Computational challenges in genome wide association studies: data processing, variant annotation and epistasis

Pablo Cingolani

PhD.

School of Computer Science

McGill University
Montreal, Quebec, Canada
July 2015

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

Chapter 1

Conclusions

1.1 Contributions

In this thesis I contributed to three steps involved in the analysis of human sequencing data and identifying the links between genetic variants and disease. Each step is characterized by very different problems that need to be addressed.

i) The first step is to reduce large amounts of information generated by high throughput experiments into a manageable summary. In our case, it involves reducing the raw sequencing information to a variant call set, but it could be any other features to be analyzed (RNA expression, transcript structure, enrichment peaks, genome reference assembly, etc.). This is mainly done by mapping reads to a reference genome and then using variant call algorithms. This step is characterized by requiring fast parallel algorithms and usually, due to the amount of data involved, I/O can be one of the bottlenecks. Algorithms that work on "chunks of data" instead of the whole data-set are preferred, and in many cases exist, because working on disjoint data makes the problem easier to parallelize. Usually several stages of these highly specialized algorithms are combined into a "data analysis pipeline". Programming data analysis pipelines is not trivial since it requires process coordination, robustness, scalability

and flexibility (data processing pipelines, particularly in research environments, tend to change often). Although data pipeline solutions are often available in the form of libraries, these libraries tend to make pipeline programming cumbersome or create new programming paradigms thus introducing a steep learning curve. In Chapter **??**, we address problems related to pipeline programming in a novel way by creating a new programming language, BDS, that simplifies the creation of robust, scalable and flexible data pipelines. Although the main rationale behind the development of BDS was managing our sequencing data pipelines, it is a flexible programming language that can be applied to many large data pipelines.

ii) The second step in our data analysis consists of functional annotation, prioritization and filtering of genetic variants. The main concern in the annotation step is performing an adequate filtering of what should be considered relevant variants for our experiment. Until not long ago there were no publicly available packages for functional annotation of genomic variants, in chapter **??** we introduced SnpEff & SnpSift, two variant annotation solutions that quickly became widely adopted by the research community.

iii) Finally, in Chapter **??**, we analyse the problem of finding genetic links to complex disease. This is known to be a difficult problem affected by several hidden co-factors that bias the results (e.g. population structure). Furthermore there are limitations, evidenced by missing heritability, implying that genomic links to complex disease may not be found using traditional GWAS methodologies. We show that alternative models that combine higher level information, may help to boost statistical significance.

iii.a) We proposed a new methodology for addressing a difficult problem: the detection of interacting genomic loci (epistasis) that affect disease risk. Our models combine genotype information and co-evolutionary evidence. We show that efficient algorithms make these studies computationally feasible, albeit using relatively large

computational resources.

iii.b) We were involved in a major project on GWAS of type II diabetes using a cohort of multi-ethnic unrelated individuals which results uncovered new genes linked to diabetes. We applied our epistatic GWAS models to data form this type II diabetes sequencing study of over 13,000 individuals finding suggestive evidence of interaction.

These three chapters (three steps) complete our journey from "raw data" to "biological insight" trying to find the genetic causes of complex disease.

1.2 Future work

Here we propose several improvements, extensions and future directions of work for each of the topics discussed in this thesis.

BigDataScript We are adding native support for new clusters and frameworks, such as LSF, Mesos, Kubertes as well as a *"Generic cluster"* API which allows the user to customize BigDataScript for any cluster or framework by encapsulating task management via user defined scripts. On the language specification side, we are exploring ways to add functional constructs such as `map`, `apply`, `filter` as well as support for *map/reduce* and *scatter/gather* which are convenient ways to define some problems in data pipeline programming. Finally we, will incorporate user-defined data structures or a basic class mechanism (BDS currently supports maps and list).

Variant annotations In an effort coordinated with the developers of other annotations tools (such as ANNOVAR [3], ENSEMBLs Variant effect predictor -VEP- [2], JAnnovar [1], etc.) we are creating new annotation standard for VCF files. We are actively collaborating

with the *"Global Alliance for Genomics and Health"* (GA4GH) in the creation of variant annotation specification & API definitions. We plan to extend SnpEff's variant annotation capabilities to *haplotype-based* annotations, which means taking into account phasing information to calculate compound variant effects (e.g. phased SNPs affecting the same codon or compensating frame shifts within the same DNA strand). Finally, we are using information-theoretic analysis of splice sites from several species in order to improve splicing effect predictions.

GWAS Epistasis As future work, we'd like to evaluate the possibility of incorporating contextual information, such as protein domain, in order to build more specific co-evolutionary models. Other improvements include further optimization of logistic regression and Bayes factor algorithms since any improvement greatly reduces computational times. We also plan to use our methods on even larger type II diabetes cohorts that are currently being sequenced. Finally, we are evaluating the possibility of incorporating higher order interactions by clustering genes from our variant-pairs analysis and then evaluate them in a joint analysis.

1.3 Perspectives

Genomic research for complex disease is trending towards larger and larger cohorts in order to improve statistical power. Some years ago, projects involving hundreds to a thousand individuals were common. To put this in perspective, that is the population of a village, or a small town. Nowadays, projects like the those lead by the T2D consortia sequence in the order of $20,000$ people (i.e. the population of a large town). I am aware, through personal communications with other researchers, that projects are being drafted for sequencing over $100,000$ individuals (i.e. the population of a small city) and some institutions are foreseeing sequencing up to $1,000,000$ samples per year within the next few years.

As a rule of the thumb, sequence data of a single whole genome requires 800 CPU

hours of primary processing (i.e. read mapping and variant calling). For an institution willing to process $1,000,000$ genomes per year this means $800,000,000$ CPU hours just for primary processing. In order to keep up with sequencing, data analysis should be also performed within the same time-frame, thus requiring $\sim 92,000$ CPUs burning data 7×24. An over optimistic estimate that assumes no hardware failures, no software failures and no programmed outages. Having tens to hundreds of thousands of CPUs constantly analysing data in production environments poses infrastructures challenges. Most academic environments currently use their own infrastructure (local clusters), an approach that may not be easy to further scale. For this reason a shift towards cloud infrastructure is already being considered and in some cases implemented by some leading institutions (personal communications).

We developed BDS to help processing not only large dataset currently available, but also the huge datasets that experts consider likely to become available in a not so distant future. Even though BDS can currently handle a typical analysis involving thousands of CPUs, scaling further from hundreds of thousands to millions of CPUs would require additional abstraction levels. Most notably, the current processing model assumes the existence of a file system, which is used primarily for storing data and logging. Under a cloud based environment of $100K$ to $1M$ CPUs, this model is likely to break down and further abstraction would be required. For instance, cloud based environments use the concept of buckets instead of files and this should be abstracted away and unified for the user to be able to write more transparent and portable pipelines.

In the context of these large studies, variant annotation and selection for further study would also require some improvements. As opposed to the previous problem of processing large datasets, variant annotations is challenging not because of the computational challenges, but rather due to restricted biological knowledge. There are obvious needs for better predictions, but more precise models could be developed with help of some systematic variant studies. For instance, a systematic analysis of loss of function and nonsense mediated

decay variants would entail creating all possible stop gained mutations in one or more genes and analysing the protein output in each case (obviously this is an ambitious and challenging project to say the least, but so were other projects like 1KG, EST, GTeX and ENCODE, just to mention a few). Lower impact variants, such as non-synonymous, pose even further challenges since there is no consensus on how to measure 'partial protein gain/loss of function' (e.g. in protein affected by a non-synonymous variant, interaction with protein X is degrades by 50% whereas interaction with protein Y is improved 20%). Such analysis, which are beyond the current state of technology, could only be feasible by supporting long term technology development projects.

Finally, [clinical variant effect prediction going mainsteram] [silo effect with every research company creating their own "curated database"] [ClinGen is a good perspective]

[missing heritability] This quest for ever bigger sample sizes shows how elusive the genetic causes of complex diseases are. It might be true that huge sample sizes are needed to uncover risk loci, but perhaps one of the reasons why traditional GWAS studies are not finding as many associations as expected is just that we they are looking at the wrong place by not taking into account other possibilities, such as epistasis.

References

[1] Marten Jäger, Kai Wang, Sebastian Bauer, Damian Smedley, Peter Krawitz, and Peter N Robinson. Jannovar: a java library for exome annotation. *Human mutation*, 35(5):548–555, 2014.

[2] William McLaren, Bethan Pritchard, Daniel Rios, Yuan Chen, Paul Flicek, and Fiona Cunningham. Deriving the consequences of genomic variants with the ensembl api and snp effect predictor. *Bioinformatics*, 26(16):2069–2070, 2010.

[3] K. Wang, M. Li, and H. Hakonarson. Annovar: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic acids research*, 38(16):e164–e164, 2010.