Annotating Regulatory Variants With `Fig`

Frank Austin Nothaft

Abstract

Although large scale sequencing experiments such as the 1,000 Genomes Project have given us a good understanding of the distribution of variants in the human population, many questions about these variants still remain. Many questions surround the function of variants seen in the non-coding regions of the genome. While the "grammar" used to turn coding sequence into proteins is well understood, the effects of non-coding variants can currently only be assessed through statistical measures such as GWAS and eQTL studies.

In this paper, we introduce `Fig`, a tool that **F**inds **I**nteresting **G**enotypes by annotating non-coding variants. `Fig` uses a proposed grammar, as well as known transcription factor binding sites to annotate haplotypes of variants. We implement `Fig` using the `ADAM` API, and evaluate variant calls from the third phase of the 1,000 Genomes project.

1 Introduction

The recent $10,000\times$ drop in the cost of genome sequencing has enabled population scale DNA and RNA sequencing experiments. Experiments such as the 1,000 Genome Project [1] and The Cancer Genome Atlas [14] have allowed us to analyze patterns of variation across large cohorts, which has led to an improved understanding of both population-level variance, and the relationship between genetic variation and diseases like cancer.

Although these experiments have provided us with a large set of data about human variation, several large questions remain. One large question pertains to the role of variation that occurs outside of the coding portions of the human genome. Although projects such as ENCODE [5] have profiled the interaction of regulatory elements with genomic sequence, these projects merely provide a glimpse into the interaction between sequence and regulatory effects. While there is an emerging literature that is studying the "grammatical architecture" of regulatory regions [8, 16], this literature largely depends on synthetic experiments [13] to drive inference. While the results from these synthetic experiments are valuable, and will improve our ability to model the impact of variants on regulation, we instead seek to ask if there is a way for us to understand this relationship from existing variation datasets?

In this work, we present `Fig`, a tool that applies "grammar" to annotate the effect of variants called near regulatory sites. In the remainder of this paper, we present summary results from looking at phase 3 of the 1,000 Genomes project, and explain our methods. As this is preliminary work, we focus on the opportunities for extending this work. Although we have not tackled this problem in this work, we believe that this technique will prove useful for filtering out the importance of individual variants that occur in linkage disequilibrium (LD) blocks, which complicate studies that look to statistically derive relationships between variation and expression (expression quantitative trait loci tests, eQTL) or phenotypes (genome wide association studies, GWAS).

2 Results

We ran `Fig` on phased genotype data from phase 3 of the 1,000 Genomes project. To annotate transcription factor binding sites (TFBS), we used data from the ENCODE project that was compiled by Kheradpour and Manolis [6]. For reproducibility, we have made the scripts used to run this experiment available from `https://https://github.com/fnothaft/fig/blob/run-scripts/fig_analysis.sh`. We annotated all regions that were 0 to 1,500 base pairs (bp) ahead of the transcription start site for all annotated genes[1].

First, we looked at the number of variants that occurred in the annotated regions. This is depicted in Figure 1. Interestingly enough, although most annotated regions do contain a variant, approximately 43% of annotated regions do not contain variants. This is lower than expected, as the variation rate in the human genome is 0.1%[2] Under a Binomial distribution with $p = 0.001$ and $n = 1500$, we expect only 22% of regions to not have variants.

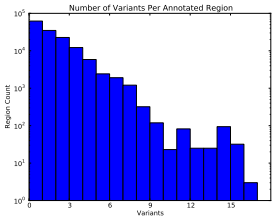

Figure 1: Number of variants per region

After looking at this, we looked at the number of sites that were modified. This data is shown in Figure 2. We looked at the number of sites that were lost, or the number of sites that were modified. We found that very few sites were lost. The vast majority of regions did not lose a single TFBS, and the most TFBS lost in a single region was 4. We would like to examine this closer to identify the characteristics of regions that lost multiple sites. For example, this effect could occur because of clustered variation (such as a multiple nucleotide polymorphism or medium size INDEL), or because of a variant at a site that is overlapped by multiple TFBS.

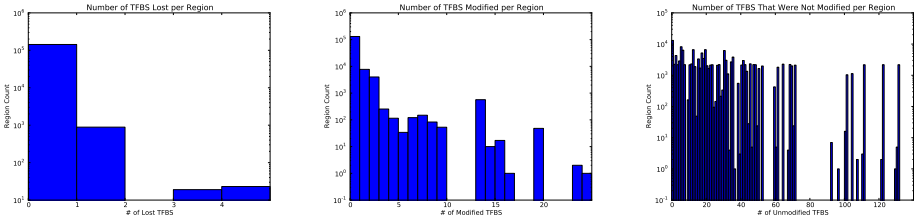

Figure 2: Number of TFBS lost, modified, and unmodified over all regions

Additionally, we would like to do a further review of the modified sites. Specifically, we would

[1]We used the GRCh37 gene annotations available from `ftp://ftp.ensembl.org/pub/release-75/gtf/homo_sapiens/`.

[2]This rate actually underestimates the rate of variation in a *single* individual. Specifically, studies such as the 1,000 Genomes project have identified that approximately one "common" variant (present in $> 5\%$ of the population) appears per 1,000bp.

like to look at the degree of modification of these sites, as measured by the predicted binding affinity of the unmodified site relative to the modified site.

Finally, we looked at changes in spacing between TFBS. These spacing modifications are caused by INDEL variants. Here, we measure the number of potential TF interactions that could be gained/lost by a spacing change, under the assumption that two TFs can interact if they are separated by an interval that is a multiple of 5 bp. This is shown in Figure 3. Approximately 89% of annotated regions did not see any modified spacing. In the annotated regions that did see modified spacing, there was no clear distribution of effects. However, it is interesting to note that the change affects are biased towards *increased* interactions.

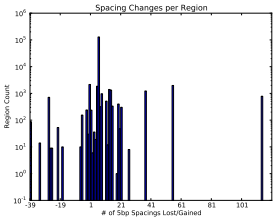

Figure 3: Spacing changes over regions

There are several additional analyses we would like to incorporate here. Currently, we do not check to see if two TFs with 5 bp spacing are predicted to interact. This could be estimated using ENCODE data [5]. Additionally, there are odd gaps/spikes in the spacing changes. We would like to review the data further to see if there are consistent spacing change motifs.

Additionally, we measured the GC ratio change across regions. Although we did not compute a change histogram, the average GC ratio change was less than 0.0001%. This makes sense, as the GC ratio of 1,500 bp regions will be stable unless we see very large variants.

3 Methods

`Fig` is implemented using the `ADAM` API [9, 11]. `ADAM` is a set of data formats and functional operators for processing genomic datasets on a distributed computing farm, and is built on top of Apache `Spark` [18, 19]. By building on top of `ADAM`, `Fig` is able to rapidly process large datasets. `Fig` is open source software under an Apache 2 license and is available from `https://www.github.com/fnothaft/fig`. `Fig` is decomposed into several steps:

1. Per region to annotate, extract the reference sequence of the region. We annotate the reference sequences with TFBS by running a broadcast region join[3] of the TFBS features against the reference sequences. The binding affinities of all TFBS are scored using a user-provided position weight matrix (PWM) per motif.

2. We then compute the "variant" regions by loading in the genotypes for all samples, and running a region join against the reference regions. This effectively labels all genotypes

[3] A region join joins tuples from two datasets, where the tuples in each dataset are keyed by the genomic region that the value in the tuple covers. *Broadcast* implies a specific execution strategy for the join; specifically, the tuples from the lower cardinality dataset are broadcast to all of the nodes on the cluster. For more details, see §5.1 of Nothaft et al [11].

that are in a region that is to be annotated with the gene ID. From here, we then group all genotypes together that are from the same sample and that are located in the region assocated with a single gene ID.

3. Once genotypes are collected by gene and sample, we flatten the variants associated with these genotypes into haplotypes[4]. The TFBS on these haplotypes are then annotated.

We use a simple algorithm for flattening variants into haplotypes. Starting from genotypes, we look at the phased genotype states for all genotypes. We do this by looping over the haplotypes that we are computing. If the genotype call of a genotype on a specific strand is reference, we discard the variant associated with that genotype. We the sort the variants associated with a strand by position. We then run a recursive algorithm that loops over the variants. At each variant, the reference sequence is replaced by the variant sequence. If the variant is an INDEL, we track the difference in length between the reference and variant sequences. This difference is important, as we need it for tracking changes in spacing between TFBS pairs.

We annotate haplotypes using the grammar cards reviewed by Weingarten-Gabbay and Segal [16]. Currently, we support the following annotations:

- We predict **changed affinity** by recomputing the TFBS sequence affinity using the PWM for each TF and the variant and reference sequences. The change in affinity is expressed as a ratio.
- We predict the **loss of a binding site** when the predicted affinity of a site is reduced to 0 by a variant that modifies the TFBS sequence.
- We track the **relative distance** between two TFBS. The TFBS spacing can be impacted by the presence of INDEL variants. Currently, we track the total number of TFBS that have spacing that is a multiple of a 5 bp distance, as 5 bp distances lead to peaks in expression.
- We compute the GC ratio for the annotated sequences, as a proxy for **local sequence context**. This is expressed as a change ratio between the reference and variant sequence.

Eventually, we plan to incorporate these annotations into a GWAS or eQTL pipeline. At the current moment, we solely use the annotations to calculate a variety of rollups. Specifically, we calculate the average number of modifications across the dataset, the average modifications per sample, and the value distributions of each metric across the whole dataset.

As mentioned above, we built `Fig` using the `ADAM` API so that we could make use of distributed computing to accelerate our analyses. We launched a 32 node cluster on Amazon's Elastic Compute 2 (EC2) farm to run our analysis of the 1,000 Genomes dataset. The majority of the runtime (approximately 1.5 hours out of a two hour job) is spent executing the region join of genotypes against regions to be annotated. This is to be expected since the region join discards all genotypes that are outside of the regions to be annotated, which leads to an approximately 99% reduction in dataset size. For efficiency, we have pre-converted the 1,000 Genomes genotype collection from Variant Call Format (`VCF`, [3]) into `ADAM`'s genotype representation. This data is stored in the `eggo` repository [4] and is accessible at `s3://bdg-eggo/1kg/genotypes`. Preconverting to `ADAM` provides several advantages: `ADAM`'s genotype format is 66% smaller than compressed `VCF` and is more efficient to parse, as it is a binary format.

[4]Currently, we only support the processing of phased genotypes. If the genotypes are unphased, we cannot associate variants to a specific haplotype. In future work, we hope to add support for unphased genotypes, under some form of pooled/graph model. This would predict *all possible* modifications. This pooled model is necessary for processing somatic variant calls, which is a problem we are interested in.

4 Future Work

In its current state, `Fig` is capable of annotating regulatory regions with possible modifications. While this is conceptually useful, `Fig`'s output is not meaningfully interpretable. To make `Fig` a useful tool, we propose extending it to integrate in with a GWAS/eQTL pipeline. This approach has been proposed previously by Levo and Segal [8]. This approach could be implemented in multiple forms:

1. A simplistic annotator could be used to "filter" out variants that were predicted *not* to have an impact on functional regulation. This simplistic annotator would not necessarily improve the results of a GWAS or eQTL run, but would be useful for eliminating variants that segregate inside of a LD block that were unlikely to be impactful.

2. A tighter integration of the annotation and eQTL engines could combine the "grammar card" annotations emitted by `Fig` with the modifications in expression trends reviewed by Weingarten-Gabbay and Segal [16]. While this *could* lead to better eQTL results, some effects are hard to model (e.g., weak binding affinities). Additionally, it is not obvious as to how a similar integration would work for annotation and GWAS.

Additionally, `Fig` is not a complete annotator. If being used in conjunction with eQTL/GWAS tools, or in a traditional variant calling and annotation pipeline (e.g., the `GATK` Best Practices Pipeline [2] or `HugeSeq` [7]), it would be useful to annotate coding variants, *a la* `Annovar` [15] or `VEP` [10]. Additionally, it would be useful to extend `Fig` to annotate more of the functional effect variant cards. Several cards are not annotated right now (neither nucleosome positioning nor mutual interaction between TFs are predicted, nor do we predict the *gain* of TFBS due to variants) and would be useful to integrate.

Beyond the fact that `Fig` is designed for annotating non-coding variants, `Fig` differs from traditional variant annotators in that it does not annotate individual variants, but rather haplotypes. While the approach used by `Fig` (specifically, the flattening of variants into haplotypes) works for small datasets, it is likely that a graph-based approach (see Novak et al [12]) would be necessary for annotating larger contiguous regions. While a graph-based approach doesn't reduce the complexity of annotating a longer contiguous region, a graph-based approach makes it easier to construct the haplotype that describes the region. For example, in a string-based haplotype annotator, all variants on a single phased branch of a chromosome would need to be collected together and integrated serially into the haplotype. In a graph-based haplotype annotator, individual variants modify a limited portion of the overall graph.

As mentioned earlier, a future aim for this project is to add support for somatic callsets under a pooled annotation model. We originally were targeting The Cancer Genome Atlas [17], but refocused on the 1,000 Genomes project due to issues surrounding data access and quality. Long term, our interest is in using the annotation of regulatory variants to look at diseases such as Acute Myeloid Leukemia (AML), which are known to have very few coding variants [14].

5 Conclusion

In this paper, we presented `Fig`, a haplotype-based tool for annotating variants that overlap with regulatory regions. `Fig` is built on top of the `ADAM` API, and is designed for processing very large datasets. We have demonstrated `Fig` on the genotype calls from the 1,000 Genomes dataset, and

have plotted future enhancements for the `Fig` project, which include integration with GWAS/eQTL tools.

References

[1] 1000 Genomes Project Consortium et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, 2012.

[2] G. A. Auwera, M. O. Carneiro, C. Hartl, R. Poplin, G. del Angel, A. Levy-Moonshine, T. Jordan, K. Shakir, D. Roazen, J. Thibault, et al. From FastQ data to high-confidence variant calls: The Genome Analysis Toolkit best practices pipeline. *Current Protocols in Bioinformatics*, pages 11–10, 2013.

[3] P. Danecek, A. Auton, G. Abecasis, C. A. Albers, E. Banks, M. A. DePristo, R. E. Handsaker, G. Lunter, G. T. Marth, S. T. Sherry, et al. The variant call format and VCFtools. *Bioinformatics*, 27(15):2156–2158, 2011.

[4] B. D. Genomics. eggo. `https://www.github.com/bigdatagenomics/eggo`.

[5] M. B. Gerstein, A. Kundaje, M. Hariharan, S. G. Landt, K.-K. Yan, C. Cheng, X. J. Mu, E. Khurana, J. Rozowsky, R. Alexander, et al. Architecture of the human regulatory network derived from ENCODE data. *Nature*, 489(7414):91–100, 2012.

[6] P. Kheradpour and M. Kellis. Systematic discovery and characterization of regulatory motifs in ENCODE TF binding experiments. *Nucleic acids research*, 42(5):2976–2987, 2014.

[7] H. Y. Lam, C. Pan, M. J. Clark, P. Lacroute, R. Chen, R. Haraksingh, M. O'Huallachain, M. B. Gerstein, J. M. Kidd, C. D. Bustamante, et al. Detecting and annotating genetic variations using the HugeSeq pipeline. *Nature biotechnology*, 30(3):226–229, 2012.

[8] M. Levo and E. Segal. In pursuit of design principles of regulatory sequences. *Nature Reviews Genetics*, 15(7):453–468, 2014.

[9] M. Massie, F. Nothaft, C. Hartl, C. Kozanitis, A. Schumacher, A. D. Joseph, and D. A. Patterson. ADAM: Genomics formats and processing patterns for cloud scale computing. Technical report, UCB/EECS-2013-207, EECS Department, University of California, Berkeley, 2013.

[10] W. McLaren, B. Pritchard, D. Rios, Y. Chen, P. Flicek, and F. Cunningham. Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor. *Bioinformatics*, 26(16):2069–2070, 2010.

[11] F. A. Nothaft, M. Massie, T. Danford, Z. Zhang, U. Laserson, C. Yeksigian, J. Kottalam, A. Ahuja, J. Hammerbacher, M. Linderman, M. J. Franklin, A. D. Joseph, and D. A. Patterson. Rethinking data-intensive science using scalable analytics systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '15)*, 2015.

[12] A. Novak, Y. Rosen, D. Haussler, and B. Paten. Canonical, stable, general mapping using context schemes. *arXiv preprint arXiv:1501.04128*, 2015.

[13] E. Sharon, Y. Kalma, A. Sharp, T. Raveh-Sadka, M. Levo, D. Zeevi, L. Keren, Z. Yakhini, A. Weinberger, and E. Segal. Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters. *Nature biotechnology*, 30(6):521–530, 2012.

[14] The Cancer Genome Atlas Research Network et al. Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. *The New England journal of medicine*, 368(22):2059, 2013.

[15] 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.

[16] S. Weingarten-Gabbay and E. Segal. The grammar of transcriptional regulation. *Human genetics*, 133(6):701–711, 2014.

[17] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, Cancer Genome Atlas Research Network, et al. The Cancer Genome Atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113–1120, 2013.

[18] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI '12)*, page 2. USENIX Association, 2012.

[19] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: cluster computing with working sets. In *Proceedings of the USENIX Conference on Hot Topics in Cloud Computing (HotCloud '10)*, page 10, 2010.