Thesis Proposal: Scalable Systems and Algorithms for Genomic Variant Analysis

Frank Austin Nothaft

Abstract

Since 2001, improvements in DNA sequencing technologies have allowed the cost of sequencing a single human genome to drop from $1 billion to under $1,000. This drastic change in the economics of DNA sequencing has enabled both the use of sequencing in clinical medicine and the genomic study of large cohorts. However, this shift brings new problems: the cost of analyzing genomic data is growing more quickly than the cost of collecting genomic data is falling. This trend is occurring because the decline in the cost of sequencing is outpacing Moore's law.

Genomics is experiencing quintessential "big data" problems. To uncover the link between genomic variation and traits, we need to statistically analyze large cohorts. Because each genome represents $\mathcal{O}(1-100\text{GB})$ of data, moderately sized cohorts like the 1,000 Genomes project (which contains more than 70TB of data) present significant data storage and processing challenges. By collecting and processing this data, we are able to uncover statistical linkages between genomic variation and diseases. However, these statistical correlations do not always provide biological insight: in some cancers, such as Acute Myeloid Leukemia (AML), very few genes are modified by mutations. In diseases like AML, we need to make use of both statistical links and novel genomic annotation techniques to understand the underlying disease process.

This problem requires a two pronged approach that tackles both infrastructural and algorithmic problems. We have developed the ADAM system, which provides an efficient API for distributing genomic analyses across many nodes. On top of ADAM, we are building Avocado, a distributed variant caller that introduces efficient algorithms for identifing genomic variants. In addition to having lower runtime complexity than prior variant detection algorithms, the core algorithm in Avocado computes canonical representations of variation, which simplifies downstream statistical analyses. To enable very large scale genotype-phenotype association queries, we are building Gnocchi, a map-reduce based statistical query engine. Finally, we are developing a novel algorithm, Fig, which can be used to identify variants that modify the effect of regulatory protein binding sites in the genome. By understanding how genome variation can impact transcriptional regulation, we hope to better understand complex diseases like AML.

1 Thesis Statement

Although all cancers are defined by a set of mutations that distrupt the normal function of the genome, these mutations often occur at low frequencies. To understand the biology behind these low frequency mutations, we need to run large scale ($\mathcal{O}(> 100,000)$ individuals) statistical analyses. Although the cost of genome sequencing has plummeted in the last fifteen years, the analysis of cohorts of this size is limited by the computational cost of analysing large quantities of genomic data. To make these analyses tractible, we must enable the use of novel parallel computing technology, and development more efficient genome analysis algorithms. By coupling these improved analyses with advanced genomic annotation algorithms, we can better understand the role of variants in non-coding regions of the genome, and their impact on diseases like AML.

2 Background

Since the completion of the Human Genome Project in 2003, genome sequencing costs have dropped by more than $10,000\times$ [26]. The rapidly declining cost of sequencing a single human genome has enabled large sequencing projects like the 1,000 Genomes Project [37] and the Cancer Genome Atlas (TCGA, [45]). As these large sequencing projects perform analysis that process terabytes to petabytes of genomic data, they have created a demand for genomic analysis tools that can efficiently process data at this scale [33, 40]. Figure 1 juxtaposes the rapid drop in the cost of sequencing a single genome against the amount of data collected by each successive generation of sequencing projects. New DNA sequencers such as the Illumina X10 provide sufficient throughput such that a single hospital or research lab can generate 750GB of genomic data per day, or 275TB of genomic data per year [13].

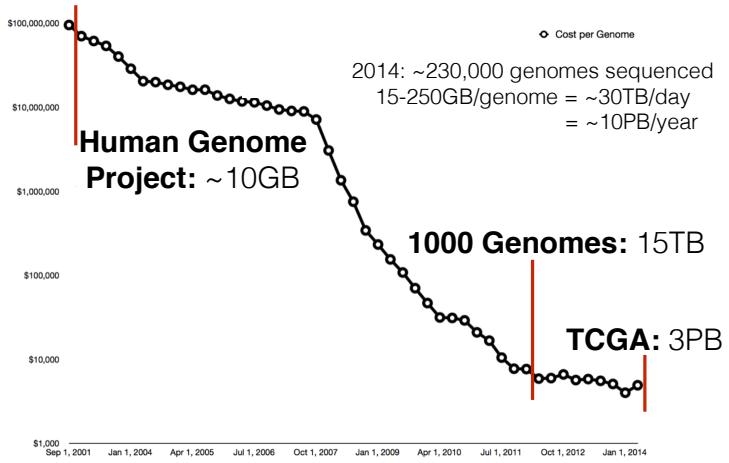

Figure 1: Genome sequencing costs over time, from NHGRI numbers [26], with data volumes of major genome sequencing projects overlaid.

Over a similar time range, commercial needs led to the development of horizontally scalable analytics systems. The development and deployment of MapReduce at Google [6, 7] spawned the development of a variety of distributed analytics tools and the Hadoop ecosystem [1]. In turn, these systems led to other systems that provided more fluent programming models and higher performance [49]. The demand for these systems has been driven by the increase in the amount of data available to analysts, and has coincided with the development of statistical systems that are accessible to non-experts, such as Scikit-learn [30] and MLI [39].

With the rapid drop in the cost of sequencing a genome, and the accompanying growth in available data, there is a good opportunity to apply modern, horizontally scalable analytics systems to genomics. New projects such as the 100K for UK, which aims to sequence the genomes of 100,000 individuals in the United Kingdom [11], and the Department of Veterans Affairs' Million Veteran project [43] will generate three to four *orders of magnitude* more data than prior projects like the 1,000 Genomes Project [37]. Additionally, periodic releases of new reference datasets such as reference genomes necessitates the periodic re-analysis of these large datasets. These projects use the current "best practice" genomic variant calling pipeline [3], which takes approximately 120 hours to process a single, high-quality human genome using a single, beefy node [41]. To address these challenges, scientists have started to apply computer systems techniques such as map-reduce [16, 22, 34] and columnar storage [10] to custom scientific compute/storage systems.

While these systems have improved analysis cost and performance, current implementations incur significant overheads imposed by the legacy formats and codebases that they use.

Additionally, although these experiments have provided us with a large set of data about human variation, several large questions remain. One question pertains to the role of variation that occurs outside of the coding portions of the human genome. Although the human genome is more than three billion bases long, the portion of the genome that codes for proteins (the exome) is only approximately 1% of the genome. Although we can accurately predict how modifications to DNA sequence inside of the exome will modify the structure and composition of proteins [23], no validated rules exist for the remaining 99% of the genome. However, 81% of the sequence outside of the exome is responsible for regulating the rate at which a gene in the exome is translated into a protein [12]. This estimate was derived by identifying the positions where transcription factors (TFs[1]) bind to the genome. As we explain in Figure 2, sequence variants that occur in regulatory regions can modify the rate at which nearby genes are transcribed and translated into proteins [18, 44].

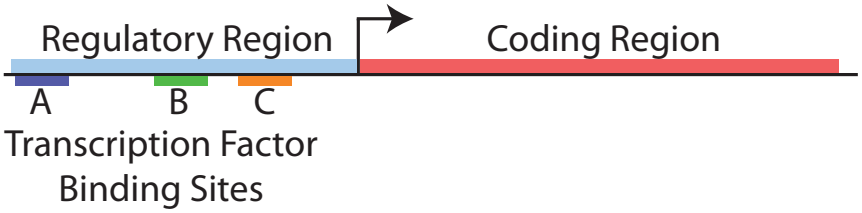

Figure 2: This figure represents the area around the start of a gene in the genome. The arrow in the figure is the transcription start site (TSS), which is the start of the coding portion of the gene. In the approximately 1,500 bases before the TSS, there is a regulatory region where transcription factors will bind to the genome. Variants that occur in the regulatory region can have a variety of effects [18, 44]. For example, a variant that occurs at TFBS A may reduce the likelihood that the TF binds. In some cases, for a TF to cause transcription to start, it must be part of a "complex" of multiple TFs. If TFBS B and C form a large protein complex, a variant that changes the distance between the two TFs (by inserting or deleting sequence) may mean that the TFs cannot form the larger complex, and may disable the translation of the DNA into RNA.

We are interested in understanding these variants so that we can understand diseases that are known to have genomic drivers, but where few variants occur in coding sections of the genome. One such disease is AML, which has eight clinically distinct subtypes [42]. Although several of the subtypes of AML are known to be caused by specific genomic variants, we do not know the drivers for the remaining subtypes of the disease. Our understanding of variation in AML is complicated by the relatively low rate of mutation seen in AML, relative to other cancers. Figure 3 depicts the long tailed distribution of mutations in AML.

While we can use the aforementioned statistical techniques to identify correlative links between these variants and the diseases we would like to study, these statistical techniques do not explain the underlying disease biology, and cannot be used to determine causation. To develop a causitive model, we build upon projects such as ENCODE [12] that have profiled the interaction of regulatory elements with genomic sequence, While there is emerging literature that studies the "grammatical architecture" of regulatory regions [18, 44], this literature largely depends on synthetic experiments [35] to drive inference. While the results from these synthetic experiments improve our ability to model the impact of variants on regulation, we instead seek to ask if there is a way for us

[1]Transcription factors are proteins whose function is to bind to DNA and control the rate at which DNA is transcribed into RNA. This RNA can then be translated into a protein.

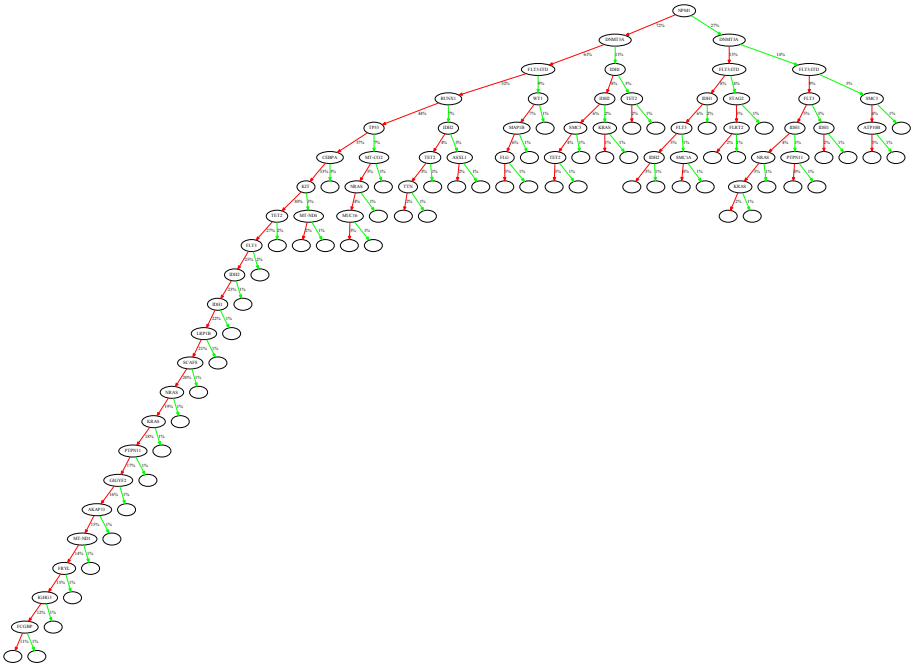

Figure 3: The nodes in this decision tree are genes known to be mutated in a patient in the LAML subset of the TCGA [42]. Each node is a gene. At each node, the left arrow indicates that the gene was *not* mutated, while the right arrow indicates that the gene was mutated. This tree is notable because it demonstrates the long tail of mutations in AML: if we trace down the nodes running down the left side of the tree, we see that 11% of AML patients have no coding mutations. This figure was generated by Ravi Pandya from MSR as part of the UC Berkeley/OHSU/MSR collaboration on the BeatAML project [28].

to understand this relationship from existing variation datasets? We believe we can answer these questions by building a variant annotation engine that incorporates knowledge of this regulatory grammar, and integrating these annotations into our correlative genomic models.

3 Work To Date

To address the immediate computational needs of sequencing analyses, we embarked on the design of the ADAM system [21, 27]. ADAM is a genomic processing system that was built using open source technologies that are designed for data intensive computing, such as Apache Parquet [2] and Spark [48, 49]. The main goal of building the ADAM system was to demonstrate that a well architected system that was built using commodity technologies could outperform optimized, genomics-specific systems, while letting bioinformaticians write genomic analyses using higher level primitives.

To make it possible to program genomic analyses against higher level abstractions, we created the decomposed stack shown in Figure 4, which was inspired by the stack models common in networking [50]. Our stack model is distinguished by the use of a schema as a "narrow waist". Conventional genomics pipelines rely on binary data formats that do not distinguish the logical structure of data from the physical representation of data. This is problematic, as legacy genomic formats such as SAM/BAM [19] use the blurring of lines between logical and physical representations to perform "optimizations" that make analyses brittle to write. In the popular Genome

Analysis Toolkit (GATK, [22]), analyses are written against the "walker" API, which presents a sorted iterator over the genome. By blurring the bounds between the logical and physical views of the data, the GATK can accelerate "walker" traversals by pushing a sort order invariant down into the data. However, as we have shown in prior work [27], this approach can lead to subtle correctness bugs. In this prior work, we demonstrate how these same algorithms can be written using common relational primitives such as aggregates and joins, and we show that this allows us to scale genomic analyses linearly to run on thousands of cores, without modifiying the analysis algorithms.

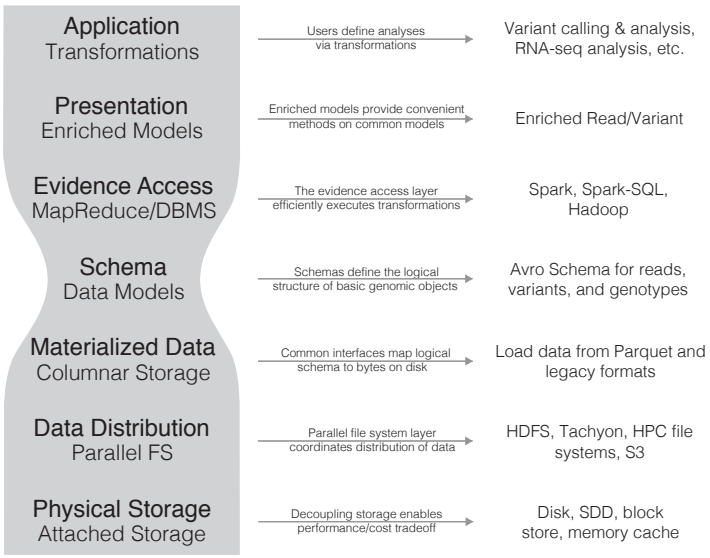

Figure 4: The stack model used in ADAM.

As a continuation of this work, we are working on a manuscript that demonstrates how ADAM can be used to improve the performance of conventional variant calling pipelines, without sacrificing accuracy. Figure 5 demonstrates the structure of a genomic variant analysis pipeline; in this manuscript, we are focusing on the read preprocessing and variant likelihood estimation stages. We have evaluated a hybrid variant calling pipeline that uses ADAM, along with the GATK's HaplotypeCaller [8]. We have compared this to the GATK's "Best Practices" pipeline [3], and have been able to achieve a 3.5× latency improvement while improving cost by 2×. The two pipelines generate statistically equivalent variant calls. As part of the manuscript, we will demonstrate how ADAM makes it less expensive to process large genomic datasets by reanalyzing the 50TB Simons Genome Diversity Dataset [36].

We have been developing several tools that run on top of the ADAM processing framework. One of these tools is the Avocado variant caller, which we aim to use to replace the GATK. Avocado is a fully distributed variant caller, which calls variants using a local reassembly strategy. Variant calling relies on the fact that the genomes of samples in a species are more similar than dissimilar; if we did a pairwise comparison of any two human genomes, we would expect the two genomes to have the same sequence at 99.9% of locations. To identify variants, we align DNA sequence fragments to the reference genome[2] and then run a statistical algorithm to identify sequence edits. We are in the process of preparing a manuscript on the novel local reassembly algorithm used by Av-

[2]A reference genome represents the "average" genome for a species. The Human Genome Project [15] generated the first human reference genome from a pool of twenty individuals.

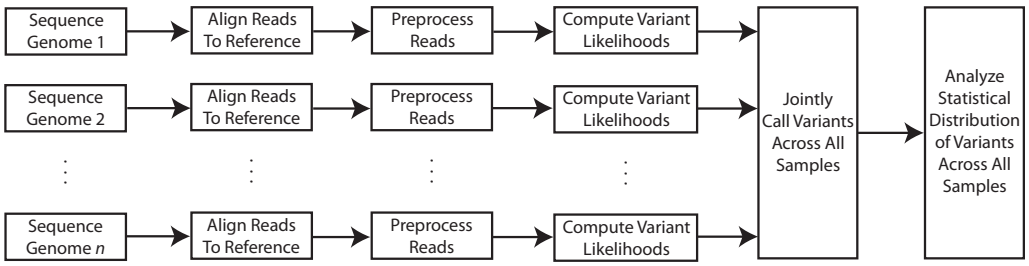

Figure 5: A general pipeline architecture for analyzing genomic data across a cohort of n individuals. ADAM performs the read preprocessing stage, Avocado performs the variant likelihood estimation and joint variant calling stages, and Gnocchi and Fig are used to analyze the variants found across the cohort. For short read alignment, we use the SNAP aligner [47], which was previously developed as a collaboration between UC Berkeley and MSR.

ocado. Conventional variant callers (such as the GATK's HaplotypeCaller, Platypus [32], and Scalpel [25]) make use of local realignment or reassembly to identify variants. These algorithms use the sequenced reads to generate possible haplotypes[3] that contain the variants from this region. To identify the haplotypes that are most likely to exist, we then and then score the pairwise compatibility of each read aligned to this region versus each potential haplotype using a string edit distance model. These algorithms achieve poor throughput because of the existance of a $\mathcal{O}(n^2)$ stage where all haplotype-read pairs are scored. In Avocado, we exploit the structure of the de Bruijn graph that is assembled from the reads at the locus to achieve best case $\mathcal{O}(n)$ performance. Additionally, our algorithm provides provably canonical representations of insertion and deletion variants, which simplifies the joint analysis of multiple samples. This canonical representation is important for insertion/deletion (INDEL) variants, as traditional hidden Markov model [9] or dynamic programming approaches [38] can generate multiple edit representations for a single INDEL variant [29].

4 Proposed Work

In the remainder of my thesis work, I plan to tackle three problems:

- How can we use the efficient algorithms in ADAM and Avocado to jointly analyze thousands of samples?
- How can we parallelize the statistical analysis of genotype-phenotype associations on large cohorts?
- Can we use genomic annotations to better understand the impact of low frequency variation on transcriptional regulation?

As mentioned earlier, we have already demonstrated the potential of ADAM for acheving order-of-magnitude improvements in throughput when processing a single sample, and we have shown how we can use the efficient algorithms in Avocado to achieve faster region reassembly. Currently,

[3]In this setting, a haplotype represents the local sequence of a single chromosome. "Local" implies that this sequence is a substring of the full chromosome: for variant calling, typical haplotype lengths range from 1,000 to 5,000 bases.

memory capacity backpressure limits the number of genomes we can process given a fixed amount of hardware. This is caused by metadata that is replicated when we translate data between PARQUET's columnar format on disk and SPARK's row oriented format in memory. However, we have recently developed a hybrid normalized columnar scheme that reduces our memory consumption by 11×.

Along with this, we are working on several other parallel data translation problems. Specifically, while we introduced optimized join algorithms for finding overlapping genomic regions in the latest ADAM paper [27], we are working on a further optimizations to these join algorithms for joint variant calling. When jointly analyzing multiple genomes, we perform a sweeping pass over the genome in coordinate space. Although this approach is implicitly serial, we are working to express this algorithm as an adaptive self join. Figure 6 provides a graphical explanation of the join algorithm's function: the adaptive portion of this join selects the granularity used for grouping base observations. In areas of the genome where there are no insertion or deletion variants, we work at single base pair granularity. This granularity expands to match the size of the largest deletion whenever one is present. By using this self join instead of a naïve algorithm that divides the genome into fixed size chunks, we will be able to increase our parallelism while reducing the instantaneous amount of data that we materialize into memory and preserving the ability to correctly call deletion variants.

Figure 6: This figure depicts an example 12 base long section of the reference genome, which is covered by five reads, where each read is eight bases long. The reference genome sequence is represented by the top horizontal line, while the reads are represented by the other five horizontal lines. In this region of the genome, we have evidence of two variants: reads two and three show a `CTC` → `C` deletion at positions 4–6, and reads four and five contain a `T` → `A` substitution at position 9. The vertical lines represent the splits chosen by our adaptive join algorithm. The adaptivity in the join is necessary to successfully identify the two-base deletion. If we had naïvely decided to use a one base window, we would split the deletion into three separate alleles: position 4 would have not been recorded as a variant, while positions 5 and 6 would be independently recorded as separate one base deletions. Although these two edit representations appear to be congruent, the second representation is undesirable as it obfuscates the underlying biology of the variant.

Although tools such as PLINK [5, 31] can already be used to run genotype-phenotype analyses, these tools have largely been designed to work with legacy genetic data. The data generated by modest sized sequencing experiments (such as the 1,000 Genomes project, which has 1.8TB of genotypic data) has grown to the point where the data is too large to fit in memory on a single machine. To work around this, these tools rely on ad hoc parallelization methods [5], such as parallel script dispatch. These ad hoc methods are both inefficient and prone to machine failures, and cannot make use of efficient distributed file systems, like HDFS. The I/O patterns of genotype-phenotype analyses that are run on genotypes generated by next generation sequencing workflows are particularly inefficient, as these queries only touch select fields from the genotype data, but legacy tools must read the entirety of each VCF genotype record. By building the GNOCCHI

genotype-phenotype correlation engine on top of ADAM, we will be able to make use of our columnar file format to eliminate the I/O bottleneck present in current genotype-phenotype tools. This will also enable the efficient parallelization of genotype-phenotype analyses across hundreds of machines. To date, we have implemented a case-control model for use in parallel genome wide association studies (GWAS), which we are in the process of validating. We plan to extend Gnocchi to support a more general set of statistical models.

Additionally, with the work on genotype clustering in Gnocchi, we see a great opportunity to make a contribution to the state-of-the-art in scalable algorithms for machine learning. While recent research at the intersection of computer systems and machine learning has focused on algorithms for training a single statistical model on systems that are "tall and skinny" [24, 46], genomics exhibits neither of these characteristics. Unlike Internet scale companies, where feature vectors are small and there are many users, feature vectors for genomic datasets are comprised of all of the genotypes seen in an individual sample. For a whole genome sequencing run, this sparse vector can represent more than 1 million genotypes. Although fast algorithms are known for fitting linear mixed models [20] for learning genotype-phenotype correlations, there has not been significant research into fast clustering methods for "wide and flat" data. In preliminary work, we have found that sampling-based approaches can be used to efficiently apply parallel methods for clustering on tall and skinny datasets [4, 24] to this wide and flat data, but at the loss of some accuracy. We will improve our clustering accuracy by deriving efficient parallel methods that are optimized for wide and flat data.

While genotype-phenotype analyses can generate correlative findings, these findings do not necessarily shed light on the underlying biology that drives these correlations. This is particularly troublesome for the non-coding regions of the genome, where traditional variant effect annotation methods that look for impacts on protein composition cannot be used [23]. We have done some preliminary investigations to see how variation impacts the regulatory regions upstream of the transcription start site (TSS) of a gene. We implemented this as the Fig tool, an ADAM-based engine for annotating genomic variants. In this preliminary experiment, we searched for variants from phase 3 of the 1,000 Genomes project that modified transcription factor binding affinity or spacing. Since TFBS spacing requires knowledge about whether two TFBS are on the same strand of DNA, we made use of the phasing from the 1,000 Genomes project to annotate phased haplotypes. To annotate TFBSs, we used data from the ENCODE project that was compiled by Kheradpour and Manolis [14] and the GRCh37 gene annotations.

Our preliminary experiments showed several interesting results. First, we found that the regulatory regions upstream of the TSS have a lower rate of variation than expected. Fifty seven percent of the annotated regions contained variants, which is lower than expected as the rate of common variation in the human genome is 0.1%[4]. Under a Binomial distribution with $p = 0.001$ and $n = 1500$, we expect 78% of regions to have variants. This implies that these regions are evolutionarily conserved. However, we did see several regions where more than three TFBSs were lost, which warrants further investigation. Additionally, we found that these variants were unlikely to modify the binding affinity of TFBSs. Binding compatibility was lost at fewer than 1% of sites, and binding affinity was decreased at approximately 10% of sites. Figure 7 depicts the distributions discussed above.

Ultimately, we would like to unify the annotations generated by Fig with the associations generated by Gnocchi. By applying these methods to a large dataset, such as the GEUVADIS subset of the 1,000 Genomes project [17], we will be able to verify whether the synthetic models for regulatory grammar [18, 35, 44] are applicable in vivo. Additionally, this work will serve as a

[4]This rate underestimates the rate of variation in a single individual. This estimate is derived from the 1,000 Genomes project, where one "common" variant (present in $> 5\%$ of the population) were seen per 1,000bp.

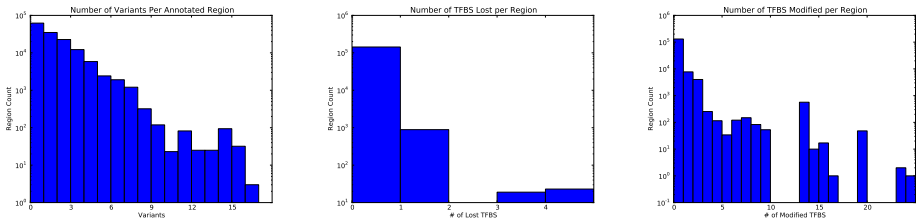

Figure 7: From left to right, these graphs show the number of variants per regulatory region, the number of TFBS where a variant rendered the TFBS sequence incompatible for binding the TF, and the number of TFBS where a variant caused the predicted binding affinity to change. These numbers were generated using genotype data from the 1,000 Genomes project [37] and TFBS annotations from Kheradpour and Kellis [14].

valuable way to calibrate the functional annotations generated by Fig. Although we have gained TFBS data from the ENCODE project [12, 14], it is unlikely to believe that all predicted TFBS have an equal impact on transcriptional regulation across all genes. By feeding this data back into the Fig annotation engine, we can better predict the effect of variants that occur in the regulatory regions. With this knowledge, we can better understand the role of the non-coding variants in diseases by being able to predict the impact of these variants on gene expression.

References

[1] Apache. Hadoop. `http://hadoop.apache.org`.

[2] Apache. Parquet. `http://parquet.apache.org`.

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

[4] Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. Scalable k-means++. *Proceedings of the VLDB Endowment 5*, 7 (2012), 622–633.

[5] Chang, C. C., Chow, C. C., Tellier, L. C., Vattikuti, S., Purcell, S. M., and Lee, J. J. Second-generation PLINK: rising to the challenge of larger and richer datasets. *GigaScience 4* (2015).

[6] Dean, J., and Ghemawat, S. MapReduce: simplified data processing on large clusters. In *Proceedings of the Symposium on Operating System Design and Implementation (OSDI '04)* (2004), ACM.

[7] Dean, J., and Ghemawat, S. MapReduce: simplified data processing on large clusters. *Communications of the ACM 51*, 1 (2008), 107–113.

[8] DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. A., del Angel, G., Rivas, M. A., Hanna, M., et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics 43*, 5 (2011), 491–498.

[9] Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge Univ Press, 1998.

[10] Fritz, M. H.-Y., Leinonen, R., Cochrane, G., and Birney, E. Efficient storage of high throughput DNA sequencing data using reference-based compression. *Genome Research 21*, 5 (2011), 734–740.

[11] Genomics England. 100,000 genomes project. `https://www.genomicsengland.co.uk/`.

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

[13] Illumina. Illumina introduces the HiSeq X^{TM} Ten sequencing system, January 2014.

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

[15] Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., et al. Initial sequencing and analysis of the human genome. *Nature 409*, 6822 (2001), 860–921.

[16] Langmead, B., Schatz, M. C., Lin, J., Pop, M., and Salzberg, S. L. Searching for SNPs with cloud computing. *Genome Biology 10*, 11 (2009), R134.

[17] Lappalainen, T., Sammeth, M., Friedländer, M. R., ACt Hoen, P., Monlong, J., Rivas, M. A., Gonzàlez-Porta, M., Kurbatova, N., Griebel, T., Ferreira, P. G., et al. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature 501*, 7468 (2013), 506–511.

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

[19] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., et al. The sequence alignment/map format and SAMtools. *Bioinformatics 25*, 16 (2009), 2078–2079.

[20] Lippert, C., Listgarten, J., Liu, Y., Kadie, C. M., Davidson, R. I., and Heckerman, D. FaST linear mixed models for genome-wide association studies. *Nature Methods 8*, 10 (2011), 833–835.

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

[22] McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M., et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research 20*, 9 (2010), 1297–1303.

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

[24] Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., et al. MLlib: Machine learning in Apache Spark. *arχiv preprint arχiv:1505.06807* (2015).

[25] Narzisi, G., O'Rawe, J. A., Iossifov, I., Fang, H., Lee, Y.-h., Wang, Z., Wu, Y., Lyon, G. J., Wigler, M., and Schatz, M. C. Accurate de novo and transmitted INDEL detection in exome-capture data using microassembly. *Nature Methods 11*, 10 (2014), 1033–1036.

[26] NHGRI. DNA sequencing costs. `http://www.genome.gov/sequencingcosts/`.

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

[28] Paten, B., Diekhans, M., Druker, B. J., Friend, S., Guinney, J., Gassner, N., Guttman, M., Kent, W. J., Mantey, P., Margolin, A. A., et al. The NIH BD2K center for big data in translational genomics. *Journal of the American Medical Informatics Association* (2015), ocv047.

[29] Paten, B., Novak, A., and Haussler, D. Mapping to a reference genome structure. *arχiv preprint arχiv:1404.5010* (2014).

[30] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research 12* (2011), 2825–2830.

[31] Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., Daly, M. J., et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics 81*, 3 (2007), 559–575.

[32] Rimmer, A., Phan, H., Mathieson, I., Iqbal, Z., Twigg, S. R., Wilkie, A. O., McVean, G., Lunter, G., WGS500 Consortium, et al. Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nature Genetics 46*, 8 (2014), 912–918.

[33] Schadt, E. E., Linderman, M. D., Sorenson, J., Lee, L., and Nolan, G. P. Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics 11*, 9 (2010), 647–657.

[34] Schatz, M. C. CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics 25*, 11 (2009), 1363–1369.

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

[36] Simons Foundation. Simons genome diversity project dataset. `https://www.simonsfoundation.org/life-sciences/simons-genome-diversity-project-dataset/`.

[37] Siva, N. 1000 genomes project. *Nature Biotechnology 26*, 3 (2008), 256–256.

[38] Smith, T. F., and Waterman, M. S. Identification of common molecular subsequences. *Journal of Molecular Biology 147*, 1 (1981), 195–197.

[39] Sparks, E. R., Talwalkar, A., Smith, V., Kottalam, J., Pan, X., Gonzalez, J., Franklin, M. J., Jordan, M. I., and Kraska, T. MLI: An API for distributed machine learning. In *Proceedings of the International Conference on Data Mining (ICDM '13)* (2013), IEEE, pp. 1187–1192.

[40] Stein, L. D., et al. The case for cloud computing in genome informatics. *Genome Biology 11*, 5 (2010), 207.

[41] Talwalkar, A., Liptrap, J., Newcomb, J., Hartl, C., Terhorst, J., Curtis, K., Bresler, M., Song, Y. S., Jordan, M. I., and Patterson, D. SMASH: A benchmarking toolkit for human genome variant calling. *Bioinformatics* (2014), btu345.

[42] The Cancer Genome Atlas Research Network and others. Genomic and epigenomic landscapes of adult de novo Acute Myeloid Leukemia. *The New England Journal of Medicine 368*, 22 (2013), 2059.

[43] U.S. Department of Veterans Affairs. Million veteran program (MVP). `http://www.research.va.gov/mvp`.

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

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

[46] Zadeh, R. B., Meng, X., Yavuz, B., Staple, A., Pu, L., Venkataraman, S., Sparks, E., Ulanov, A., and Zaharia, M. linalg: Matrix computations in Apache Spark. *arχiv preprint arχiv:1509.02256* (2015).

[47] Zaharia, M., Bolosky, W. J., Curtis, K., Fox, A., Patterson, D., Shenker, S., Stoica, I., Karp, R. M., and Sittler, T. Faster and more accurate sequence alignment with SNAP. *arχiv preprint arχiv:1111.5572* (2011).

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

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

[50] Zimmermann, H. OSI reference model–the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications 28*, 4 (1980), 425–432.