Research statement
Maxwell W Libbrecht
November 10, 2016

Machine learning plays an ever-increasing role in genomics. With the advent of sequencing-based genomics assays, we now have thousands of measurements of the biological activity at each genomic position, including where proteins bind to the DNA and what parts of the genome are transcribed. Moreover, recent innovations in fields such as single-cell measurement and genome editing promise rapid expansion of this wealth of data. As the amount of available data increases, adept analysis of these data will yield ever more biological insights.

However, several challenges must be overcome to reap the full potential of these data [4]. First, analysis must employ heterogeneous data sets that measure a variety of activities and that may harbor a variety of experimental biases. Second, machine learning models must be complex enough to be accurate while being simple enough to admit biological interpretation. Third, because the genome’s sequence is best understood using base pair-resolution analysis, such analyses must scale to billions of bases by using modern distributed computing resources.

My research focuses on the development of new machine learning methods that address these challenges. My contributions include developing novel machine learning methods, using these methods to solve computational problems in genomics, and applying these solutions to derive new insights into genome biology.

Current research
Semi-automated genome annotation

A class of methods called semi-automated genome annotation (SAGA) algorithms are widely used to model heterogeneous genomics data sets [7]. These algorithms take as input a collection of genomics data sets, performed in a given cellular condition, that measure local biochemical activity, such as the packing density of the DNA or the presence of...
proteins bound to the DNA. They simultaneously partition the genome and label each segment with an integer state index such that positions with the same label have similar patterns of activity. These methods are “semi-automated” because a human performs a functional interpretation of the states after the annotation process. SAGA methods use time-series models such as dynamic Bayesian networks or hidden Markov models. The final output of this process is an annotation of the genome, where each genomic position is labeled with a category of its biochemical activity such as “Promoter”, “Transcribed” or “Quiescent”.

I employed the SAGA framework to study genome biology in two ways. First, I produced a comprehensive reference of regulatory activity in human cells [1]. To do this, I used a SAGA method to annotate 164 human cell types and developed a machine learning-based strategy to automate the state interpretation step. I used these annotations to produce a measure of the functional importance of each genomic position permitting aggregation of information across cell types into a multi-cell type view. Using the importance measure, I combined all annotations into a single cell type-agnostic encyclopedia that catalogs all human functional regulatory elements. This resource makes it easy for a researcher or clinician to quickly and intuitively understand the importance of a genomic locus, such as in the study of disease-associated, evolutionarily conserved or positively selected loci.

Second, I investigated chromatin domains by augmenting existing genome annotation methods with a method I developed called entropic graph-based posterior regularization (EGPR; see next section) [5,6]. The regulation of a gene by a promoter directly upstream of its transcription start site is well understood, but this type of local regulation does not explain the large effect of genomic neighborhood on gene regulation. The neighborhood effect is in part the consequence of domain-scale regulation, in which regions of hundreds or thousands of kilobases known as domains are regulated as a unit. Chromatin conformation is an important part of this domain-scale regulation, so it is crucial to model this effect jointly with other measures of chromatin state. Previous SAGA models could incorporate any data set that can be represented as a vector defined linearly across the genome, but they could not incorporate information about the genome’s 3D conformation in the nucleus because this type of data is inherently pairwise. Motivated by the observation that pairs of loci close in 3D tend to occupy the same type of domain, I exploited 3D conformation information by encouraging this property to hold during genome annotation using EGPR. Through this analysis, I was able to produce a comprehensive model of chromatin domains that unified the previously-reported domain types, and I identified a new domain type characterized by clusters of tightly-regulated genes expressed in a small number of cell types, termed “specific expression domains”.

**Entropic graph-based posterior regularization (EGPR)**

I developed a machine learning method, entropic graph-based posterior regularization (EGPR), that combines probabilistic models with graph-based methods for semi-supervised learning [5,6]. Graph-based methods have been successful in solving many types of semi-supervised learning problems by optimizing a graph smoothness criterion. This criterion states that data instances nearby in a given graph are likely to have similar properties. For example, in the
case of annotation of genome domains using 3D genome conformation (see previous section), this criterion encourages positions close in 3D to receive the same domain label.

However, there has been little study of graph smoothness criteria in unsupervised models. A graph smoothness criterion cannot be directly incorporated into a generative unsupervised model because it is usually not clear what probabilistic process generated the data instances with respect to the graph, and incorporating the graph directly into a factorizable (i.e. time-series) model would break the model’s factorizable structure, making exact inference methods like belief propagation intractable. Entropic graph-based posterior regularization provides a way to express a graph smoothness criterion in a probabilistic model by defining a regularization term on an auxiliary posterior distribution variable. I found that this regularizer is effective at representing genome conformation information in a genome annotation method (previous section). Moreover, this approach would likely also be effective in other fields that involve time-series data, such as finance or natural language.

**Representative set selection using submodular optimization**

Submodular optimization, a discrete analogue to continuous convex optimization, has been used with great success in many fields, but it is not yet widely used in biology. I demonstrated its value for biological problems through two applications. First, in joint work with Kai Wei, a graduate student in the Department of Electrical Engineering, I developed a method for selecting panels of genomics assays [3]. The application of genomics assays is limited by their cost; it is typical to perform only five to ten assays on each cell type out of the hundreds of possible types of assays. Therefore, in order to derive as much information as possible, it is important to choose a diverse panel of assays to perform. However, selecting these assays is typically done in an ad hoc fashion. We showed that a method based on submodular optimization identified diverse panels that perform better in all of the common applications of these data sets than panels chosen by alternative selection strategies.

Second, I developed a method for removing redundancy in protein sequence data sets [2]. Redundancy is ubiquitous in such sequence data sets because some proteins, such as those involved in disease or industrial applications, are more heavily studied than proteins with less important or unknown function. Redundancy is typically handled by selecting a representative subset of the original data set. I found that a method based on submodular optimization finds representative subsets of sequences with greater functional divergence than those selected by the currently-used heuristics, as measured via comparison to an independent gold standard based on protein 3D structure. Therefore, we are advocating replacing the ubiquitous existing heuristic preprocessing step with this submodular optimization tool.

The use of submodular optimization for these two problems confers the advantages that (1) the resulting methods are theoretically optimal under some assumptions, and (2) the methods are easy to understand and modify because they apply generic optimization methods to optimize an intuitive objective function.

**Future plans: Unsupervised methods for learning the molecular programs that drive cellular biology**

I aim to develop three unsupervised machine learning methods that will advance our understanding of molecular biology. **Unsupervised** machine learning methods identify patterns in data in an agnostic way, without the user specifying a target to predict. Examples include clustering, principle components analysis (PCA) and the semi-automated genome annotation methods described above. Because of their unbiased nature, unsupervised methods have great potential to discover new biology mechanisms.
Aim 1: Continuous state space model of regulatory programs

Our cells implement regulatory programs the drive cellular activity, guided by the DNA sequence. For example, a given sequence may indicate that a locus will act as an enhancer in mesodermal cells. With the availability of thousands of genomics assays that measure various properties of the genome in various cell types, we have the opportunity to discover what these regulatory programs are. These genomics data sets can be represented as a matrix defined over the genome, where each column represents the activity of a given position. I aim to simultaneously learn what regulatory programs exist in the genome and which bases are regulated by which programs.

I will develop a state space model for this task. This model assigns a vector $v \in \mathbb{R}^K$ to each position in the genome, where a given value $v_k$ in this vector—called a factor—could indicate, for example, the degree to which this position acts as an enhancer in mesodermal cells. Fast dynamic programming optimization algorithms designed for state space models permit them to take into account spatial relationships between functional elements while maintaining efficient genome-wide inference. For example, such a model can represent the fact that, when a promoter is activated in a given set of tissues, it will drive expression of a neighboring gene in the same tissues. The choice of $K$ represents a trade-off: a lower $K$ results in easier interpretation, while a higher $K$ results in a more detailed representation. This type of model is conceptually similar to document topic models, which are widely used for processing text documents.

The fact that a state space model uses a continuous representation means that it is much more flexible than the existing discrete models that are most frequently used for genome interpretation. These existing discrete models cannot easily handle varying strengths of elements or mixtures of element types, such as when a position is transcribed in one cell type but acts as an enhancer in another. More specifically, to retain the same level of reconstruction as the number of input data sets grows, a continuous state space model requires that the number of factors grows linearly; for a discrete model the number of states must grow exponentially.

Such a representation will both implicate new types of regulatory programs and uncover the function of individual loci. First, each learned factor represents a unique regulatory program; I will compare these factors to existing annotations in order to discover novel programs. I will aim to collaborate with experimental biologists to validate these novel programs. Second, the learned factors form a compressed representation of the activity of a given position across all measured cell types. This representation will enable easy understanding of, for example, disease variants or evolutionary adaptations. I will aim to find DNA sequence patterns that explain the learned factors, indicating that these patterns drive regulation.

Aim 2: Learning the biochemical program that drives 3D genome structure

The genome’s sequence drives its 3D structure. That means there is some biochemical program, implemented by transcription factors in the nucleus, that reads the sequence and outputs the 3D structure. I aim to simultaneously (1) discover the molecular program that produces this 3D structure, and (2) discover the genomic features most important for driving this program.

I propose to learn a computer program that simulates the structure-generating biochemical program. The genome’s
3D structure can be experimentally measured using the Hi-C assay, which outputs a matrix of pairwise contact frequencies between each pair of genomic positions. The desired structure-generating computer program will start at a given genomic position and walk along the DNA in either direction, reading in a representation of the genome’s sequence (see next paragraph) and outputting the predicted Hi-C data value. I will use a computational model, such as a finite state machine, that can be iteratively learned from data using, for example, a Monte Carlo algorithm.

Ideally, this program would take the DNA sequence as input directly, but that is probably too hard. Instead, I propose to additionally learn a 1D reduced representation of the structure data de novo. This reduced representation assigns a discrete label $1 \ldots k$ to each genomic position that forms the input to the structure-generating program. It is possible to learn this representation from scratch because there are $n^2$ observed contact counts available for learning $n$ reduced representation labels. This process is conceptually similar to, for example, principle components analysis (PCA), in which a high-dimensional matrix is projected down to one (or several) most-informative vectors. In order to learn both the Hi-C generating program and the reduced representation simultaneously, the overall process will alternate the following steps: (1) optimize the program such that it best predicts Hi-C from a fixed reduced representation, and (2) optimize the reduced representation such that it forms the best input into this program.

This project has two outputs, the Hi-C-generating program and the 1D reduced representation, each of which is independently valuable. First, the structure-generating program exactly specifies the processes that produces the structure, so we can simply read the program to learn what biochemical processes are involved. I will validate the model by verifying that it captures known mechanisms that drive DNA structure, then look to see if it implies new mechanisms. Second, the 1D representation captures all of the relevant DNA elements that drive 3D structure, so this representation may uncover new types of elements or imply a currently unknown role in 3D structure for existing elements. I will aim to collaborate with experimental biologists to knock out or introduce these novel DNA elements in order to confirm their effect on the genome’s structure.

**Aim 3: Understanding the cell cycle through cyclical projection of single-cell data**

Understanding the regulatory state of individual cells is crucial to understanding regulation and development. In the past few years, single-cell genomics assays have been developed that measure several molecular properties of single cells: transcription (scRNA-seq), local accessibility (scATAC-seq), and 3D structure (scHi-C). These properties vary between cells in a population due to many effects, including development, the cell cycle and random noise.

Given single-cell measurements on a population of cells, I aim to jointly infer the two time coordinates of each cell: its position in the cyclical cell cycle and its position along a linear developmental time axis. Specifically, I aim to map each cell to a point $(x, \theta)$—where the Cartesian coordinate $x$ represents the cell’s position in development and the polar coordinate $\theta$ represents its position in the cell cycle—such that the pairwise distances cells in this cylindrical space best
match the similarities between the cells’ genomic measurements. (More generally, this can be extended to any number of Cartesian and polar coordinates in order to capture multiple orthogonal biological processes.) This method is preferable to existing Cartesian projection-based methods for analyzing single cell data because it can properly represent the cyclical nature of the cell cycle. Modeling the cell cycle is particularly important for structural measurements (ATAC-seq, Hi-C) because these features necessarily track with the cell cycle as the genome is replicated then packaged into centromeres in order to undergo cell division.

Performing this projection will identify the genomic measurements that track with both the cell cycle and development. This will grant insight into questions such as the following: What regulatory elements activate to signal the start of replication or cell division? How does the genome re-establish its 3D structure after cell division? Does development checkpoint with cell divisions, or are these uncoupled? Such a projection is also easy to interpret because it lends itself easily to visualization.

**Long-term goals**

I expect future developments in biological technologies to yield new types of data and pose new computational and statistical problems. In particular, upcoming technologies for genome editing, single-cell assays and super-resolution microscopy promise new opportunities for computational methods. As with existing types of genomics data sets, analyzing such data requires methods that are scalable and that accurately model the underlying phenomena while remaining interpretable enough to afford biological understanding. I will apply my expertise garnered from developing methods for existing genomics data to develop machine learning methods for solving these new problems as they arise in new data sets.

**References**


