknews.

21 Amy Harmon. James watson had a chance to salvage his reputation on race. he made things worse. *The New York Times*, January 2019. URL: https://www.nytimes.com/2019/01/01/science/watson-dna-genetics-race.html.

22 Timothé Cynober. Why are there only 11 cell and gene therapies in europe? *Labiotech*, September 2020. URL: https://www.labiotech.eu/in-depth/atmp-cell-gene-therapy-ema/.

23 Guardian staff reporter. Cancer patients in england to be offered chance to avoid toxic side-effects. *The Guardian*, December 2020. URL: http://www.theguardian.com/society/2020/dec/28/cancer-patients-in-england-to-be-offered-chance-to-avoid-toxic-side-effects.

24 Gallery 19: DNA model, 1953 :: DNA learning center. https://www.dnalc.org/view/16430-Gallery-19-DNA-model-1953.html. Accessed: 2019-6-2.

25 Sheng Li and Christopher E Mason. The pivotal regulatory landscape of rna modifications. *Annual review of genomics and human genetics*, 15(1):127–150, 2014.

26 G J Mulder. Ueber die zusammensetzung einiger thierischen substanzen. *J. Prakt. Chem.*, 1839.

27 Christian B Anfinsen, Edgar Haber, Michael Sela, and FH White Jr. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, 47(9):1309, 1961.

28 Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

29 Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, and others. Highly accurate protein structure prediction for the human proteome. *Nature*, 596(7873):590–596, 2021.

30 V N Uversky. Posttranslational modification. In Stanley Maloy and Kelly Hughes, editors, *Brenner's Encyclopedia of Genetics (Second Edition)*, pages 425–430. Academic Press, San Diego, January 2013. URL: https://www.sciencedirect.com/science/article/pii/B9780123749840012031.

31 Robin C Friedman, Kyle Kai-How Farh, Christopher B Burge, and David P Bartel. Most mammalian mRNAs are conserved targets of microRNAs. *Genome Res.*, 19(1):92–105, January 2009. URL: http://dx.doi.org/10.1101/gr.082701.108.

32 Valer Gotea and Wojciech Makałowski. Do transposable elements really contribute to proteomes? *Trends Genet.*, 22(5):260–267, May 2006. URL: http://dx.doi.org/10.1016/j.tig.2006.03.006.

33 Seth W Cheetham, Geoffrey J Faulkner, and Marcel E Dinger. Overcoming challenges and dogmas to understand the functions of pseudogenes. *Nat. Rev. Genet.*, 21(3):191–201, March 2020. URL: http://dx.doi.org/10.1038/s41576-019-0196-1.

34 Chava Kimchi-Sarfaty, Jung Mi Oh, In-Wha Kim, Zuben E Sauna, Anna Maria Calcagno, Suresh V Ambudkar, and Michael M Gottesman. A" silent" polymorphism in the mdr 1 gene changes substrate specificity. *Science*, 315(5811):525–528, 2007.

35 Christina McCarthy, Alejandra Carrea, and Luis Diambra. Bicodon bias can determine the role of synonymous snps in human diseases. *BMC genomics*, 18:1–11, 2017.

36 Suresh V Ambudkar, In-Wha Kim, and Zuben E Sauna. The power of the pump: mechanisms of action of p-glycoprotein (abcb1). *European Journal of Pharmaceutical Sciences*, 27(5):392–400, 2006.

37 Nicolas Jarraud. Mecabricks. https://mecabricks.com/, 2019.

38 A G Murzin, S E Brenner, T Hubbard, and C Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247(4):536–540, April 1995. URL: http://dx.doi.org/10.1006/jmbi.1995.0159.

39 Eiko I Fried. What are psychological constructs? on the nature and statistical modelling of emotions, intelligence, personality traits and mental disorders. *Health Psychol. Rev.*, 11(2):130–134, June 2017. URL: http://dx.doi.org/10.1080/17437199.2017.1306718.

40 Ian Hacking, Emeritus University Professor Ian Hacking, and Jan Hacking. *The Social Construction of What?* Harvard University Press, 1999. URL: https://play.google.com/store/books/details?id=XkCR1p2YMRwC.

41 Thomas Szasz. The myth of mental illness. In James M Humber and Robert F Almeder, editors, *Biomedical Ethics and the Law*, pages 113–122. Springer US, Boston, MA, 1976. URL: https://doi.org/10.1007/978-1-4684-2223-8_10.

42 Jocelyn Kaiser. Genetics may explain up to 25% of same-sex behavior, giant analysis reveals. *Science Magazine*, August 2019. URL: https://www.sciencemag.org/news/2019/08/genetics-may-explain-25-same-sex-behavior-giant-analysis-reveals.

43 Ken Richardson. What IQ tests test. *Theory Psychol.*, 12(3):283–314, June 2002. URL: https://doi.org/10.1177/0959354302012003012.

44 Angela Saini. The disturbing return of scientific racism. https://www.wired.co.uk/article/superior-the-return-of-race-science-angela-saini. Accessed: 2021-2-11. URL: https://www.wired.co.uk/article/superior-the-return-of-race-science-angela-saini.

45 U Schüklenk, E Stein, J Kerin, and W Byne. The ethics of genetic research on sexual orientation. *Hastings Cent. Rep.*, 27(4):6–13, July 1997. URL: https://www.ncbi.nlm.nih.gov/pubmed/9271716.

46 Xiaolin Wu and Xi Zhang. Responses to critiques on machine learning of criminality perceptions (addendum of arxiv:1611.04135). *arXiv*, November 2016. URL: http://arxiv.org/abs/1611.04135, arXiv:1611.04135.

47 Luke Stark. Facial recognition, emotion and race in animated social media. *First Monday*, September 2018. URL: http://journals.uic.edu/ojs/index.php/fm/article/view/9406.

48 Andrew Pulrang. Disabled people explained: why we say we don't want to be cured — disability thinking. https://disabilitythinking.com/disabilitythinking/2019/4/22/disabled-people-explained-why-we-say-we-dont-want-to-be-cured, April 2019. Accessed: 2021-2-11. URL: https://disabilitythinking.com/disabilitythinking/2019/4/22/disabled-people-explained-why-we-say-we-dont-want-to-be-cured.

49 Paul Steven Miller and Rebecca Leah Levine. Avoiding genetic genocide: understanding good intentions and eugenics in the complex dialogue between the medical and disability communities. *Genet. Med.*, 15(2):95–102, February 2013. URL: http://dx.doi.org/10.1038/gim.2012.102.

50 Michael Le Page. We don't know what a fifth of our genes do – and won't find out soon. *New Scientist*, February 2019. URL: https://www.newscientist.com/article/2194516-we-dont-know-what-a-fifth-of-our-genes-do-and-wont-find-out-soon/.

51 Elizabeth Pennisi. Genomics. DNA study forces rethink of what it means to be a gene. *Science*, 316(5831):1556–1557, June 2007. URL: http://dx.doi.org/10.1126/science.316.5831.1556.

52 Franziska Pfeiffer, Carsten Gröber, Michael Blank, Kristian Händler, Marc Beyer, Joachim L Schultze, and Günter Mayer. Systematic evaluation of error rates and causes in short samples in next-generation sequencing. *Sci. Rep.*, 8(1):10950, July 2018.

53 A P Jason de Koning, Wanjun Gu, Todd A Castoe, Mark A Batzer, and David D Pollock. Repetitive elements may comprise over two-thirds of the human genome. *PLoS Genet.*, 7(12):e1002384, December 2011.

54 Brian J Haas and Michael C Zody. Advancing RNA-Seq analysis. *Nat. Biotechnol.*, 28(5):421–423, May 2010.

55 Roger Bumgarner. Overview of DNA microarrays: types, applications, and their future. *Curr. Protoc. Mol. Biol.*, Chapter 22:Unit 22.1., January 2013.

56 Karen H Miga. Completing the human genome: the progress and challenge of satellite DNA assembly. *Chromosome Res.*, 23(3):421–426, September 2015. URL: http://dx.doi.org/10.1007/s10577-015-9488-2.

57 W James Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler. The human genome browser at UCSC. *Genome Res.*, 12(6):996–1006, June 2002. URL: http://dx.doi.org/10.1101/gr.229102.

58 C Harger, G Chen, A Farmer, W Huang, J Inman, D Kiphart, F Schilkey, M P Skupski, and J Weller. The genome sequence DataBase. *Nucleic Acids Res.*, 28(1):31–32, January 2000. URL: http://dx.doi.org/10.1093/nar/28.1.31.

59 T Hubbard, D Barker, E Birney, G Cameron, Y Chen, L Clark, T Cox, J Cuff, V Curwen, T Down, R Durbin, E Eyras, J Gilbert, M Hammond, L Huminiecki, A Kasprzyk, H Lehvaslaiho, P Lijnzaad, C Melsopp, E Mongin, R Pettett, M Pocock, S Potter, A Rust, E Schmidt, S Searle, G Slater, J Smith, W Spooner, A Stabenau, J Stalker, E Stupka, A Ureta-Vidal, I Vastrik, and M Clamp. The ensembl genome database project. *Nucleic Acids Res.*, 30(1):38–41, January 2002. URL: http://dx.doi.org/10.1093/nar/30.1.38.

60 Genome browser FAQ. https://genome.ucsc.edu/FAQ/FAQreleases.html. Accessed: 2020-12-13. URL: https://genome.ucsc.edu/FAQ/FAQreleases.html.

61 GATK Team. Human genome reference builds - GRCh38 or hg38 - b37 - hg19. https://gatk.broadinstitute.org/hc/en-us/articles/360035890951, June 2020. Accessed: 2020-12-13. URL: https://gatk.broadinstitute.org/hc/en-us/articles/360035890951.

62 Girum Fitihamlak Ejigu and Jaehee Jung. Review on the computational genome annotation of sequences obtained by next-generation sequencing. *Biology*, 9(9):295, 2020.

63 Steven L Salzberg. Open questions: how many genes do we have? *BMC Biol.*, 16(1):94, August 2018.

64 Alice Meadows, Laurel L Haak, and Josh Brown. Persistent identifiers: the building blocks of the research information infrastructure. *Insights Imaging*, 32(1):9, March 2019. URL: http://insights.uksg.org/articles/10.1629/uksg.457/.

65 James Vincent. Scientists rename human genes to stop microsoft excel from misreading them as dates. https://www.theverge.com/2020/8/6/21355674/human-genes-rename-microsoft-excel-misreading-dates, August 2020. Accessed: 2021-2-7. URL: https://www.theverge.com/2020/8/6/21355674/human-genes-rename-microsoft-excel-misreading-dates.

66 S T Sherry. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.*, 29(1):308–311, 2001. URL: http://dx.doi.org/10.1093/nar/29.1.308.

67 Ryan J Andrews, Levi Baber, and Walter N Moss. RNAStructuromeDB: a genome-wide database for RNA structural inference. *Sci. Rep.*, 7(1):17269, December 2017. URL: http://dx.doi.org/10.1038/s41598-017-17510-y.

68 Katherine E Richardson, Charles C Kirkpatrick, and Brent M Znosko. RNA CoSSMos 2.0: an improved searchable database of secondary structure motifs in RNA three-dimensional structures. *Database*, January 2020. URL: http://dx.doi.org/10.1093/database/baz153.

69 Robert Petryszak, Maria Keays, Y Amy Tang, Nuno A Fonseca, Elisabet Barrera, Tony Burdett, Anja Füllgrabe, Alfonso Muñoz-Pomer Fuentes, Simon Jupp, Satu Koskinen, Oliver Mannion, Laura Huerta, Karine Megy, Catherine Snow, Eleanor Williams, Mitra Barzine, Emma Hastings, Hendrik Weisser, James Wright, Pankaj Jaiswal, Wolfgang Huber, Jyoti Choudhary, Helen E Parkinson, and Alvis Brazma. Expression atlas update—an integrated database of gene and protein expression in humans, animals and plants. 2016. URL: http://dx.doi.org/10.1093/nar/gkv1045.

70 Irene Papatheodorou, Pablo Moreno, Jonathan Manning, Alfonso Muñoz-Pomer Fuentes, Nancy George, Silvie Fexova, Nuno A Fonseca, Anja Füllgrabe, Matthew Green, Ni Huang, Laura Huerta, Haider Iqbal, Monica Jianu, Suhaib Mohammed, Lingyun Zhao, Andrew F Jarnuczak, Simon Jupp, John Marioni, Kerstin Meyer, Robert Petryszak, Cesar Augusto Prada Medina, Carlos Talavera-López, Sarah Teichmann, Juan Antonio Vizcaino, and Alvis Brazma. Expression atlas update: from tissues to single cells. *Nucleic Acids Res.*, 48(D1):D77–D83, January 2020. URL: http://dx.doi.org/10.1093/nar/gkz947.

71 Ashraful Haque, Jessica Engel, Sarah A. Teichmann & Tapio Lönnberg. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Medicine*, August 2017. URL: https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-017-0467-4.

72 I Illumina. Understanding illumina quality scores. *Technical Note: Informatics*, 2014. URL: https://www.illumina.com/documents/products/technotes/technote_Q-Scores.pdf.

73 Günter P Wagner, Koryu Kin, and Vincent J Lynch. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory Biosci.*, 131(4):281–285, December 2012. URL: http://dx.doi.org/10.1007/s12064-012-0162-3.

74 H Pimentel. What the FPKM? a review of RNA-Seq expression units. 2014. URL: https://haroldpimentel.wordpress.com/2014/05/08/what-the-fpkm-a-review-rna-seq-expression-units/.

75 M D Robinson, D J McCarthy, and G K Smyth. Edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010. URL: http://dx.doi.org/10.1093/bioinformatics/btp616.

76 Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, 15(12):550, 2014. URL: http://dx.doi.org/10.1186/s13059-014-0550-8.

77 Robert T Hersh. Atlas of protein sequence and structure, 1966. *Syst. Biol.*, 16(3):262–263, September 1967.

78 Sangya Pundir, Maria J Martin, Claire O'Donovan, and The UniProt Consortium. UniProt tools. *Curr. Protoc. Bioinformatics*, pages 1.29.1–1.29.15, 2016. URL: http://dx.doi.org/10.1002/0471250953.bi0129s53.

79 M Wang, M Weiss, M Simonovic, G Haertinger, S P Schrimpf, M O Hengartner, and C von Mering. PaxDb, a database of protein abundance averages across all three domains of life. *Mol. Cell. Proteomics*, 11(8):492–500, August 2012. URL: http://dx.doi.org/10.1074/mcp.O111.014704.

80 Patroklos Samaras, Tobias Schmidt, Martin Frejno, Siegfried Gessulat, Maria Reinecke, Anna Jarzab, Jana Zecha, Julia Mergner, Piero Giansanti, Hans-Christian Ehrlich, Stephan Aiche, Johannes Rank, Harald Kienegger, Helmut Krcmar, Bernhard Kuster, and Mathias Wilhelm. ProteomicsDB: a multi-omics and multi-organism resource for life science research. *Nucleic Acids Res.*, 48(D1):D1153–D1163, January 2020. URL: http://dx.doi.org/10.1093/nar/gkz974.

81 Björn Schwanhäusser, Dorothea Busse, Na Li, Gunnar Dittmar, Johannes Schuchhardt, Jana Wolf, Wei Chen, and Matthias Selbach. Global quantification of mammalian gene expression control. *Nature*, 473(7347):337–342, May 2011. URL: http://dx.doi.org/10.1038/nature10098.

82 S P Gygi, Y Rochon, B R Franza, and R Aebersold. Correlation between protein and mRNA abundance in yeast. *Mol. Cell. Biol.*, 19(3):1720–1730, March 1999. URL: https://www.ncbi.nlm.nih.gov/pubmed/10022859.

83 Idit Kosti, Nishant Jain, Dvir Aran, Atul J Butte, and Marina Sirota. Cross-tissue analysis of gene and protein expression in normal and cancer tissues. *Sci. Rep.*, 6:24799, May 2016. URL: http://dx.doi.org/10.1038/srep24799.

84 Protein data bank.

85 Adrian Furnham. Response bias, social desirability and dissimulation. *Pers. Individ. Dif.*, 7(3):385–400, January 1986. URL: https://www.sciencedirect.com/science/article/pii/0191886986900140.

86 B Knäuper and H U Wittchen. Diagnosing major depression in the elderly: evidence for response bias in standardized diagnostic interviews? *J. Psychiatr. Res.*, 28(2):147–164, March 1994. URL: http://dx.doi.org/10.1016/0022-3956(94)90026-4.

87 Golding, Golding, Pembrey, Jones, and The Alspac Study Team. ALSPAC-The avon longitudinal study of parents and children. *Paediatr. Perinat. Epidemiol.*, 15(1):74–87, 2001. URL: http://dx.doi.org/10.1046/j.1365-3016.2001.00325.x.

88 Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, Adrian Cortes, Samantha Welsh, Alan Young, Mark Effingham, Gil McVean, Stephen Leslie, Naomi Allen, Peter Donnelly, and Jonathan Marchini. The UK biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203–209, October 2018. URL: http://dx.doi.org/10.1038/s41586-018-0579-z.

89 Kendall Powell. The broken promise that undermines human genome research. *Nature*, 590(7845):198–201, February 2021. URL: https://www.nature.com/articles/d41586-021-00331-5.

90 Joanna S Amberger, Carol A Bocchini, Alan F Scott, and Ada Hamosh. Omim. org: leveraging knowledge across phenotype–gene relationships. *Nucleic acids research*, 47(D1):D1038–D1043, 2019.

91 Peter D Stenson, Matthew Mort, Edward V Ball, Katy Evans, Matthew Hayden, Sally Heywood, Michelle Hussain, Andrew D Phillips, and David N Cooper. The human gene mutation database: towards a comprehensive repository of inherited mutation data for medical research, genetic diagnosis and next-generation sequencing studies. *Human genetics*, 136(6):665–677, 2017.

92 Annalisa Buniello, Jacqueline A L MacArthur, Maria Cerezo, Laura W Harris, James Hayhurst, Cinzia Malangone, Aoife McMahon, Joannella Morales, Edward Mountjoy, Elliot Sollis, Daniel Suveges, Olga Vrousgou, Patricia L Whetzel, Ridwan Amode, Jose A Guillen, Harpreet S Riat, Stephen J Trevanion, Peggy Hall, Heather Junkins, Paul Flicek, Tony Burdett, Lucia A Hindorff, Fiona Cunningham, and Helen Parkinson. The NHGRI-EBI GWAS catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic Acids Res.*, 47(D1):D1005–D1012, January 2019. URL: http://dx.doi.org/10.1093/nar/gky1120.

93 J Macarthur L. Emery. GWAS catalog: exploring SNP-trait associations. http://europepmc.org/article/CTX/C7914, December 2017. Accessed: 2020-9-3. URL: http://europepmc.org/article/CTX/C7914.

94 Joshua C Denny, Marylyn D Ritchie, Melissa A Basford, Jill M Pulley, Lisa Bastarache, Kristin Brown-Gentry, Deede Wang, Dan R Masys, Dan M Roden, and Dana C Crawford. PheWAS: demonstrating the feasibility of a phenome-wide scan to discover gene–disease associations. *Bioinformatics*, 26(9):1205–1210, March 2010. URL: https://academic.oup.com/bioinformatics/article-abstract/26/9/1205/201211.

95 Ivana Barbaric, Gaynor Miller, and T Neil Dear. Appearances can be deceiving: phenotypes of knockout mice. *Brief. Funct. Genomic. Proteomic.*, 6(2):91–103, June 2007.

96 Aihua Zhang, Hui Sun, Guangli Yan, Ping Wang, and Xijun Wang. Mass spectrometry-based metabolomics: applications to biomarker and metabolic pathway research. *Biomed. Chromatogr.*, 30(1):7–12, January 2016. URL: http://dx.doi.org/10.1002/bmc.3453.

97 Ganesh A Viswanathan, Jeremy Seto, Sonali Patil, German Nudelman, and Stuart C Sealfon. Getting started in biological pathway construction and analysis. *PLoS Comput. Biol.*, 4(2):e16, February 2008. URL: http://dx.doi.org/10.1371/journal.pcbi.0040016.

98 Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, Marija Milacic, Corina Duenas Roca, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Solomon Shorser, Thawfeek Varusai, Guilherme Viteri, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The reactome pathway knowledgebase. *Nucleic Acids Res.*, 46(D1):D649–D655, January 2018. URL: http://dx.doi.org/10.1093/nar/gkx1132.

99 Minoru Kanehisa, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshiaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, and Yoshihiro Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Res.*, 36(Database issue):D480–4, January 2008. URL: http://dx.doi.org/10.1093/nar/gkm882.

100 Wilfrid Blunt. *The Compleat Naturalist: A Life of Linnaeus*. Frances Lincoln, 2001. URL: https://play.google.com/store/books/details?id=B3YOvgAACAAJ.

101 Ehret. *Plantae et papiliones rariores*. Volume 1748. [London :s.n.],, 1748. URL: https://www.biodiversitylibrary.org/item/205762.

102 Dr Isabelle Charmantier. Linnaeus and race. https://www.linnean.org/learning/who-was-linnaeus/linnaeus-and-race. Accessed: 2020-11-30. URL: https://www.linnean.org/learning/who-was-linnaeus/linnaeus-and-race.

103 Lars J Jensen and Peer Bork. Ontologies in quantitative biology: a basis for comparison, integration, and discovery. *PLoS Biol.*, 8(5):e1000374, May 2010. URL: http://dx.doi.org/10.1371/journal.pbio.1000374.

104 James A Overton, Heiko Dietze, Shahim Essaid, David Osumi-Sutherland, and Christopher J Mungall. ROBOT: a command-line tool for ontology development. In *ICBO*. ceur-ws.org, 2015. URL: http://ceur-ws.org/Vol-1515/demo6.pdf.

105

Eran Eden, Roy Navon, Israel Steinfeld, Doron Lipson, and Zohar Yakhini. GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinformatics*, 10:48, February 2009. URL: http://dx.doi.org/10.1186/1471-2105-10-48.

106 M Ashburner, C A Ball, J A Blake, D Botstein, H Butler, J M Cherry, A P Davis, K Dolinski, S S Dwight, J T Eppig, M A Harris, D P Hill, L Issel-Tarver, A Kasarskis, S Lewis, J C Matese, J E Richardson, M Ringwald, G M Rubin, and G Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat. Genet.*, 25(1):25–29, May 2000. URL: http://dx.doi.org/10.1038/75556.

107 Paul D Thomas. The gene ontology and the meaning of biological function. *Methods Mol. Biol.*, 1446:15–24, 2017. URL: http://dx.doi.org/10.1007/978-1-4939-3743-1_2.

108 Evelyn Camon, Michele Magrane, Daniel Barrell, Vivian Lee, Emily Dimmer, John Maslen, David Binns, Nicola Harte, Rodrigo Lopez, and Rolf Apweiler. The gene ontology annotation (GOA) database: sharing knowledge in uniprot with gene ontology. *Nucleic Acids Res.*, 32(Database issue):D262–6, January 2004. URL: http://dx.doi.org/10.1093/nar/gkh021.

109 Christopher J Mungall, Carlo Torniai, Georgios V Gkoutos, Suzanna E Lewis, and Melissa A Haendel. Uberon, an integrative multi-species anatomy ontology. *Genome Biol.*, 13(1):R5, January 2012. URL: http://dx.doi.org/10.1186/gb-2012-13-1-r5.

110 Venkat S Malladi, Drew T Erickson, Nikhil R Podduturi, Laurence D Rowe, Esther T Chan, Jean M Davidson, Benjamin C Hitz, Marcus Ho, Brian T Lee, Stuart Miyasato, Gregory R Roe, Matt Simison, Cricket A Sloan, J Seth Strattan, Forrest Tanaka, W James Kent, J Michael Cherry, and Eurie L Hong. Ontology application and use at the ENCODE DCC. *Database*, March 2015. URL: http://dx.doi.org/10.1093/database/bav010.

111 Lynn M Schriml, Elvira Mitraka, James Munro, Becky Tauber, Mike Schor, Lance Nickle, Victor Felix, Linda Jeng, Cynthia Bearer, Richard Lichenstein, Katharine Bisordi, Nicole Campion, Brooke Hyman, David Kurland, Connor Patrick Oates, Siobhan Kibbey, Poorna Sreekumar, Chris Le, Michelle Giglio, and Carol Greene. Human disease ontology 2018 update: classification, content and workflow expansion. *Nucleic Acids Res.*, 47(D1):D955–D962, January 2019. URL: http://dx.doi.org/10.1093/nar/gky1032.

112 Sebastian Köhler, Michael Gargano, Nicolas Matentzoglu, Leigh C Carmody, David Lewis-Smith, Nicole A Vasilevsky, Daniel Danis, Ganna Balagura, Gareth Baynam, Amy M Brower, Tiffany J Callahan, Christopher G Chute, Johanna L Est, Peter D Galer, Shiva Ganesan, Matthias Griese, Matthias Haimel, Julia Pazmandi, Marc Hanauer, Nomi L Harris, Michael J Hartnett, Maximilian Hastreiter, Fabian Hauck, Yongqun He, Tim Jeske, Hugh Kearney, Gerhard Kindle, Christoph Klein, Katrin Knoflach, Roland Krause, David Lagorce, Julie A McMurry, Jillian A Miller, Monica C Munoz-Torres, Rebecca L Peters, Christina K Rapp, Ana M Rath, Shahmir A Rind, Avi Z Rosenberg, Michael M Segal, Markus G Seidel, Damian Smedley, Tomer Talmy, Yarlalu Thomas, Samuel A Wiafe, Julie Xian, Zafer Yüksel, Ingo Helbig, Christopher J Mungall, Melissa A Haendel, and Peter N Robinson. The human phenotype ontology in 2021. *Nucleic Acids Res.*, 49(D1):D1207–D1217, January 2021. URL: http://dx.doi.org/10.1093/nar/gkaa1043.

113 James Malone, Ele Holloway, Tomasz Adamusiak, Misha Kapushesky, Jie Zheng, Nikolay Kolesnikov, Anna Zhukova, Alvis Brazma, and Helen Parkinson. Modeling sample variables with an experimental factor ontology. *Bioinformatics*, 26(8):1112–1118, April 2010. URL: http://dx.doi.org/10.1093/bioinformatics/btq099.

114 S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, October 1990. URL: http://dx.doi.org/10.1016/S0022-2836(05)80360-2.

115 Antonina Andreeva, Dave Howorth, Cyrus Chothia, Eugene Kulesha, and Alexey G Murzin. SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Res.*, 42(Database issue):D310–4, January 2014. URL: http://dx.doi.org/10.1093/nar/gkt1242.

116 C A Orengo, A D Michie, S Jones, D T Jones, M B Swindells, and J M Thornton. CATH – a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997. URL: http://dx.doi.org/10.1016/s0969-2126(97)00260-8.

117 Gergely Csaba, Fabian Birzele, and Ralf Zimmer. Systematic comparison of SCOP and CATH: a new gold standard for protein structure analysis. *BMC Struct. Biol.*, 9:23, April 2009. URL: http://dx.doi.org/10.1186/1472-6807-9-23.

118 J Gough, K Karplus, R Hughey, and C Chothia. Assignment of homology to genome sequences using a library of hidden markov models that represent all proteins of known structure. *J. Mol. Biol.*, 313(4):903–919, November 2001. URL: http://dx.doi.org/10.1006/jmbi.2001.5080.

119 Hai Fang and Julian Gough. DcGO: database of domain-centric ontologies on functions, phenotypes, diseases and more. *Nucleic Acids Res.*, 41(Database issue):D536–44, January 2013. URL: http://dx.doi.org/10.1093/nar/gks1080.

120 Hai Fang and Julian Gough. A domain-centric solution to functional genomics via dcGO predictor. *BMC Bioinformatics*, 14 Suppl 3:S9, February 2013. URL: http://dx.doi.org/10.1186/1471-2105-14-S3-S9.

121 Yuxiang Jiang, Tal Ronnen Oron, Wyatt T Clark, Asma R Bankapur, Daniel D'Andrea, Rosalba Lepore, Christopher S Funk, Indika Kahanda, Karin M Verspoor, Asa Ben-Hur, Da Chen Emily Koo, Duncan Penfold-Brown, Dennis Shasha, Noah Youngs, Richard Bonneau, Alexandra Lin, Sayed M E Sahraeian, Pier Luigi Martelli, Giuseppe Profiti, Rita Casadio, Renzhi Cao, Zhaolong Zhong, Jianlin Cheng, Adrian Altenhoff, Nives Skunca, Christophe Dessimoz, Tunca Dogan, Kai Hakala, Suwisa Kaewphan, Farrokh Mehryary, Tapio Salakoski, Filip Ginter, Hai Fang, Ben Smithers, Matt Oates, Julian Gough, Petri Törönen, Patrik Koskinen, Liisa Holm, Ching-Tai Chen, Wen-Lian Hsu, Kevin Bryson, Domenico Cozzetto, Federico Minneci, David T Jones, Samuel Chapman, Dukka Bkc, Ishita K Khan, Daisuke Kihara, Dan Ofer, Nadav Rappoport, Amos Stern, Elena Cibrian-Uhalte, Paul Denny, Rebecca E Foulger, Reija Hieta, Duncan Legge, Ruth C Lovering, Michele Magrane, Anna N Melidoni, Prudence Mutowo-Meullenet, Klemens Pichler, Aleksandra Shypitsyna, Biao Li, Pooya Zakeri, Sarah ElShal, Léon-Charles Tranchevent, Sayoni Das, Natalie L Dawson, David Lee, Jonathan G Lees, Ian Sillitoe, Prajwal Bhat, Tamás Nepusz, Alfonso E Romero, Rajkumar Sasidharan, Haixuan Yang, Alberto Paccanaro, Jesse Gillis, Adriana E Sedeño-Cortés, Paul Pavlidis, Shou Feng, Juan M Cejuela, Tatyana Goldberg, Tobias Hamp, Lothar Richter, Asaf Salamov, Toni Gabaldon, Marina Marcet-Houben, Fran Supek, Qingtian Gong, Wei Ning, Yuanpeng Zhou, Weidong Tian, Marco Falda, Paolo Fontana, Enrico Lavezzo, Stefano Toppo, Carlo Ferrari, Manuel Giollo, Damiano Piovesan, Silvio C E Tosatto, Angela Del Pozo, José M Fernández, Paolo Maietta, Alfonso Valencia, Michael L Tress, Alfredo Benso, Stefano Di Carlo, Gianfranco Politano, Alessandro Savino, Hafeez Ur Rehman, Matteo Re, Marco Mesiti, Giorgio Valentini, Joachim W Bargsten, Aalt D J van Dijk, Branislava Gemovic, Sanja Glisic, Vladmir Perovic, Veljko Veljkovic, Nevena Veljkovic, Danillo C Almeida-E-Silva, Ricardo Z N Vencio, Malvika Sharan, Jörg Vogel, Lakesh Kansakar, Shanshan Zhang, Slobodan Vucetic, Zheng Wang, Michael J E Sternberg, Mark N Wass, Rachael P Huntley, Maria J Martin, Claire O'Donovan, Peter N Robinson, Yves Moreau, Anna Tramontano, Patricia C Babbitt, Steven E Brenner, Michal Linial, Christine A Orengo, Burkhard Rost, Casey S Greene, Sean D Mooney, Iddo Friedberg, and Predrag Radivojac. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.*, 17(1):184, September 2016. URL: http://dx.doi.org/10.1186/s13059-016-1037-6.

122 Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, Gaurav Pandey, Jeffrey M Yunes, Ameet S Talwalkar, Susanna Repo, Michael L Souza, Damiano Piovesan, Rita Casadio, Zheng Wang, Jianlin Cheng, Hai Fang, Julian Gough, Patrik Koskinen, Petri Törönen, Jussi Nokso-Koivisto, Liisa Holm, Domenico Cozzetto, Daniel W A Buchan, Kevin Bryson, David T Jones, Bhakti Limaye, Harshal Inamdar, Avik Datta, Sunitha K Manjari, Rajendra Joshi, Meghana Chitale, Daisuke Kihara, Andreas M Lisewski, Serkan Erdin, Eric Venner, Olivier Lichtarge, Robert Rentzsch, Haixuan Yang, Alfonso E Romero, Prajwal Bhat, Alberto Paccanaro, Tobias Hamp, Rebecca Kaßner, Stefan Seemayer, Esmeralda Vicedo, Christian Schaefer, Dominik Achten, Florian Auer, Ariane Boehm, Tatjana Braun, Maximilian Hecht, Mark Heron, Peter Hönigschmid, Thomas A Hopf, Stefanie Kaufmann, Michael Kiening, Denis Krompass, Cedric Landerer, Yannick Mahlich, Manfred Roos, Jari Björne, Tapio

Salakoski, Andrew Wong, Hagit Shatkay, Fanny Gatzmann, Ingolf Sommer, Mark N Wass, Michael J E Sternberg, Nives Škunca, Fran Supek, Matko Bošnjak, Panče Panov, Sašo Džeroski, Tomislav Šmuc, Yiannis A I Kourmpetis, Aalt D J van Dijk, Cajo J F ter Braak, Yuanpeng Zhou, Qingtian Gong, Xinran Dong, Weidong Tian, Marco Falda, Paolo Fontana, Enrico Lavezzo, Barbara Di Camillo, Stefano Toppo, Liang Lan, Nemanja Djuric, Yuhong Guo, Slobodan Vucetic, Amos Bairoch, Michal Linial, Patricia C Babbitt, Steven E Brenner, Christine Orengo, Burkhard Rost, Sean D Mooney, and Iddo Friedberg. A large-scale evaluation of computational protein function prediction. *Nat. Methods*, 10(3):221–227, March 2013. URL: http://dx.doi.org/10.1038/nmeth.2340.

123 Hashem A Shihab, Julian Gough, David N Cooper, Peter D Stenson, Gary L A Barker, Keith J Edwards, Ian N M Day, and Tom R Gaunt. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden markov models. *Hum. Mutat.*, 34(1):57–65, January 2013. URL: http://dx.doi.org/10.1002/humu.22225.

124 Peter D Stenson, Matthew Mort, Edward V Ball, Katy Howells, Andrew D Phillips, Nick St Thomas, and David N Cooper. The human gene mutation database: 2008 update. *Genome Med.*, 1(1):13, January 2009. URL: http://dx.doi.org/10.1186/gm13.

125 William McLaren, Laurent Gil, Sarah E Hunt, Harpreet Singh Riat, Graham R S Ritchie, Anja Thormann, Paul Flicek, and Fiona Cunningham. The ensembl variant effect predictor. *Genome Biol.*, 17(1):122, June 2016. URL: http://dx.doi.org/10.1186/s13059-016-0974-4.

126 Gregory McInnes, Roxana Daneshjou, Panagiostis Katsonis, Olivier Lichtarge, Rajgopal Srinivasan, Sadhna Rana, Predrag Radivojac, Sean D Mooney, Kymberleigh A Pagel, Moses Stamboulian, Yuxiang Jiang, Emidio Capriotti, Yanran Wang, Yana Bromberg, Samuele Bovo, Castrense Savojardo, Pier Luigi Martelli, Rita Casadio, Lipika R Pal, John Moult, Steven E Brenner, and Russ Altman. Predicting venous thromboembolism risk from exomes in the critical assessment of genome interpretation (CAGI) challenges. *Hum. Mutat.*, 40(9):1314–1320, September 2019. URL: http://dx.doi.org/10.1002/humu.23825.

127 Laura Kasak, Jesse M Hunter, Rupa Udani, Constantina Bakolitsa, Zhiqiang Hu, Aashish N Adhikari, Giulia Babbi, Rita Casadio, Julian Gough, Rafael F Guerrero, Yuxiang Jiang, Thomas Joseph, Panagiotis Katsonis, Sujatha Kotte, Kunal Kundu, Olivier Lichtarge, Pier Luigi Martelli, Sean D Mooney, John Moult, Lipika R Pal, Jennifer Poitras, Predrag Radivojac, Aditya Rao, Naveen Sivadasan, Uma Sunderam, V G Saipradeep, Yizhou Yin, Jan Zaucha, Steven E Brenner, and M Stephen Meyn. CAGI SickKids challenges: assessment of phenotype and variant predictions derived from clinical and genomic data of children with undiagnosed diseases. *Hum. Mutat.*, 40(9):1373–1391, September 2019. URL: http://dx.doi.org/10.1002/humu.23874.

128 The Turing Way Community, Becky Arnold, Louise Bowler, Sarah Gibson, Patricia Herterich, Rosie Higman, Anna Krystalli, Alexander Morley, Martin O'Reilly, and Kirstie Whitaker. The turing way: a handbook for reproducible data science. March 2019. URL: https://zenodo.org/record/3233986.

129 C Glenn Begley and Lee M Ellis. Drug development: raise standards for preclinical cancer research. *Nature*, 483(7391):531–533, March 2012. URL: http://dx.doi.org/10.1038/483531a.

130 Florian Prinz, Thomas Schlange, and Khusru Asadullah. Believe it or not: how much can we rely on published data on potential drug targets? *Nat. Rev. Drug Discov.*, 10(9):712–712, 2011. URL: http://dx.doi.org/10.1038/nrd3439-c1.

131 Open Science Collaboration. PSYCHOLOGY. estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716, August 2015. URL: http://dx.doi.org/10.1126/science.aac4716.

132 Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, May 2016. URL: http://dx.doi.org/10.1038/533452a.

133 Dorothy Bishop. Rein in the four horsemen of irreproducibility. *Nature*, 568(7753):435, April 2019. URL: http://dx.doi.org/10.1038/d41586-019-01307-2.

134 Daniele Fanelli. How many scientists fabricate and falsify research? a systematic review and meta-analysis of survey data. *PLoS One*, 4(5):e5738, May 2009. URL: http://dx.doi.org/10.1371/journal.pone.0005738.

135 John P A Ioannidis. Why most published research findings are false. *PLoS Med.*, 2(8):e124, August 2005. URL: http://dx.doi.org/10.1371/journal.pmed.0020124.

136 Daniel Lakens, Federico G Adolfi, Casper J Albers, Farid Anvari, Matthew A J Apps, Shlomo E Argamon, Thom Baguley, Raymond B Becker, Stephen D Benning, Daniel E Bradford, Erin M Buchanan, Aaron R Caldwell, Ben Van Calster, Rickard Carlsson, Sau-Chin Chen, Bryan Chung, Lincoln J Colling, Gary S Collins, Zander Crook, Emily S Cross, Sameera Daniels, Henrik Danielsson, Lisa DeBruine, Daniel J Dunleavy, Brian D Earp, Michele I Feist, Jason D Ferrell, James G Field, Nicholas W Fox, Amanda Friesen, Caio Gomes, Monica Gonzalez-Marquez, James A Grange, Andrew P Grieve, Robert Guggenberger, James Grist, Anne-Laura van Harmelen, Fred Hasselman, Kevin D Hochard, Mark R Hoffarth, Nicholas P Holmes, Michael Ingre, Peder M Isager, Hanna K Isotalus, Christer Johansson, Konrad Juszczyk, David A Kenny, Ahmed A Khalil, Barbara Konat, Junpeng Lao, Erik Gahner Larsen, Gerine M A Lodder, Jiří Lukavský, Christopher R Madan, David Manheim, Stephen R Martin, Andrea E Martin, Deborah G Mayo, Randy J McCarthy, Kevin McConway, Colin McFarland, Amanda Q X Nio, Gustav Nilsonne, Cilene Lino de Oliveira, Jean-Jacques Orban de Xivry, Sam Parsons, Gerit Pfuhl, Kimberly A Quinn, John J Sakon, S Adil Saribay, Iris K Schneider, Manojkumar Selvaraju, Zsuzsika Sjoerds, Samuel G Smith, Tim Smits, Jeffrey R Spies, Vishnu Sreekumar, Crystal N Steltenpohl, Neil Stenhouse, Wojciech Świątkowski, Miguel A Vadillo, Marcel A L M Van Assen, Matt N Williams, Samantha E Williams, Donald R Williams, Tal Yarkoni, Ignazio Ziano, and Rolf A Zwaan. Justify your alpha. *Nature Human Behaviour*, 2(3):168–171, February 2018. URL: https://www.nature.com/articles/s41562-018-0311-x.

137 Daniel J Benjamin, James O Berger, Magnus Johannesson, Brian A Nosek, E-J Wagenmakers, Richard Berk, Kenneth A Bollen, Björn Brembs, Lawrence Brown, Colin Camerer, David Cesarini, Christopher D Chambers, Merlise Clyde, Thomas D Cook, Paul De Boeck, Zoltan Dienes, Anna Dreber, Kenny Easwaran, Charles Efferson, Ernst Fehr, Fiona Fidler, Andy P Field, Malcolm Forster, Edward I George, Richard Gonzalez, Steven Goodman, Edwin Green, Donald P Green, Anthony G Greenwald, Jarrod D Hadfield, Larry V Hedges, Leonhard Held, Teck Hua Ho, Herbert Hoijtink, Daniel J Hruschka, Kosuke Imai, Guido Imbens, John P A Ioannidis, Minjeong Jeon, James Holland Jones, Michael Kirchler, David Laibson, John List, Roderick Little, Arthur Lupia, Edouard Machery, Scott E Maxwell, Michael McCarthy, Don A Moore, Stephen L Morgan, Marcus Munafó, Shinichi Nakagawa, Brendan Nyhan, Timothy H Parker, Luis Pericchi, Marco Perugini, Jeff Rouder, Judith Rousseau, Victoria Savalei, Felix D Schönbrodt, Thomas Sellke, Betsy Sinclair, Dustin Tingley, Trisha Van Zandt, Simine Vazire, Duncan J Watts, Christopher Winship, Robert L Wolpert, Yu Xie, Cristobal Young, Jonathan Zinman, and Valen E Johnson. Redefine statistical significance. *Nat Hum Behav*, 2(1):6–10, January 2018. URL: http://dx.doi.org/10.1038/s41562-017-0189-z.

138 Norbert L Kerr. HARKing: hypothesizing after the results are known. 1998. URL: http://dx.doi.org/10.1207/s15327957pspr0203_4.

139 Megan L Head, Luke Holman, Rob Lanfear, Andrew T Kahn, and Michael D Jennions. The extent and consequences of p-hacking in science. *PLoS Biol.*, 13(3):e1002106, March 2015. URL: http://dx.doi.org/10.1371/journal.pbio.1002106.

140 R A Fisher and F Yates. *Statistical Methods, Experimental Design, and Scientific Inference: A Re-Issue of Statistical Methods for Research Workers, the Design of Experiments, and Statistical Methods and Scientific Inference*. OUP Oxford, April 1990. URL: https://www.amazon.co.uk/Statistical-Methods-Experimental-Scientific-Inference/dp/0198522290.

141 Olive Jean Dunn. Estimation of the means of dependent variables. *The Annals of Mathematical Statistics*, 29(4):1095–1111, 1958. URL: http://dx.doi.org/10.1214/aoms/1177706443.

142 Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. URL: http://dx.doi.org/10.1111/j.2517-6161.1995.tb02031.x.

143 R Silberzahn, E L Uhlmann, D P Martin, P Anselmi, F Aust, E Awtrey, Š Bahník, F Bai, C Bannard, E Bonnier, R Carlsson, F Cheung, G Christensen, R Clay, M A Craig, A Dalla Rosa, L Dam, M H Evans, I Flores Cervantes, N Fong, M Gamez-Djokic, A Glenz, S Gordon-McKeon, T J Heaton, K Hederos, M Heene, A J Hofelich Mohr, F Högden, K Hui, M Johannesson, J Kalodimos, E Kaszubowski, D M Kennedy, R Lei, T A Lindsay, S Liverani, C R Madan, D Molden, E Molleman, R D Morey, L B Mulder, B R Nijstad, N G Pope, B Pope, J M Prenoveau, F Rink, E Robusto, H Roderique, A Sandberg, E Schlüter, F D Schönbrodt, M F Sherman, S A Sommer, K Sotak, S Spain, C Spörlein, T Stafford, L Stefanutti, S Tauber, J Ullrich, M Vianello, E-J Wagenmakers, M Witkowiak, S Yoon, and B A Nosek. Many analysts, one data set: making transparent how variations in analytic choices affect results. *Advances in Methods and Practices in Psychological Science*, 1(3):337–356, September 2018. URL: https://doi.org/10.1177/2515245917747646.

144 Rotem Botvinik-Nezer, Felix Holzmeister, Colin F Camerer, Anna Dreber, Juergen Huber, Magnus Johannesson, Michael Kirchler, Roni Iwanir, Jeanette A Mumford, R Alison Adcock, Paolo Avesani, Blazej M Baczkowski, Aahana Bajracharya, Leah Bakst, Sheryl Ball, Marco Barilari, Nadège Bault, Derek Beaton, Julia Beitner, Roland G Benoit, Ruud M W J Berkers, Jamil P Bhanji, Bharat B Biswal, Sebastian Bobadilla-Suarez, Tiago Bortolini, Katherine L Bottenhorn, Alexander Bowring, Senne Braem, Hayley R Brooks, Emily G Brudner, Cristian B Calderon, Julia A Camilleri, Jaime J Castrellon, Luca Cecchetti, Edna C Cieslik, Zachary J Cole, Olivier Collignon, Robert W Cox, William A Cunningham, Stefan Czoschke, Kamalaker Dadi, Charles P Davis, Alberto De Luca, Mauricio R Delgado, Lysia Demetriou, Jeffrey B Dennison, Xin Di, Erin W Dickie, Ekaterina Dobryakova, Claire L Donnat, Juergen Dukart, Niall W Duncan, Joke Durnez, Amr Eed, Simon B Eickhoff, Andrew Erhart, Laura Fontanesi, G Matthew Fricke, Shiguang Fu, Adriana Galván, Remi Gau, Sarah Genon, Tristan Glatard, Enrico Glerean, Jelle J Goeman, Sergej A E Golowin, Carlos González-García, Krzysztof J Gorgolewski, Cheryl L Grady, Mikella A Green, João F Guassi Moreira, Olivia Guest, Shabnam Hakimi, J Paul Hamilton, Roeland Hancock, Giacomo Handjaras, Bronson B Harry, Colin Hawco, Peer Herholz, Gabrielle Herman, Stephan Heunis, Felix Hoffstaedter, Jeremy Hogeveen, Susan Holmes, Chuan-Peng Hu, Scott A Huettel, Matthew E Hughes, Vittorio Iacovella, Alexandru D Iordan, Peder M Isager, Ayse I Isik, Andrew Jahn, Matthew R Johnson, Tom Johnstone, Michael J E Joseph, Anthony C Juliano, Joseph W Kable, Michalis Kassinopoulos, Cemal Koba, Xiang-Zhen Kong, Timothy R Koscik, Nuri Erkut Kucukboyaci, Brice A Kuhl, Sebastian Kupek, Angela R Laird, Claus Lamm, Robert Langner, Nina Lauharatanahirun, Hongmi Lee, Sangil Lee, Alexander Leemans, Andrea Leo, Elise Lesage, Flora Li, Monica Y C Li, Phui Cheng Lim, Evan N Lintz, Schuyler W Liphardt, Annabel B Losecaat Vermeer, Bradley C Love, Michael L Mack, Norberto Malpica, Theo Marins, Camille Maumet, Kelsey McDonald, Joseph T McGuire, Helena Melero, Adriana S Méndez Leal, Benjamin Meyer, Kristin N Meyer, Glad Mihai, Georgios D Mitsis, Jorge Moll, Dylan M Nielson, Gustav Nilsonne, Michael P Notter, Emanuele Olivetti, Adrian I Onicas, Paolo Papale, Kaustubh R Patil, Jonathan E Peelle, Alexandre Pérez, Doris Pischedda, Jean-Baptiste Poline, Yanina Prystauka, Shruti Ray, Patricia A Reuter-Lorenz, Richard C Reynolds, Emiliano Ricciardi, Jenny R Rieck, Anais M Rodriguez-Thompson, Anthony Romyn, Taylor Salo, Gregory R Samanez-Larkin, Emilio Sanz-Morales, Margaret L Schlichting, Douglas H Schultz, Qiang Shen, Margaret A Sheridan, Jennifer A Silvers, Kenny Skagerlund, Alec Smith, David V Smith, Peter Sokol-Hessner, Simon R Steinkamp, Sarah M Tashjian, Bertrand Thirion, John N Thorp, Gustav Tinghög, Loreen Tisdall, Steven H Tompson, Claudio Toro-Serey, Juan Jesus Torre Tresols, Leonardo Tozzi, Vuong Truong, Luca Turella, Anna E van 't Veer, Tom Verguts, Jean M Vettel, Sagana Vijayarajah, Khoi Vo, Matthew B Wall, Wouter D Weeda, Susanne Weis, David J White, David Wisniewski, Alba Xifra-Porxas, Emily A Yearling, Sangsuk Yoon, Rui Yuan, Kenneth S L Yuen, Lei Zhang, Xu Zhang, Joshua E Zosky, Thomas E Nichols, Russell A Poldrack, and Tom Schonberg. Variability in the analysis of a single neuroimaging dataset by many teams. *Nature*, 582(7810):84–88, June 2020. URL: http://dx.doi.org/10.1038/s41586-020-2314-9.

145 Lauren Cadwallader, Jason A Papin, Feilim Mac Gabhann, and Rebecca Kirk. Collaborating with our community to increase code sharing. *PLoS Comput. Biol.*, 17(3):e1008867, March 2021. URL: http://dx.doi.org/10.1371/journal.pcbi.1008867.

146 Mark D Wilkinson, Michel Dumontier, I Jsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, Jildau Bouwman, Anthony J Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J G Gray, Paul Groth, Carole Goble, Jeffrey S Grethe, Jaap Heringa, Peter A C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J Lusher, Maryann E Martone, Albert Mons, Abel L Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR guiding principles for scientific data management and stewardship. *Sci Data*, 3:160018, March 2016. URL: http://dx.doi.org/10.1038/sdata.2016.18.

147 Anna-Lena Lamprecht, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Arcila, Eva Martin Del Pico, Victoria Dominguez Del Angel, Stephanie Van De Sandt, Jon Ison, Paula Andrea Martinez, and Others. Towards FAIR principles for research software. *Data Science*, 3(1):37–59, 2020. URL: https://content.iospress.com/articles/data-science/ds190026.

148 SLOW-SCIENCE.org — bear with us, while we think. http://slow-science.org/. Accessed: 2021-2-14. URL: http://slow-science.org/.

149 Hai Fang, Matt E Oates, Ralph B Pethica, Jenny M Greenwood, Adam J Sardar, Owen J L Rackham, Philip C J Donoghue, Alexandros Stamatakis, David A de Lima Morais, and Julian Gough. A daily-updated tree of (sequenced) life as a reference for genome research. *Sci. Rep.*, 3:2015, 2013. URL: http://dx.doi.org/10.1038/srep02015.

150 Elias Dohmen, Lukas P M Kremer, Erich Bornberg-Bauer, and Carsten Kemena. DOGMA: domain-based transcriptome and proteome quality assessment. *Bioinformatics*, 32(17):2577–2581, September 2016. URL: http://dx.doi.org/10.1093/bioinformatics/btw231.

151 Joel Cracraft and Michael J Donoghue. *Assembling the Tree of Life*. Oxford University Press, July 2004. URL: https://www.amazon.co.uk/Assembling-Tree-Life-Joel-Cracraft/dp/0195172345.

152 Iupac-Iub Comm on Biochem Nomencl and Iupac-Iub Comm on. A one-letter notation for amino acid sequences. tentative rules. 1968. URL: http://dx.doi.org/10.1021/bi00848a001.

153 NCBI Resource Coordinators and NCBI Resource Coordinators. Database resources of the national center for biotechnology information. 2017. URL: http://dx.doi.org/10.1093/nar/gkw1071.

154 G Parra, K Bradnam, and I Korf. CEGMA: a pipeline to accurately annotate core genes in eukaryotic genomes. 2007. URL: http://dx.doi.org/10.1093/bioinformatics/btm071.

155 Eugene V Koonin, Natalie D Fedorova, John D Jackson, Aviva R Jacobs, Dmitri M Krylov, Kira S Makarova, Raja Mazumder, Sergei L Mekhedov, Anastasia N Nikolskaya, B Sridhar Rao, Igor B Rogozin, Sergei Smirnov, Alexander V Sorokin, Alexander V Sverdlov, Sona Vasudevan, Yuri I Wolf, Jodie J Yin, and Darren A Natale. A comprehensive evolutionary classification of proteins encoded in complete eukaryotic genomes. *Genome Biol.*, 5(2):R7, January 2004. URL: http://dx.doi.org/10.1186/gb-2004-5-2-r7.

156 Felipe A Simão, Robert M Waterhouse, Panagiotis Ioannidis, Evgenia V Kriventseva, and Evgeny M Zdobnov. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*, 31(19):3210–3212, October 2015. URL: http://dx.doi.org/10.1093/bioinformatics/btv351.

157 Roman L Tatusov, Natalie D Fedorova, John D Jackson, Aviva R Jacobs, Boris Kiryutin, Eugene V Koonin, Dmitri M Krylov, Raja Mazumder, Sergei L Mekhedov, Anastasia N Nikolskaya, B Sridhar Rao, Sergei Smirnov, Alexander V Sverdlov, Sona Vasudevan, Yuri I Wolf, Jodie J Yin, and Darren A Natale. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, 4:41, September 2003. URL: http://dx.doi.org/10.1186/1471-2105-4-41.

158 Evgenia V Kriventseva, Dmitry Kuznetsov, Fredrik Tegenfeldt, Mosè Manni, Renata Dias, Felipe A Simão, and Evgeny M Zdobnov. OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. *Nucleic Acids Res.*, 47(D1):D807–D811, January 2019. URL: http://dx.doi.org/10.1093/nar/gky1053.

159 Paul A Kitts, Deanna M Church, Françoise Thibaud-Nissen, Jinna Choi, Vichet Hem, Victor Sapojnikov, Robert G Smith, Tatiana Tatusova, Charlie Xiang, Andrey Zherikov, Michael DiCuccio, Terence D Murphy, Kim D Pruitt, and Avi Kimchi. Assembly: a resource for assembled genomes at NCBI. *Nucleic Acids Res.*, 44(D1):D73–80, January 2016. URL: http://dx.doi.org/10.1093/nar/gkv1226.

160 David Sims, Ian Sudbery, Nicholas E Ilott, Andreas Heger, and Chris P Ponting. Sequencing depth and coverage: key considerations in genomic analyses. *Nat. Rev. Genet.*, 15(2):121–132, February 2014. URL: http://dx.doi.org/10.1038/nrg3642.

161 Mick Watson and Amanda Warr. Errors in long-read assemblies can critically affect protein prediction. *Nat. Biotechnol.*, 37(2):124–126, February 2019. URL: http://dx.doi.org/10.1038/s41587-018-0004-z.

162 Yehudit Hasin, Marcus Seldin, and Aldons Lusis. Multi-omics approaches to disease. *Genome Biol.*, 2017. URL: http://dx.doi.org/10.1186/s13059-017-1215-1.

163 Marylyn D Ritchie, Emily R Holzinger, Ruowang Li, Sarah A Pendergrass, and Dokyoon Kim. Methods of integrating data to uncover genotype-phenotype interactions. *Nat. Rev. Genet.*, 16(2):85–97, February 2015. URL: http://dx.doi.org/10.1038/nrg3868.

164 Vessela N Kristensen, Ole Christian Lingjærde, Hege G Russnes, Hans Kristian M Vollan, Arnoldo Frigessi, and Anne-Lise Børresen-Dale. Principles and methods of integrative genomic analyses in cancer. *Nat. Rev. Cancer*, 14(5):299–313, May 2014. URL: http://dx.doi.org/10.1038/nrc3721.

165 Chang Lu, Jan Zaucha, Rihab Gam, Hai Fang, Ben Smithers, Matt E Oates, Miguel Bernabe-Rubio, James Williams, Natalie Zelenka, Arun Prasad Pandurangan, and others. Hypothesis-free phenotype prediction within a genetics-first framework. *Nature Communications*, 14(1):919, 2023.

166 Asif Javed, Saloni Agrawal, and Pauline C Ng. Phen-Gen: combining phenotype and genotype to analyze rare disorders. *Nat. Methods*, 11(9):935–937, September 2014. URL: http://dx.doi.org/10.1038/nmeth.3046.

167 Damian Smedley, Anika Oellrich, Sebastian Köhler, Barbara Ruef, Sanger Mouse Genetics Project, Monte Westerfield, Peter Robinson, Suzanna Lewis, and Christopher Mungall. PhenoDigm: analyzing curated annotations to associate animal models with human diseases. *Database*, 2013:bat025, May 2013. URL: http://dx.doi.org/10.1093/database/bat025.

168 Peter N Robinson, Sebastian Köhler, Anika Oellrich, Sanger Mouse Genetics Project, Kai Wang, Christopher J Mungall, Suzanna E Lewis, Nicole Washington, Sebastian Bauer, Dominik Seelow, Peter Krawitz, Christian Gilissen, Melissa Haendel, and Damian Smedley. Improved exome prioritization of disease genes through cross-species phenotype comparison. *Genome Res.*, 24(2):340–348, February 2014. URL: http://dx.doi.org/10.1101/gr.160325.113.

169 Martin Kircher, Daniela M Witten, Preti Jain, Brian J O'Roak, Gregory M Cooper, and Jay Shendure. A general framework for estimating the relative pathogenicity of human genetic variants. *Nat. Genet.*, 46(3):310–315, March 2014. URL: http://dx.doi.org/10.1038/ng.2892.

170 Damian Smedley, Julius O B Jacobsen, Marten Jäger, Sebastian Köhler, Manuel Holtgrewe, Max Schubach, Enrico Siragusa, Tomasz Zemojtel, Orion J Buske, Nicole L Washington, William P Bone, Melissa A Haendel, and Peter N Robinson. Next-generation diagnostics and disease-gene discovery with the exomiser. *Nat. Protoc.*, 10(12):2004–2015, November 2015. URL: https://www.nature.com/articles/nprot.2015.124.

171 Hui Yang, Peter N Robinson, and Kai Wang. Phenolyzer: phenotype-based prioritization of candidate genes for human diseases. *Nat. Methods*, 12(9):841–843, September 2015. URL: http://dx.doi.org/10.1038/nmeth.3484.

172 Damian Smedley, Max Schubach, Julius O B Jacobsen, Sebastian Köhler, Tomasz Zemojtel, Malte Spielmann, Marten Jäger, Harry Hochheiser, Nicole L Washington, Julie A McMurry, Melissa A Haendel, Christopher J Mungall, Suzanna E Lewis, Tudor Groza, Giorgio Valentini, and Peter N Robinson. A Whole-Genome analysis framework for effective identification of pathogenic regulatory variants in mendelian disease. *Am. J. Hum. Genet.*, 99(3):595–606, September 2016. URL: http://dx.doi.org/10.1016/j.ajhg.2016.07.005.

173 Tomasz Zemojtel, Sebastian Köhler, Luisa Mackenroth, Marten Jäger, Jochen Hecht, Peter Krawitz, Luitgard Graul-Neumann, Sandra Doelken, Nadja Ehmke, Malte Spielmann, Nancy Christine Oien, Michal R Schweiger, Ulrike Krüger, Götz Frommer, Björn Fischer, Uwe Kornak, Ricarda Flöttmann, Amin Ardeshirdavani, Yves Moreau, Suzanna E Lewis, Melissa Haendel, Damian Smedley, Denise Horn, Stefan Mundlos, and Peter N Robinson. Effective diagnosis of genetic disease by computational phenotype analysis of the disease-associated genome. *Sci. Transl. Med.*, 6(252):252ra123, September 2014. URL: http://dx.doi.org/10.1126/scitranslmed.3009262.

174 Sebastian Köhler, Marcel H Schulz, Peter Krawitz, Sebastian Bauer, Sandra Dölken, Claus E Ott, Christine Mundlos, Denise Horn, Stefan Mundlos, and Peter N Robinson. Clinical diagnostics in human genetics with semantic similarity searches in ontologies. *Am. J. Hum. Genet.*, 85(4):457–464, October 2009. URL: http://dx.doi.org/10.1016/j.ajhg.2009.09.003.

175 Peristera Paschou, Elad Ziv, Esteban G Burchard, Shweta Choudhry, William Rodriguez-Cintron, Michael W Mahoney, and Petros Drineas. PCA-correlated SNPs for structure identification in worldwide human populations. *PLoS Genet.*, 3(9):1672–1686, September 2007. URL: http://dx.doi.org/10.1371/journal.pgen.0030160.

176 M B Eisen, P T Spellman, P O Brown, and D Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. U. S. A.*, 95(25):14863–14868, December 1998. URL: http://dx.doi.org/10.1073/pnas.95.25.14863.

177 Habtom W Ressom, Rency S Varghese, Zhen Zhang, Jianhua Xuan, and Robert Clarke. Classification algorithms for phenotype prediction in genomics and proteomics. *Front. Biosci.*, 13:691–708, January 2008. URL: http://dx.doi.org/10.2741/2712.

178 Michael H Cho, George R Washko, Thomas J Hoffmann, Gerard J Criner, Eric A Hoffman, Fernando J Martinez, Nan Laird, John J Reilly, and Edwin K Silverman. Cluster analysis in severe emphysema subjects using phenotype and genotype data: an exploratory investigation. *Respir. Res.*, 11:30, March 2010. URL: http://dx.doi.org/10.1186/1465-9921-11-30.

179 Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957. URL: https://play.google.com/store/books/details?id=wdtoPwAACAAJ.

180 A Zimek, E Schubert, and H P Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Stat. Anal. Data Min.*, 2012. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11161.

181 The 1000 Genomes Project Consortium and The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015. URL: http://dx.doi.org/10.1038/nature15393.

182 Cynthia L Smith, Carroll-Ann W Goldsmith, and Janan T Eppig. The mammalian phenotype ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biol.*, 6(1):R7, 2005. URL: http://dx.doi.org/10.1186/gb-2004-6-1-r7.

183

P N Robinson and S Mundlos. The human phenotype ontology. *Clin. Genet.*, 77(6):525–534, 2010. URL: http://dx.doi.org/10.1111/j.1399-0004.2010.01436.x.

184 Lynn Marie Schriml, Cesar Arze, Suvarna Nadendla, Yu-Wei Wayne Chang, Mark Mazaitis, Victor Felix, Gang Feng, and Warren Alden Kibbe. Disease ontology: a backbone for disease semantic integration. *Nucleic Acids Res.*, 40(Database issue):D940–6, January 2012. URL: http://dx.doi.org/10.1093/nar/gkr972.

185 Alex Bateman, Ewan Birney, Lorenzo Cerruti, Richard Durbin, Laurence Etwiller, Sean R Eddy, Sam Griffiths-Jones, Kevin L Howe, Mhairi Marshall, and Erik L L Sonnhammer. The pfam protein families database. *Nucleic Acids Res.*, 30(1):276–280, January 2002. URL: http://dx.doi.org/10.1093/nar/30.1.276.

186 Jan Poland and Thomas Zeugmann. Clustering pairwise distances with missing data: maximum cuts versus normalized cuts. *Discovery Science*, pages 197–208, 2006. URL: http://dx.doi.org/10.1007/11893318_21.

187 Barry Smith, Werner Ceusters, Bert Klagges, Jacob Köhler, Anand Kumar, Jane Lomax, Chris Mungall, Fabian Neuhaus, Alan L Rector, and Cornelius Rosse. Relations in biomedical ontologies. *Genome Biol.*, 6(5):R46, April 2005. URL: http://dx.doi.org/10.1186/gb-2005-6-5-r46.

188 H J Lowe and G O Barnett. Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches. *JAMA*, 271(14):1103–1108, April 1994. URL: https://www.ncbi.nlm.nih.gov/pubmed/8151853.

189 1000 Genomes Project Consortium, Goncalo R Abecasis, Adam Auton, Lisa D Brooks, Mark A DePristo, Richard M Durbin, Robert E Handsaker, Hyun Min Kang, Gabor T Marth, and Gil A McVean. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, November 2012. URL: http://dx.doi.org/10.1038/nature11632.

190 Bjarni V Halldorsson, Hannes P Eggertsson, Kristjan HS Moore, Hannes Hauswedell, Ogmundur Eiriksson, Magnus O Ulfarsson, Gunnar Palsson, Marteinn T Hardarson, Asmundur Oddsson, Brynjar O Jensson, and others. The sequences of 150,119 genomes in the uk biobank. *Nature*, 607(7920):732–740, 2022.

191 Susan Fairley, Ernesto Lowy-Gallego, Emily Perry, and Paul Flicek. The international genome sample resource (IGSR) collection of open human genomic variation resources. *Nucleic Acids Res.*, 48(D1):D941–D947, January 2020. URL: http://dx.doi.org/10.1093/nar/gkz836.

192 Heng Li. Tabix: fast retrieval of sequence features from generic TAB-delimited files. *Bioinformatics*, 27(5):718–719, March 2011. URL: http://dx.doi.org/10.1093/bioinformatics/btq671.

193 Cath Tyner. The UCSC genome browser coordinate counting systems. http://genome.ucsc.edu/blog/the-ucsc-genome-browser-coordinate-counting-systems/, December 2016. Accessed: 2021-2-7. URL: http://genome.ucsc.edu/blog/the-ucsc-genome-browser-coordinate-counting-systems/.

194 G M Church. The personal genome project. *Mol. Syst. Biol.*, 1:2005.0030, December 2005. URL: http://dx.doi.org/10.1038/msb4100040.

195 Philipp G Sand. A lesson not learned: allele misassignment. *Behav. Brain Funct.*, 3(1):65, 2007. URL: http://dx.doi.org/10.1186/1744-9081-3-65.

196 2.3. clustering — scikit-learn 0.21.2 documentation. https://scikit-learn.org/stable/modules/clustering.html. Accessed: 2019-6-9. URL: https://scikit-learn.org/stable/modules/clustering.html.

197 Lucien Marie Le Cam and Jerzy Neyman. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1967. URL: https://play.google.com/store/books/details?id=EN6isXsFdKgC.

198 Peter H Sudmant, Tobias Rausch, Eugene J Gardner, Robert E Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, Kai Ye, Goo Jun, Markus Hsi-Yang Fritz, and others. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75–81, 2015.

199 Sofia Inez Iqbal Kring, Lesli Hingstrup Larsen, Claus Holst, Søren Toubro, Torben Hansen, Arne Astrup, Oluf Pedersen, and Thorkild IA Sørensen. Genotype-phenotype associations in obesity dependent on definition of the obesity phenotype. *Obesity Facts*, 1(3):138–145, 2008.

200 Michael Cariaso and Greg Lennon. Snpedia: a wiki supporting personal genome annotation, interpretation and analysis. *Nucleic acids research*, 40(D1):D1308–D1312, 2012.

201 Kunihiro Yoshida, Yusaku Shimizu, Hiroshi Morita, Tomomi Okano, Haruya Sakai, Takako Ohata, Naomichi Matsumoto, Katsuya Nakamura, Ko-ichi Tazawa, Shinji Ohara, and others. Severity and progression rate of cerebellar ataxia in 16q-linked autosomal dominant cerebellar ataxia (16q-adca) in the endemic nagano area of japan. *The Cerebellum*, 8:46–51, 2009.

202 Regina Brigelius-Flohé and Maret G Traber. Vitamin e: function and metabolism. *The FASEB journal*, 13(10):1145–1155, 1999.

203 Momoko Horikoshi, Hanieh Yaghootkar, Dennis O Mook-Kanamori, Ulla Sovio, H Rob Taal, Branwen J Hennig, Jonathan P Bradfield, Beate St Pourcain, David M Evans, Pimphen Charoen, Marika Kaakinen, Diana L Cousminer, Terho Lehtimäki, Eskil Kreiner-Møller, Nicole M Warrington, Mariona Bustamante, Bjarke Feenstra, Diane J Berry, Elisabeth Thiering, Thiemo Pfab, Sheila J Barton, Beverley M Shields, Marjan Kerkhof, Elisabeth M van Leeuwen, Anthony J Fulford, Zoltán Kutalik, Jing Hua Zhao, Marcel den Hoed, Anubha Mahajan, Virpi Lindi, Liang-Kee Goh, Jouke-Jan Hottenga, Ying Wu, Olli T Raitakari, Marie N Harder, Aline Meirhaeghe, Ioanna Ntalla, Rany M Salem, Karen A Jameson, Kaixin Zhou, Dorota M Monies, Vasiliki Lagou, Mirna Kirin, Jani Heikkinen, Linda S Adair, Fowzan S Alkuraya, Ali Al-Odaib, Philippe Amouyel, Ehm Astrid Andersson, Amanda J Bennett, Alexandra I F Blakemore, Jessica L Buxton, Jean Dallongeville, Shikta Das, Eco J C de Geus, Xavier Estivill, Claudia Flexeder, Philippe Froguel, Frank Geller, Keith M Godfrey, Frédéric Gottrand, Christopher J Groves, Torben Hansen, Joel N Hirschhorn, Albert Hofman, Mads V Hollegaard, David M Hougaard, Elina Hyppönen, Hazel M Inskip, Aaron Isaacs, Torben Jørgensen, Christina Kanaka-Gantenbein, John P Kemp, Wieland Kiess, Tuomas O Kilpeläinen, Norman Klopp, Bridget A Knight, Christopher W Kuzawa, George McMahon, John P Newnham, Harri Niinikoski, Ben A Oostra, Louise Pedersen, Dirkje S Postma, Susan M Ring, Fernando Rivadeneira, Neil R Robertson, Sylvain Sebert, Olli Simell, Torsten Slowinski, Carla M T Tiesler, Anke Tönjes, Allan Vaag, Jorma S Viikari, Jacqueline M Vink, Nadja Hawwa Vissing, Nicholas J Wareham, Gonneke Willemsen, Daniel R Witte, Haitao Zhang, Jianhua Zhao, Meta-Analyses of Glucose- and Insulin-related traits Consortium (MAGIC), James F Wilson, Michael Stumvoll, Andrew M Prentice, Brian F Meyer, Ewan R Pearson, Colin A G Boreham, Cyrus Cooper, Matthew W Gillman, George V Dedoussis, Luis A Moreno, Oluf Pedersen, Maiju Saarinen, Karen L Mohlke, Dorret I Boomsma, Seang-Mei Saw, Timo A Lakka, Antje Körner, Ruth J F Loos, Ken K Ong, Peter Vollenweider, Cornelia M van Duijn, Gerard H Koppelman, Andrew T Hattersley, John W Holloway, Berthold Hocher, Joachim Heinrich, Chris Power, Mads Melbye, Mònica Guxens, Craig E Pennell, Klaus Bønnelykke, Hans Bisgaard, Johan G Eriksson, Elisabeth Widén, Hakon Hakonarson, André G Uitterlinden, Anneli Pouta, Debbie A Lawlor, George Davey Smith, Timothy M Frayling, Mark I McCarthy, Struan F A Grant, Vincent W V Jaddoe, Marjo-Riitta Jarvelin, Nicholas J Timpson, Inga Prokopenko, Rachel M Freathy, and Early Growth Genetics (EGG) Consortium. New loci associated with birth weight identify genetic links between intrauterine growth and adult height and metabolism. *Nat. Genet.*, 45(1):76–82, January 2013. URL: http://dx.doi.org/10.1038/ng.2477.

204 David A Hinds, George McMahon, Amy K Kiefer, Chuong B Do, Nicholas Eriksson, David M Evans, Beate St Pourcain, Susan M Ring, Joanna L Mountain, Uta Francke, George Davey-Smith, Nicholas J Timpson, and Joyce Y Tung. A genome-wide association meta-analysis of self-reported allergy identifies shared and allergy-specific susceptibility loci. *Nat. Genet.*, 45(8):907–911, August 2013. URL: http://dx.doi.org/10.1038/ng.2686.

205 Elise B Robinson, Beate St Pourcain, Verneri Anttila, Jack A Kosmicki, Brendan Bulik-Sullivan, Jakob Grove, Julian Maller, Kaitlin E Samocha, Stephan J Sanders, Stephan Ripke, Joanna Martin, Mads V Hollegaard, Thomas Werge, David M Hougaard, iPSYCH-SSI-Broad Autism Group, Benjamin M Neale, David M Evans, David Skuse, Preben Bo Mortensen, Anders D Børglum, Angelica Ronald, George Davey Smith, and Mark J Daly. Genetic risk for autism spectrum disorders and neuropsychiatric variation in the general population. *Nat. Genet.*, 48(5):552–555, May 2016. URL: http://dx.doi.org/10.1038/ng.3529.

206 Beate St Pourcain, C M A Haworth, O S P Davis, Kai Wang, Nicholas J Timpson, David M Evans, John P Kemp, Angelica Ronald, Tom Price, Emma Meaburn, Susan M Ring, Jean Golding, Hakon Hakonarson, R Plomin, and George Davey Smith. Heritability and genome-wide analyses of problematic peer relationships during childhood and adolescence. *Hum. Genet.*, 134(6):539–551, June 2015. URL: http://dx.doi.org/10.1007/s00439-014-1514-5.

207 Emmanouela Repapi, Ian Sayers, Louise V Wain, Paul R Burton, Toby Johnson, Ma'en Obeidat, Jing Hua Zhao, Adaikalavan Ramasamy, Guangju Zhai, Veronique Vitart, Jennifer E Huffman, Wilmar Igl, Eva Albrecht, Panos Deloukas, John Henderson, Raquel Granell, Wendy L McArdle, Alicja R Rudnicka, Wellcome Trust Case Control Consortium, Inês Barroso, Ruth J F Loos, Nicholas J Wareham, Linda Mustelin, Taina Rantanen, Ida Surakka, Medea Imboden, H Erich Wichmann, Ivica Grkovic, Stipan Jankovic, Lina Zgaga, Anna-Liisa Hartikainen, Leena Peltonen, Ulf Gyllensten, Asa Johansson, Ghazal Zaboli, Harry Campbell, Sarah H Wild, James F Wilson, Sven Gläser, Georg Homuth, Henry Völzke, Massimo Mangino, Nicole Soranzo, Tim D Spector, Ozren Polasek, Igor Rudan, Alan F Wright, Markku Heliövaara, Samuli Ripatti, Anneli Pouta, Asa Torinsson Naluai, Anna-Carin Olin, Kjell Torén, Matthew N Cooper, Alan L James, Lyle J Palmer, Aroon D Hingorani, S Goya Wannamethee, Peter H Whincup, George Davey Smith, Shah Ebrahim, Tricia M McKeever, Ian D Pavord, Andrew K MacLeod, Andrew D Morris, David J Porteous, Cyrus Cooper, Elaine Dennison, Seif Shaheen, Stefan Karrasch, Eva Schnabel, Holger Schulz, Harald Grallert, Nabila Bouatia-Naji, Jérôme Delplanque, Philippe Froguel, John D Blakey, NSHD Respiratory Study Team, John R Britton, Richard W Morris, John W Holloway, Debbie A Lawlor, Jennie Hui, Fredrik Nyberg, Marjo-Riitta Jarvelin, Cathy Jackson, Mika Kähönen, Jaakko Kaprio, Nicole M Probst-Hensch, Beate Koch, Caroline Hayward, David M Evans, Paul Elliott, David P Strachan, Ian P Hall, and Martin D Tobin. Genome-wide association study identifies five loci associated with lung function. *Nat. Genet.*, 42(1):36–44, January 2010. URL: http://dx.doi.org/10.1038/ng.501.

208 Naomi R Wray, Jian Yang, Ben J Hayes, Alkes L Price, Michael E Goddard, and Peter M Visscher. Pitfalls of predicting complex traits from SNPs. *Nat. Rev. Genet.*, 14(7):507–515, July 2013. URL: http://dx.doi.org/10.1038/nrg3457.

209 Bastian Greshake, Philipp E Bayer, Helge Rausch, and Julia Reda. openSNP–a crowdsourced web resource for personal genomics. *PLoS One*, 9(3):e89204, March 2014. URL: http://dx.doi.org/10.1371/journal.pone.0089204.

210 Chenguang Zhao and Zheng Wang. GOGO: an improved algorithm to measure the semantic similarity between gene ontology terms. *Sci. Rep.*, 2018. URL: http://dx.doi.org/10.1038/s41598-018-33219-y.

211 James Z Wang, Zhidian Du, Rapeeporn Payattakool, Philip S Yu, and Chin-Fu Chen. A new method to measure the semantic similarity of GO terms. *Bioinformatics*, 23(10):1274–1281, May 2007. URL: http://dx.doi.org/10.1093/bioinformatics/btm087.

212 Mark F Rogers, Hashem A Shihab, Matthew Mort, David N Cooper, Tom R Gaunt, and Colin Campbell. FATHMM-XF: accurate prediction of pathogenic point mutations via extended features. *Bioinformatics*, 34(3):511–513, February 2018. URL: http://dx.doi.org/10.1093/bioinformatics/btx536.

213 Alice B Popejoy and Stephanie M Fullerton. Genomics is failing on diversity. *Nature*, 538(7624):161–164, October 2016. URL: http://dx.doi.org/10.1038/538161a.

214 Kasper Lage, Niclas Tue Hansen, E Olof Karlberg, Aron C Eklund, Francisco S Roque, Patricia K Donahoe, Zoltan Szallasi, Thomas Skøt Jensen, and Søren Brunak. A large-scale analysis of tissue-specific pathology and gene expression of human disease genes and complexes. *Proc. Natl. Acad. Sci. U. S. A.*, 105(52):20870–20875, December 2008. URL: http://dx.doi.org/10.1073/pnas.0810772105.

215 Eitan E Winter, Leo Goodstadt, and Chris P Ponting. Elevated rates of protein secretion, evolution, and disease among tissue-specific genes. *Genome Res.*, 14(1):54–61, January 2004. URL: http://dx.doi.org/10.1101/gr.1924004.

216 Owen J L Rackham, Hashem A Shihab, Michael R Johnson, and Enrico Petretto. EvoTol: a protein-sequence based evolutionary intolerance framework for disease-gene prioritization. *Nucleic Acids Res.*, 43(5):e33, March 2015. URL: http://dx.doi.org/10.1093/nar/gku1322.

217 Agne Antanaviciute, Catherine Daly, Laura A Crinnion, Alexander F Markham, Christopher M Watson, David T Bonthron, and Ian M Carr. GeneTIER: prioritization of candidate disease genes using tissue-specific gene expression profiles. *Bioinformatics*, 31(16):2728–2735, August 2015. URL: http://dx.doi.org/10.1093/bioinformatics/btv196.

218 Arjun Raj and Alexander van Oudenaarden. Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell*, 135(2):216–226, October 2008. URL: http://dx.doi.org/10.1016/j.cell.2008.09.050.

219 Jong Kyoung Kim, Aleksandra A Kolodziejczyk, Tomislav Ilicic, Sarah A Teichmann, and John C Marioni. Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nat. Commun.*, 6:8687, October 2015. URL: http://dx.doi.org/10.1038/ncomms9687.

220 Nils Eling, Michael D Morgan, and John C Marioni. Challenges in measuring and understanding biological noise. *Nat. Rev. Genet.*, 20(9):536–548, September 2019. URL: http://dx.doi.org/10.1038/s41576-019-0130-6.

221 Simon Anders, Davis J McCarthy, Yunshun Chen, Michal Okoniewski, Gordon K Smyth, Wolfgang Huber, and Mark D Robinson. Count-based differential expression analysis of RNA sequencing data using R and bioconductor. *Nat. Protoc.*, 8(9):1765–1786, September 2013. URL: http://dx.doi.org/10.1038/nprot.2013.099.

222 Michael I Love, Simon Anders, and Wolfgang Huber. DESeq2 vignette: analyzing RNA-seq data with DESeq2. http://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html, February 2021. Accessed: 2021-3-21. URL: http://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html.

223 Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, and Ali Mortazavi. A survey of best practices for RNA-seq data analysis. *Genome Biol.*, 17:13, January 2016. URL: http://dx.doi.org/10.1186/s13059-016-0881-8.

224 Anton Pottegård, Maija Bruun Haastrup, Tore Bjerregaard Stage, Morten Rix Hansen, Kasper Søltoft Larsen, Peter Martin Meegaard, Line Haugaard Vrdlovec Meegaard, Henrik Horneberg, Charlotte Gils, Dorthe Dideriksen, Lise Aagaard, Anna Birna Almarsdottir, Jesper Hallas, and Per Damkier. SearCh for humouristic and extravagant acronyms and thoroughly inappropriate names for important clinical trials (SCIENTIFIC): qualitative and quantitative systematic study. *BMJ*, 349:g7092, December 2014. URL: http://dx.doi.org/10.1136/bmj.g7092.

225 Imad Abugessaisa, Shuhei Noguchi, Akira Hasegawa, Jayson Harshbarger, Atsushi Kondo, Marina Lizio, Jessica Severin, Piero Carninci, Hideya Kawaji, and Takeya Kasukawa. FANTOM5 CAGE profiles of human and mouse reprocessed for GRCh38 and GRCm38 genome assemblies. *Sci Data*, 4:170107, August 2017. URL: http://dx.doi.org/10.1038/sdata.2017.107.

226

Alexandra Witze. Wealthy funder pays reparations for use of HeLa cells. *Nature*, 587(7832):20–21, November 2020. URL: http://dx.doi.org/10.1038/d41586-020-03042-5.

227 Danielle M Pastor, Lisa S Poritz, Thomas L Olson, Christina L Kline, Leonard R Harris, Walter A Koltun, Vernon M Chinchilli, and Rosalyn B Irby. Primary cell lines: false representation or model system? a comparison of four human colorectal tumors and their coordinately established cell lines. *Int. J. Clin. Exp. Med.*, 3(1):69–83, February 2010. URL: https://www.ncbi.nlm.nih.gov/pubmed/20369042.

228 Gurvinder Kaur and Jannette M Dufour. Cell lines: valuable tools or useless artifacts. *Spermatogenesis*, 2(1):1–5, January 2012. URL: http://dx.doi.org/10.4161/spmg.19885.

229 Graham Bell. Replicates and repeats. *BMC Biol.*, 14:28, April 2016. URL: http://dx.doi.org/10.1186/s12915-016-0254-5.

230 Peter J A Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J L de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, June 2009. URL: http://dx.doi.org/10.1093/bioinformatics/btp163.

231 Sangya Pundir, Maria J Martin, Claire O'Donovan, and UniProt Consortium. UniProt tools. *Curr. Protoc. Bioinformatics*, 53:1.29.1–1.29.15, March 2016. URL: http://dx.doi.org/10.1002/0471250953.bi0129s53.

232 Hai Fang. dcGOR: an R package for analysing ontologies and protein domain annotations. *PLoS Comput. Biol.*, 10(10):e1003929, October 2014. URL: http://dx.doi.org/10.1371/journal.pcbi.1003929.

233 Fran Supek, Matko Bošnjak, Nives Škunca, and Tomislav Šmuc. REVIGO summarizes and visualizes long lists of gene ontology terms. *PLoS One*, 6(7):e21800, July 2011. URL: http://dx.doi.org/10.1371/journal.pone.0021800.

234 Christopher Buccitelli and Matthias Selbach. mRNAs, proteins and the emerging principles of gene expression control. *Nat. Rev. Genet.*, 21(10):630–644, October 2020. URL: http://dx.doi.org/10.1038/s41576-020-0258-4.

235 Goro Terai and Kiyoshi Asai. Improving the prediction accuracy of protein abundance in escherichia coli using mRNA accessibility. *Nucleic Acids Res.*, 48(14):e81, August 2020. URL: http://dx.doi.org/10.1093/nar/gkaa481.

236 D V Klopfenstein, Liangsheng Zhang, Brent S Pedersen, Fidel Ramírez, Alex Warwick Vesztrocy, Aurélien Naldi, Christopher J Mungall, Jeffrey M Yunes, Olga Botvinnik, Mark Weigel, Will Dampier, Christophe Dessimoz, Patrick Flick, and Haibao Tang. GOATOOLS: a python library for gene ontology analyses. *Sci. Rep.*, 8(1):10872, July 2018. URL: http://dx.doi.org/10.1038/s41598-018-28948-z.

237 Edison Ong, Zuoshuang Xiang, Bin Zhao, Yue Liu, Yu Lin, Jie Zheng, Chris Mungall, Mélanie Courtot, Alan Ruttenberg, and Yongqun He. Ontobee: a linked ontology data server to support ontology term dereferencing, linkage, query and integration. *Nucleic Acids Res.*, 45(D1):D347–D352, January 2017. URL: http://dx.doi.org/10.1093/nar/gkw918.

238 Martin Larralde, Alex Henrie, Philipp A., Spencer Mitchell, and Tatsuya Sakaguchi. Althonos/pronto: 2.4.1. February 2021. URL: https://zenodo.org/record/4552164.

239 Andrea C F Albuquerque, Jose L Campos dos Santos, and Alberto N de Castro. OntoBio: a biodiversity domain ontology for amazonian biological collected objects. In *2015 48th Hawaii International Conference on System Sciences*, 3770–3779. January 2015. URL: http://dx.doi.org/10.1109/HICSS.2015.453.

240 Steven Bird. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 69–72. aclweb.org, 2006. URL: https://www.aclweb.org/anthology/P06-4018.pdf.

241 Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. URL: https://doi.org/10.1038/s41586-020-2649-2, doi:10.1038/s41586-020-2649-2.

242 The pandas development team. Pandas-dev/pandas: pandas. February 2020. URL: https://doi.org/10.5281/zenodo.3509134, doi:10.5281/zenodo.3509134.

243 Christopher Woods. Developer's guide — MetaWards documentation. https://metawards.org/development.html?highlight=style. Accessed: 2021-5-1. URL: https://metawards.org/development.html?highlight=style.

244 Donald Stufft. Welcome to twine's documentation! — twine 3.4.2.dev1+geff3a45 documentation. https://twine.readthedocs.io/en/latest/, 2019. Accessed: 2021-5-1. URL: https://twine.readthedocs.io/en/latest/.

245 Holger Krekel, Bruno Oliveira, Ronny Pfannschmidt, Floris Bruynooghe, Brianna Laugher, and Florian Bruhin. Pytest x.y. 2004. URL: https://github.com/pytest-dev/pytest.

246 PyData Community. The PyData sphinx theme — PyData sphinx theme documentation. https://pydata-sphinx-theme.readthedocs.io/, 2019. Accessed: 2021-5-1. URL: https://pydata-sphinx-theme.readthedocs.io/.

247 Marina Lizio, Jayson Harshbarger, Hisashi Shimoji, Jessica Severin, Takeya Kasukawa, Serkan Sahin, Imad Abugessaisa, Shiro Fukuda, Fumi Hori, Sachi Ishikawa-Kato, Christopher J Mungall, Erik Arner, J Kenneth Baillie, Nicolas Bertin, Hidemasa Bono, Michiel de Hoon, Alexander D Diehl, Emmanuel Dimont, Tom C Freeman, Kaori Fujieda, Winston Hide, Rajaram Kaliyaperumal, Toshiaki Katayama, Timo Lassmann, Terrence F Meehan, Koro Nishikata, Hiromasa Ono, Michael Rehli, Albin Sandelin, Erik A Schultes, Peter A C 't Hoen, Zuotian Tatum, Mark Thompson, Tetsuro Toyoda, Derek W Wright, Carsten O Daub, Masayoshi Itoh, Piero Carninci, Yoshihide Hayashizaki, Alistair R R Forrest, Hideya Kawaji, and FANTOM consortium. Gateways to the FANTOM5 promoter level mammalian expression atlas. *Genome Biol.*, 16:22, January 2015. URL: http://dx.doi.org/10.1186/s13059-014-0560-6.

248 Jean-Baptiste Lamy. Owlready: ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. *Artif. Intell. Med.*, 80:11–28, July 2017. URL: http://dx.doi.org/10.1016/j.artmed.2017.07.002.

249 David Gomez-Cabrero, Imad Abugessaisa, Dieter Maier, Andrew Teschendorff, Matthias Merkenschlager, Andreas Gisel, Esteban Ballestar, Erik Bongcam-Rudloff, Ana Conesa, and Jesper Tegnér. Data integration in the era of omics: current and future challenges. *BMC Syst. Biol.*, 8 Suppl 2:I1, March 2014. URL: http://dx.doi.org/10.1186/1752-0509-8-S2-I1.

250 Gordon Bell, Tony Hey, and Alex Szalay. Computer science. beyond the data deluge. *Science*, 323(5919):1297–1298, March 2009. URL: http://dx.doi.org/10.1126/science.1170411.

251 S Cortijo, Z Aydin, S Ahnert, and others. Widespread inter-individual gene expression variability in arabidopsis thaliana. *Mol. Syst. Biol.*, 2019. URL: http://msb.embopress.org/content/15/1/e8591.abstract.

252

Ana Viñuela, Andrew A Brown, Alfonso Buil, Pei-Chien Tsai, Matthew N Davies, Jordana T Bell, Emmanouil T Dermitzakis, Timothy D Spector, and Kerrin S Small. Age-dependent changes in mean and variance of gene expression across tissues in a twin cohort. *Human molecular genetics*, 27(4):732–741, 2018.

253 Jialiang Yang, Tao Huang, Francesca Petralia, Quan Long, Bin Zhang, Carmen Argmann, Yong Zhao, Charles V Mobbs, Eric E Schadt, Jun Zhu, Zhidong Tu, and GTEx Consortium. Synchronized age-related gene expression changes across multiple tissues in human and the link to complex diseases. *Sci. Rep.*, 5:15145, October 2015. URL: http://dx.doi.org/10.1038/srep15145.

254 B Reinius and E Jazin. Prenatal sex differences in the human brain. *Mol. Psychiatry*, 14(11):987, 988–9, November 2009. URL: http://dx.doi.org/10.1038/mp.2009.79.

255 Gregory Stone, Ashley Choi, Meritxell Oliva, Joshua Gorham, Mahyar Heydarpour, Christine E Seidman, Jon G Seidman, Sary F Aranki, Simon C Body, Vincent J Carey, Benjamin A Raby, Barbara E Stranger, and Jochen D Muehlschlegel. Sex differences in gene expression in response to ischemia in the human left ventricular myocardium. *Hum. Mol. Genet.*, January 2019. URL: http://dx.doi.org/10.1093/hmg/ddz014.

256 Ueli Schibler. The daily timing of gene expression and physiology in mammals. *Dialogues Clin. Neurosci.*, 9(3):257–272, 2007. URL: https://www.ncbi.nlm.nih.gov/pubmed/17969863.

257 Valentine Svensson, Sarah A Teichmann, and Oliver Stegle. SpatialDE: identification of spatially variable genes. *Nat. Methods*, 15(5):343–346, May 2018. URL: http://dx.doi.org/10.1038/nmeth.4636.

258 Qingguo Wang, Joshua Armenia, Chao Zhang, Alexander V Penson, Ed Reznik, Liguo Zhang, Thais Minet, Angelica Ochoa, Benjamin E Gross, Christine A Iacobuzio-Donahue, Doron Betel, Barry S Taylor, Jianjiong Gao, and Nikolaus Schultz. Unifying cancer and normal RNA sequencing data from different sources. *Sci Data*, 5:180061, April 2018. URL: http://dx.doi.org/10.1038/sdata.2018.61.

259 Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat. Rev. Genet.*, 11(10):733–739, October 2010. URL: http://dx.doi.org/10.1038/nrg2825.

260 Rafael A Irizarry, Daniel Warren, Forrest Spencer, Irene F Kim, Shyam Biswal, Bryan C Frank, Edward Gabrielson, Joe G N Garcia, Joel Geoghegan, Gregory Germino, Constance Griffin, Sara C Hilmer, Eric Hoffman, Anne E Jedlicka, Ernest Kawasaki, Francisco Martínez-Murillo, Laura Morsberger, Hannah Lee, David Petersen, John Quackenbush, Alan Scott, Michael Wilson, Yanqin Yang, Shui Qing Ye, and Wayne Yu. Multiple-laboratory comparison of microarray platforms. *Nat. Methods*, 2(5):345–350, May 2005. URL: http://dx.doi.org/10.1038/nmeth756.

261 W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, January 2007. URL: http://dx.doi.org/10.1093/biostatistics/kxj037.

262 Jeffrey T Leek and John D Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet.*, 3(9):1724–1735, September 2007. URL: http://dx.doi.org/10.1371/journal.pgen.0030161.

263 Chao Chen, Kay Grennan, Judith Badner, Dandan Zhang, Elliot Gershon, Li Jin, and Chunyu Liu. Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PLoS One*, 6(2):e17238, February 2011. URL: http://dx.doi.org/10.1371/journal.pone.0017238.

264 Q Liu and M Markatou. Evaluation of methods in removing batch effects on RNA-seq data. *Infectious Diseases and Translational Medicine*, 2016. URL: http://www.tran-med.com/CN/article/downloadArticleFile.do?attachType=PDF&id=24.

265 Nuno A Fonseca, Robert Petryszak, John Marioni, and Alvis Brazma. iRAP - an integrated RNA-seq analysis pipeline. Preprint, June 2014. URL: http://dx.doi.org/10.1101/005991.

266 Tobias Maier, Marc Güell, and Luis Serrano. Correlation of mRNA and protein in complex biological samples. *FEBS Lett.*, 583(24):3966–3973, December 2009. URL: http://dx.doi.org/10.1016/j.febslet.2009.10.036.

267 Andreas Beyer, Jens Hollunder, Heinz-Peter Nasheuer, and Thomas Wilhelm. Post-transcriptional expression regulation in the yeast saccharomyces cerevisiae on a genomic scale. *Mol. Cell. Proteomics*, 3(11):1083–1092, November 2004. URL: http://dx.doi.org/10.1074/mcp.M400099-MCP200.

268 Gang Wu, Lei Nie, and Weiwen Zhang. Integrative analyses of posttranscriptional regulation in the yeast saccharomyces cerevisiae using transcriptomic and proteomic data. *Curr. Microbiol.*, 57(1):18–22, July 2008. URL: http://dx.doi.org/10.1007/s00284-008-9145-5.

269 Nancy Yiu-Lin Yu, Björn M Hallström, Linn Fagerberg, Fredrik Ponten, Hideya Kawaji, Piero Carninci, Alistair R R Forrest, Fantom Consortium, Yoshihide Hayashizaki, Mathias Uhlén, and Carsten O Daub. Complementing tissue characterization by integrating transcriptome profiling from the human protein atlas and from the FANTOM5 consortium. *Nucleic Acids Res.*, 43(14):6787–6798, August 2015. URL: http://dx.doi.org/10.1093/nar/gkv608.

270 The human protein atlas. https://www.proteinatlas.org/. Accessed: 2021-1-29. URL: https://www.proteinatlas.org/.

271 Hideya Kawaji, Marina Lizio, Masayoshi Itoh, Mutsumi Kanamori-Katayama, Ai Kaiho, Hiromi Nishiyori-Sueki, Jay W Shin, Miki Kojima-Ishiyama, Mitsuoki Kawano, Mitsuyoshi Murata, Noriko Ninomiya-Fukuda, Sachi Ishikawa-Kato, Sayaka Nagao-Sato, Shohei Noma, Yoshihide Hayashizaki, Alistair R R Forrest, Piero Carninci, and FANTOM Consortium. Comparison of CAGE and RNA-seq transcriptome profiling using clonally amplified and single-molecule next-generation sequencing. *Genome Res.*, 24(4):708–717, April 2014. URL: http://dx.doi.org/10.1101/gr.156232.113.

272 Mathias Uhlen, Per Oksvold, Linn Fagerberg, Emma Lundberg, Kalle Jonasson, Mattias Forsberg, Martin Zwahlen, Caroline Kampf, Kenneth Wester, Sophia Hober, Henrik Wernerus, Lisa Björling, and Fredrik Ponten. Towards a knowledge-based human protein atlas. *Nature Biotechnology*, 28(12):1248–1250, 2010. URL: http://dx.doi.org/10.1038/nbt1210-1248.

273 Mathias Uhlén, Linn Fagerberg, Björn M Hallström, Cecilia Lindskog, Per Oksvold, Adil Mardinoglu, Åsa Sivertsson, Caroline Kampf, Evelina Sjöstedt, Anna Asplund, Ingmarie Olsson, Karolina Edlund, Emma Lundberg, Sanjay Navani, Cristina Al-Khalili Szigyarto, Jacob Odeberg, Dijana Djureinovic, Jenny Ottosson Takanen, Sophia Hober, Tove Alm, Per-Henrik Edqvist, Holger Berling, Hanna Tegel, Jan Mulder, Johan Rockberg, Peter Nilsson, Jochen M Schwenk, Marica Hamsten, Kalle von Feilitzen, Mattias Forsberg, Lukas Persson, Fredric Johansson, Martin Zwahlen, Gunnar von Heijne, Jens Nielsen, and Fredrik Pontén. Proteomics. tissue-based map of the human proteome. *Science*, 347(6220):1260419, January 2015. URL: http://dx.doi.org/10.1126/science.1260419.

274 GTEx Consortium. The Genotype-Tissue expression (GTEx) project. *Nat. Genet.*, 45(6):580–585, June 2013. URL: http://dx.doi.org/10.1038/ng.2653.

275 Susan J Lindsay, Yaobo Xu, Steven N Lisgo, Lauren F Harkin, Andrew J Copp, Dianne Gerrelli, Gavin J Clowry, Aysha Talbot, Michael J Keogh, Jonathan Coxhead, Mauro Santibanez-Koref, and Patrick F Chinnery. HDBR expression: a unique resource for global and individual gene expression studies during early human brain development. *Front. Neuroanat.*, 10:86, October 2016. URL:

http://dx.doi.org/10.3389/fnana.2016.00086.

276 M Keays. ExpressionAtlas: download datasets from EMBL-EBI expression atlas. R package version 1.10.0. 2018. URL: https://bioconductor.org/packages/release/bioc/html/ExpressionAtlas.html.

277 Damian Smedley, Syed Haider, Benoit Ballester, Richard Holland, Darin London, Gudmundur Thorisson, and Arek Kasprzyk. BioMart–biological queries made easy. *BMC Genomics*, 10:22, January 2009. URL: http://dx.doi.org/10.1186/1471-2164-10-22.

278 Clarissa M Koch, Stephen F Chiu, Mahzad Akbarpour, Ankit Bharat, Karen M Ridge, Elizabeth T Bartom, and Deborah R Winter. A beginner's guide to analysis of RNA sequencing data. *Am. J. Respir. Cell Mol. Biol.*, 59(2):145–157, August 2018. URL: http://dx.doi.org/10.1165/rcmb.2017-0430TR.

279 Aaron M Newman, Chih Long Liu, Michael R Green, Andrew J Gentles, Weiguo Feng, Yue Xu, Chuong D Hoang, Maximilian Diehn, and Ash A Alizadeh. Robust enumeration of cell subsets from tissue expression profiles. *Nat. Methods*, 12(5):453–457, May 2015. URL: http://dx.doi.org/10.1038/nmeth.3337.

280 Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, Douglas A Melton, and Itai Yanai. A Single-Cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst*, 3(4):346–360.e4, October 2016. URL: http://dx.doi.org/10.1016/j.cels.2016.08.011.

281 Xuran Wang, Jihwan Park, Katalin Susztak, Nancy R Zhang, and Mingyao Li. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nat. Commun.*, 10(1):380, January 2019. URL: http://dx.doi.org/10.1038/s41467-018-08023-x.

282 Yuqing Zhang, Giovanni Parmigiani, and W Evan Johnson. ComBat-seq: batch effect adjustment for RNA-seq count data. *NAR Genom Bioinform*, 2(3):lqaa078, September 2020. URL: http://dx.doi.org/10.1093/nargab/lqaa078.

283 Laleh Haghverdi, Aaron T L Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.*, 36(5):421–427, June 2018. URL: http://dx.doi.org/10.1038/nbt.4091.

284 Alyssa C Frazee, Andrew E Jaffe, Ben Langmead, and Jeffrey T Leek. Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics*, 31(17):2778–2784, September 2015. URL: http://dx.doi.org/10.1093/bioinformatics/btv272.

285 Jeff Alstott, Ed Bullmore, and Dietmar Plenz. Powerlaw: a python package for analysis of heavy-tailed distributions. *PLoS One*, 9(1):e85777, January 2014. URL: http://dx.doi.org/10.1371/journal.pone.0085777.