PLOS COMPUTATIONAL BIOLOGY

REVIEW

Segmentation and genome annotation algorithms for identifying chromatin state and other genomic patterns

Maxwell W. Libbrecht[1☯]*, **Rachel C. W. Chan**[2,3☯], **Michael M. Hoffman**[2,3,4,5]*

1 School of Computing Science, Simon Fraser University, Burnaby, Canada, **2** Department of Computer Science, University of Toronto, Toronto, Canada, **3** Princess Margaret Cancer Centre, University Health Network, Toronto, Canada, **4** Department of Medical Biophysics, University of Toronto, Toronto, Canada, **5** Vector Institute for Artificial Intelligence, Toronto, Canada

☯ These authors contributed equally to this work.
* maxwell_libbrecht@sfu.ca (MWL); michael.hoffman@utoronto.ca (MMH)

Citation: Libbrecht MW, Chan RCW, Hoffman MM (2021) Segmentation and genome annotation algorithms for identifying chromatin state and other genomic patterns. PLoS Comput Biol 17(10): e1009423. https://doi.org/10.1371/journal.pcbi.1009423

Editor: Tamar Schlick, New York University, UNITED STATES

Published: October 14, 2021

Funding: This work was supported by the Natural Sciences and Engineering Research Council of Canada (RGPIN-2015-03948 to M.M.H.), https://www.nserc-crsng.gc.ca/. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Segmentation and genome annotation (SAGA) algorithms are widely used to understand genome activity and gene regulation. These algorithms take as input epigenomic datasets, such as chromatin immunoprecipitation-sequencing (ChIP-seq) measurements of histone modifications or transcription factor binding. They partition the genome and assign a label to each segment such that positions with the same label exhibit similar patterns of input data. SAGA algorithms discover categories of activity such as promoters, enhancers, or parts of genes without prior knowledge of known genomic elements. In this sense, they generally act in an unsupervised fashion like clustering algorithms, but with the additional simultaneous function of segmenting the genome. Here, we review the common methodological framework that underlies these methods, review variants of and improvements upon this basic framework, and discuss the outlook for future work. This review is intended for those interested in applying SAGA methods and for computational researchers interested in improving upon them.

Background and motivation

High-throughput sequencing technology has enabled numerous techniques for genome-scale measurement of chemical and physical properties of chromatin and associated molecules in individual cell types. Using sequencing assays, the Encyclopedia of DNA Elements (ENCODE) Project, the Roadmap Epigenomics Project, and myriad individual researchers have generated thousands of such datasets. These datasets quantify various facets of gene regulation such as genome-wide transcription factor binding, histone modifications, open chromatin, and RNA transcription. Each dataset measures a particular activity at billions of positions, and the collection of datasets does so in hundreds of samples across a variety of species and tissues. Transforming these quantifications of diverse properties into a holistic understanding of each part of the genome requires effective means for summarization. Segmentation and genome annotation (SAGA) algorithms (Box 1) have emerged as the predominant way to summarize activity

Box 1. Terminology

SAGA

We define a segmentation and genome annotation (SAGA) algorithm as a procedure that:

1. assigns to each position of a whole genome a label ("genome annotation"),
2. from a set of multiple ($\geq$3) classes,
3. by
   (a) integrating multiple independent observations at each position, and
   (b) modeling dependence between adjacent positions ("segmentation").

Previously, researchers have used several other terms to describe this task, including "segmentation" [1], "chromatin state annotation" [2], and "semi-automated genome annotation" [3]. We use "segmentation and genome annotation" instead of simply using "segmentation" because the latter only describes 1 of 2 important parts of the task. We use this term instead of "chromatin state annotation" because SAGA algorithms generalize to data types other than chromatin state, and indeed such uses predate the use for chromatin state alone [1,4–6].

Assay

An experiment that produces a measurement at each genomic position, such chromatin immunoprecipitation-sequencing (ChIP-seq) or assay for transposase-accessible chromatin-sequencing (ATAC-seq).

Label

One of a finite set of classes assigned to each genomic segment that shares similar activity. Other terms include "state" or "chromatin state."

Sample

A population of cells on which one can perform an assay, such as a primary tissue sample or a cell line. Other terms include "cell type," "epigenome," or "biosample."

at each position of the genome, distilling complex data into an interpretable précis of genomic activity.

SAGA algorithms take as input a collection of genomic datasets, such as ChIP-seq measurements of histone modifications or of transcription factor binding (Fig 1). The SAGA task is to use the input datasets to partition the genome into segments and assign a label to each segment. SAGA algorithms perform this task in a way that leads to positions with the same label having similar patterns in the input data.

Most existing SAGA algorithms employ a probabilistic model known as a hidden Markov model (HMM) or a related model such as a dynamic Bayesian network (DBN) (see "Hidden Markov model (HMM) formulation"). This model represents a scenario where each genomic position has an unknown label that corresponds to its activity of interest. In the model, some process generates observed data as a function of this label, and labels of neighboring positions

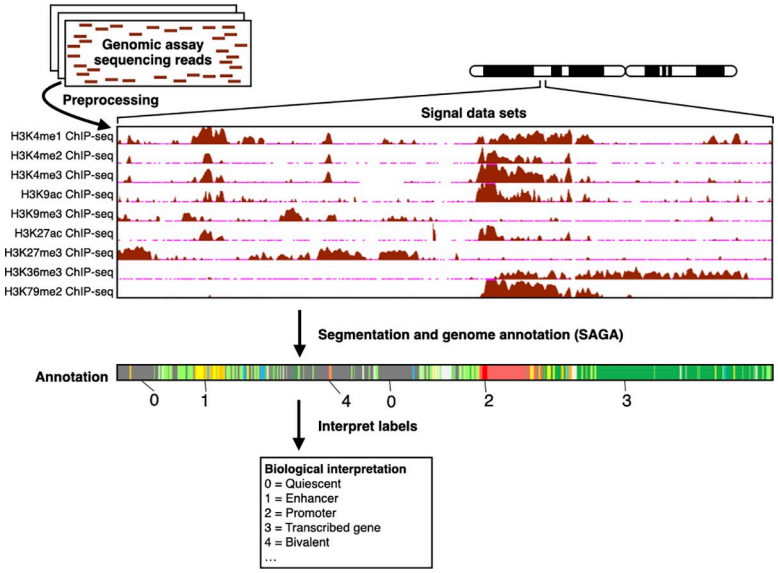

Fig 1. Overview of SAGA. First, preprocessing transforms genomic assay sequencing reads into signal datasets. Second, with signal datasets as input, a SAGA algorithm partitions the genome and assigns an integer label to each segment, yielding an annotation. Third, a researcher interprets the labels, assigning a biological interpretation to each. ChIP-seq, chromatin immunoprecipitation-followed by sequencing; SAGA, segmentation and genome annotation.

https://doi.org/10.1371/journal.pcbi.1009423.g001

influence each other. SAGA algorithms work by finding the model parameters and genome annotation that maximize the model likelihood.

The first SAGA methods were developed in the 2000s but have increased in usage recently, thanks to the wide availability of genomic datasets (Table 1). Large-scale genomic profiling

Table 1. Timeline of selected SAGA methods.

Year	Name or description	References
2007	HMMSeg	[1]
2010	Chromatin colors	[7]
2010	Chromatin states model	[8]
2012	ChromHMM	[2,9–11]
2012	Segway	[9,12–14]
2013	TreeHMM	[15]
2015	Spectacle	[16]
2015	hiHMM	[17]
2015	Ensembl Regulatory Build (with Segway, ChromHMM)	[18]
2015	EpiCSeg	[19]
2015	Segway+GBR	[3,20]
2016	IDEAS	[21–24]
2017	GenoSTAN	[25]
2017	diHMM	[26]
2018	iSeg	[27]
2018	StatePaintR	[28]
2019	RT States	[5]
2019	ConsHMM	[4]
2020	modHMM	[29]
2020	SPIN	[30]
2020	SegRNA	[6]

https://doi.org/10.1371/journal.pcbi.1009423.t001

projects such as ENCODE [31] and Roadmap Epigenomics [10] produced SAGA annotations as a primary output. Researchers have developed a large variety of SAGA strategies with the goal of improving upon the basic SAGA framework.

In this review, we summarize the main strategies used by most SAGA methods. Then, we discuss differences between methods, the challenges they face, and the outlook for future work.

This review is intended for 2 audiences. First, analysts interested in applying SAGA methods or using the resulting annotations will find this review useful for understanding how the methods work and the steps and choices involved in applying existing methods. Second, methods researchers interested in improving and extending these methods will find this review useful for understanding the diversity of existing methods and where they have room for improvement.

The structure of this review follows the steps by which a researcher proceeds from raw data to scientific insight. For each step, we review the variations in each step found in the literature and discuss considerations one must make in choosing between these variants. We devote a section to each of the following steps:

1. Selection and processing of input data
2. Formulation and optimization of probabilistic model
3. Selection of resolution parameters
4. Selection of parameters for number of labels
5. Interpreting unsupervised labels
6. Extending to multiple cell types
7. Evaluating annotations
8. Visualizing annotations

A major caveat limits our discussion: As of this writing, researchers have not performed comprehensive benchmarking of SAGA methods. This caveat likely results from challenges in evaluating these methods that we discuss. Therefore, the optimal choice for most modeling choices remains an open scientific question.

Input data

Experimental assays used for input data

SAGA methods typically use as input a number of different experimental datasets, each describing some local property of the genome [32]. Such properties might include chromatin accessibility or presence of some DNA-binding protein. Although input data initially came from microarray methods such as tiling arrays [33], they now usually come from sequence census assays [34].

A common collection of input datasets might measure histone modifications or DNA-binding proteins (using assays like ChIP-seq [35] or cleavage under targets and release using nuclease (CUT&RUN) [36]) and chromatin accessibility (using assays like deoxyribonuclease-sequencing (DNase-seq) [37,38] or ATAC-seq [39]). Supplying a SAGA algorithm with datasets that measure chromatin activity yields an output annotation that captures the regulatory state of chromatin. Creating these chromatin activity annotations has served as the predominant use of SAGA methods thus far.

Less frequently, researchers have gone beyond measurements of chromatin and DNA-binding proteins and have used SAGA methods for other kinds of data. The output annotation summarizes the input datasets, so the choice of input greatly influences the annotation's content and its subsequent interpretation. SAGA methods can work for any sort of dense linear signal along the genome. Individual studies have applied it DNA replication timing data [3,5,30], interspecies comparative genomics data [4], and RNA-seq data [6]. Other studies have even found ways to incorporate nonlinear chromatin 3D genome organization data into the SAGA framework [3,30].

The choice of input datasets is critically important. Unsupervised SAGA methods identify the patterns most prominent in their input data. Therefore, providing more input datasets does not always improve results and may hide patterns prominent only in a subset of the datasets. To simplify understanding of the resulting annotations, researchers commonly use input datasets from just a single type of biological process, such as chromatin or transcription.

Signal representation of genomic assays

Most genomic assay data so far has come from bulk samples of cells. These data depict a noisy mixture of sampling an assayed property from the many cells within the population. These cells may represent subpopulations of slightly different types or within different cell cycle stages. Thus, each subpopulation might have different characteristics in the assayed properties. In the mixture of cell subpopulations, only frequently sampled properties will rise above background noise. By comparison, less frequently sampled properties seen in a minority of cells may remain indistinguishable from background noise.

Often, the property examined by an epigenomic assay is exhibited or not exhibited by some position of a single chromosome in a single cell, with no gradations between the extremes. For example, at some nucleotide of 1 chromosome in a single cell, an interrogated histone modification is either present or it is not. A single diploid cell has 2 copies of the chromosome. Thus, at that position, each eudiploid cell can have only 0, 1, or 2 instances of the histone modification.

Summing or averaging discrete counts over a population of cells leads to a representation of the assay data called "signal." Signal appears as a continuous-scale measurement. Signal arises, however, only from the aggregation of position-specific properties, which, in each cell, may have only a small number of potential ordinal values at the moment of observation.

Unlike epigenomic assays, transcriptomic assays can measure any number of transcript copies of 1 position per cell. Despite similar data representations, one must avoid the temptation to interpret epigenomic signal intensity as one might interpret transcriptomic signal intensity. For a transcriptomic assay, greater signal intensity might reflect a greater "level" of some transcriptional property within each cell. For an epigenomic assay, greater signal intensity indicates primarily that a higher number of cells within a sample have the property of interest.

In both the epigenomic and transcriptomic cases, it remains difficult or impossible to untangle the contribution to higher signal intensity that arises from frequency of molecular activity within each cell of a subpopulation from that from the composition of subpopulations within a whole bulk population. Improvements in single-cell assays, however, may enable SAGA algorithms on data from single cells in the near future (see "Outlook for future work").

Preprocessing of input data

SAGA methods generally use a signal representation of the input data. This signal representation originates from raw experimental data, such as sequencing reads, by way of a

preprocessing procedure. For simplicity, we describe the steps of preprocessing as if a human analyst conducted them all individually, although some SAGA software packages might perform some steps without manual intervention:

Required preprocessing for all SAGA methods:

1. The analyst transforms the experimental data into raw numeric signal data.
   - For sequencing data, the analyst:
     1. aligns each sequencing read to the reference genome (producing a sequence alignment map (SAM) or binary alignment map (BAM) [40] file),
     2. may choose to extend each read to an estimated length of the DNA fragment it begins, and
     3. computes the number of reads per base or extended reads per base for each genomic position (producing a Wiggle [41], bigWig [42], or bedGraph [41] file) [12,13].
   - For microarray data, the analyst:
     1. acquires microarray signal intensity for the experimental sample and for a control sample, and
     2. computes the ratio of experimental intensity to control intensity.
2. The analyst chooses units to represent the strength of activity at each position and may perform further transformation of the raw numeric signal data into these units.
   - For sequencing data, the analyst commonly uses one of:
     - read count (no transformation),
     - fold enrichment of observed data relative to a control [9], or
     - $-\log_{10}$ Poisson p-values indicating the likelihood of statistically significant peaks relative to control [22]. The latter 2 units can mitigate experimental artifacts because they compare to a control experiment such as a ChIP input control.
   - For microarray data, the analyst commonly performs $\log_2$ transformation of the intensity ratios [7,43,44].

Optional preprocessing or preprocessing required only for specific SAGA methods:

3. The analyst may normalize data to harmonize signal across cell types [45]. Normalization proves especially important when annotating multiple cell types (see "Annotating multiple cell types").
4. To prevent large outlier signal values from dominating the results, the analyst may transform signals using 1 of 3 variance-stabilizing transformations of each signal value x:
   - asinh x [12],
   - $\log_2(x + \text{pseudocount})$ [22], or
   - an empirical variance-stabilizing transformation [46].
5. The analyst may downsample 1-bp resolution signal into bins (see "Spatial resolution"). This involves computing one of:
   - average read count,

- reads per million mapped reads fold enrichment [47],
- total count of reads [19,48,49], or
- maximum count of reads of each bin [9,21].

Binning greatly decreases the computational cost of the SAGA algorithm and can improve the data's statistical properties.

6. The analyst may binarize numeric signal data into presence/absence values (potentially producing a browser extensible data (BED) [41] file) [2,15,26,50,51]. Binarizing signal simplifies analysis by avoiding issues related to the choice of units but eliminates all but one bit of information about signal intensity per bin.

Missing data

Genomic assays usually cannot produce signal for every region of the whole genome. Regions where an assay cannot provide reliable information about the interrogated property constitute "missing data" for that assay. Missing data in sequencing assays may arise due to unmappable sequences, which occur when repetitive reads do not uniquely map to a region [52,53]. Missing data in microarray assays come from regions covered by no microarray probes. There are 3 main ways to treat regions of missing data: (1) by treating missing data as 0-valued data; (2) by decreasing the model resolution, averaging over available data so that the missing data has limited impact; or (3) statistical marginalization over the missing data [12,15,54].

When analyzing coordinated assays across multiple cell types, researchers may have to contend with having no data on some properties within a subset of cell types. This represents another kind of missing data: one with an entire dataset missing rather than only data at specific positions. Researchers can impute [26] entire missing datasets through tools such as ChromImpute [55], PREDICTD [56], or Avocado [57]. Alternatively, IDEAS [23] uses an expectation–maximization (EM) approach to perform imputation and annotation simultaneously.

Hidden Markov model (HMM) formulation

Many SAGA methods rely on an HMM, a probabilistic model of the relationships between sequences of observed events and the unobservable hidden states, which generate the observed events. The structure of HMMs, and similar models such as DBNs [58], naturally reflect the SAGA task of clustering observed data generated by processes that act on sequences of genomic positions.

Simple HMM example

As an illustration of a simple HMM, consider a dog, Rover, and his owner, Thomas. Thomas is 5 years old and too short to see out of the windows in his home. Rover can leave the house through his dog door and loves taking walks, playing indoors, and napping. Every morning, he will either wait by the door for Thomas, play with his squeaky toys, or sleep in. Whichever action he takes depends on the weather he sees outdoors. For example, on rainy days Rover will more likely nap or play with his toys indoors.

Thomas must infer the state of the weather outside, hidden to him, based on the behavior he observes from Rover. Thomas knows the weather patterns near his home. In particular, Thomas knows that rainy weather likely continues across multiple days, so his inference must take into account the whole sequence of Rover's behavior.

This scenario fits well into the HMM framework. It has a sequence of observations (Rover's behavior) generated by hidden, nonindependent unobservables (the weather outside). One would like to infer the sequence of hidden unobservables based on the sequence of observations.

Mathematical formulation

Formally, we can define an HMM over time $t \in \{1,\ldots,T\}$ as follows [59,60]. Let the sequence of observed events $\boldsymbol{X} = \{X_t\}_{t=1}^{T}$ consist of each observed event X_t at every time t. Let the sequence of hidden states $\boldsymbol{Q} = \{Q_t\}_{t=1}^{T}$ consist of each hidden state Q_t at every time t. Each Q_t takes on a value q_t from a set of m possible hidden state values (Fig 2A).

Under the Markov assumption, the probability of realizing state value q_{t+1} at the next time step $t+1$ depends only on the current state value q_t:

$$P(Q_{t+1} = q_{t+1} | Q_t = q_t, Q_{t-1} = q_{t-1}, \ldots, Q_1 = q_1) = P(Q_{t+1} = q_{t+1} | Q_t = q_t).$$

We define the transition probability $A(q_{t+1}|q_t) = P(Q_{t+1} = q_{t+1}|Q_t = q_t)$, which reflects the frequency of moving from state q_t to state q_{t+1}.

We define the emission probability $B(x_t|q_t) = P(X_t = x_t|Q_t = q_t)$ as the probability that the observable X_t is x_t if the present hidden state $Q_t = q_t$. Specifically, we assume that $B(x_t|q_t)$ depends only on $Q_t = q_t$, such that

$$P(X_t = x_t | Q_t = q_t, Q_{t-1} = q_{t-1}, \ldots, Q_1 = q_1) = P(X_t = x_t | Q_t = q_t).$$

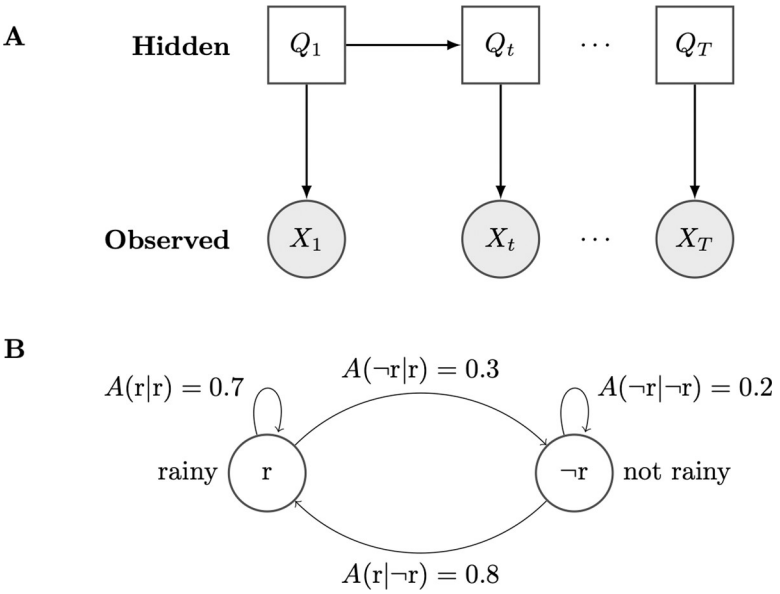

Fig 2. Two representations of an HMM. (A) Conditional dependence diagram representation of an unrolled HMM with sequence of hidden states $\{\boldsymbol{Q_t}\}_{t=1}^{T}$ and sequence of observations $\{\boldsymbol{X_t}\}_{t=1}^{T}$. In this representation, each node represents a hidden discrete (white rectangle) or observed continuous (gray circle) random variable. For every index t, each hidden random variable $\boldsymbol{Q_t}$ takes on some value $\boldsymbol{q_t}$; similarly, each observed variable $\boldsymbol{X_t}$ takes on some value $\boldsymbol{x_t}$. $\boldsymbol{X_t}$ may represent either scalar or vector observations. Solid arcs represent conditional dependence relationships between random variables. (B) State transition diagram representation of Rover and Thomas's weather example. In this representation, each node represents a potential value of the hidden variable $\boldsymbol{Q_t}$. The hidden variable takes on values **r** (rainy) or **¬r** (not rainy) on any given day t. Solid arcs represent transitions between hidden states, which have transition probabilities A. HMM, hidden Markov model.

https://doi.org/10.1371/journal.pcbi.1009423.g002

Finally, we define the hidden state probability at the first time step as $\pi_0(q_0) = P(Q_0 = q_0)$. We can fully define an HMM $M = (A, B, \pi_0)$ by specifying all of A, B and, π_0.

In the case of Rover and Thomas, we have $m = 2$ possible hidden states (rainy, not-rainy) and 3 possible observations (Rover is napping, playing indoors, or waiting by the door). To Thomas, the hidden variable Q_t captures the weather outside, while the observed variable X captures Rover's behavior. The probability of the state of the weather outside changing on a day-to-day basis is defined by the transition probabilities A (Fig 2B). The probability of Rover's behavior, given the weather, is defined by the emission probabilities B.

Algorithms for inference, decoding, and training

Inference. The main task one uses HMMs for is to quantify how well some predicted sequence of hidden states fits the observed data. Other common tasks like decoding or training serve as variations of, or build on, this inference task.

In HMMs inference, we can compute the likelihood of any sequence of hidden states $\boldsymbol{Q}$. We use the sequence of observed events $\boldsymbol{X}$ and the model probabilities M to compute the likelihood function $P(\boldsymbol{X}|\boldsymbol{Q}, M)$. The likelihood function is the probability that our predicted sequence of hidden states produced our observed sequence of observed states. We often compute the likelihood function using the forward–backward algorithm [61,62].

Viterbi decoding. Given a sequence of observed events $\boldsymbol{X}$, we often wish to know the maximum likelihood sequence of corresponding hidden states $\boldsymbol{Q}$. For example, if Thomas observes that in the past 3 mornings, Rover slept, played, and then slept again, what weather sequence outside is most likely?

To answer this question, we decode the optimal sequence of hidden states $\boldsymbol{q}^*$ such that $\boldsymbol{q}^* = \arg\max_Q P(\boldsymbol{Q}|\boldsymbol{X}, M)$. The Viterbi algorithm [63] provides an efficient solution for this problem, making it unnecessary to compare the likelihood for every possible sequence of hidden states.

Training. Usually, we do not know the model parameters (A, B, π_0) and must learn them from data. We define training as the process of learning these parameters, and training data as the sequence of observations upon which we learn. An efficient algorithm that finds the global optimum parameter values for some training data does not exist. Instead, researchers commonly train HMMs using EM [64] algorithms such as the Baum–Welch algorithm [65], which find a local optimum. Other reviews [59] describe inference and training methods in more detail.

HMMs for SAGA

We can readily apply the HMM formalization to genomic data for use in SAGA methods. Instead of time, we define the dynamic axis t in terms of physical position along a chromosome. Each position t refers to a single base pair or, in the case of lower-resolution models, a fixed-size region (see "Spatial resolution"). The observation at each genomic position usually represents genomic signal (see "Input data"). Each position's hidden state represents its label (see "Understanding labels"). As a result, decoding the most probable sequence of hidden states reveals the most probable sequence of labels across the genome. We call this resulting sequence of labels an annotation.

Many SAGA methods use an HMM structure [2,5,12,15,19,26,44,47], or generalizations thereof. For example, DBNs are generalizations of HMMs that can model connections between variables over adjacent time steps. Methods such as Segway [12] use a DBN model in their approach to the SAGA problem. This can make it easier to extend the model to tasks such as semi-supervised, instead of unsupervised, annotation [66].

Spatial resolution

Baroque music often employs a musical architecture known as "ternary form." Specifically, pieces of this structure follow a general "ABA" pattern, whereupon the second "A" section recapitulates the first with some variation. Each section contains multiple musical "sentences," which may repeat or vary. Just like linguistic sentences, each musical sentence contains clusters of notes, or motifs, between "breaths" in the musical articulation. Finer examination of the motifs shows that they contain a few notes and chords each. Finer examination of the notes themselves shows that they behave just like isolated phonemes in speech, with little meaning on their own.

The genome resembles a musical composition in that one observes different behaviors at different scales. The scale of genomic behavior one wishes to observe influences the choice of SAGA method and parameters chosen for the method. To observe nucleosome-scale behavior such as genes, promoters, and enhancers, one desires about 10^3 bp segments. To describe behavior on the scale of topological domains [67], one desires segments of length approximately 10^5 to 10^6 bp [1,3,20].

The most important parameter influencing segment length is the underlying resolution of the SAGA method. As noted above (see "Input data"), most SAGA methods downsample data into bins. To observe nucleosome-scale segment lengths (about 10^3 bp), one should use 100 bp to 200 bp resolution [2,12,21]. To observe domain-scale segment lengths (about 10^5 bp), one should use approximately 10^4 bp resolution [3,7,30]. Segway [12] and RoboCOP [68] provide some of few SAGA methods optimized for single-base resolution inference and can identify behavior on a 1-bp scale. While most existing SAGA methods handle data at just one genomic scale, there exist methods capable of learning from data at multiple genomic scales [26].

Limitations of the experimental data itself influence the choice of SAGA model resolution. Spatial imprecision in ChIP-seq data gives it an inherent resolution of about 10 bp. More precise assays such as ChIP-exo [69] and ChIP-nexus [70] can approach 1 bp resolution. Conversely, assays like DNA adenine methyltransferase identification (DamID) and Repli-seq have a coarser resolution of $\geq$100 bp.

The desired scale may also influence the choice of input data. When aiming to annotate at the domain scale, one should include data with activity at this scale, such as replication time data and Hi-C data [3,5,7,30]. The inclusion of long-range contact information from Hi-C data poses a challenge because standard algorithms for HMMs cannot be used for a probabilistic model that includes long-range dependencies. Therefore, one must instead use alternative approaches such as graph-based regularization [3] or approximate inference [30].

SAGA methods model segment length through their transition parameters. HMM models assume a geometric distribution in determination of a segment's length [71]. Related DBN methods can include constraints to tune segment length further. Constraints include the enforcement of a minimum or maximum segment length [12]. Enforcement of a minimum segment length ensures that one does not obtain segments shorter than the effective resolution of the underlying data or biological phenomena. Probabilistic models often additionally use a prior distribution on the transition parameters during training to encourage them to produce shorter or longer segment lengths.

Choosing the number of labels

Most SAGA methods require the user to define the number of labels. Using more labels increases the granularity of the resulting annotation at the cost of added complexity. Typically, the number of labels ranges from 5 to 20, with more recent work favoring 10 to 15 labels.

One might think to make the choice of number of labels automatically with a statistical approach. The Akaike information criterion (AIC), Bayes information criterion (BIC), and factorized information criterion (FIC) [72] measure the statistical support a particular number of labels has. Instead of a fixed number of labels, one may give the model flexibility to choose the number of labels during training and include a hyperparameter that encourages it to choose a higher or lower number [17]. Or one might define labels according to local minima in an optimization based on a network model of assays [51]. One could even exhaustively assign a separate label to every observed presence/absence pattern in binary data [48].

In practice, however, researchers rarely use these statistical approaches for determining the number of labels. Optimizing an information criterion does not necessarily yield the most interpretable annotation. Interpretability reigns supreme in most SAGA applications. End users find annotations most useful when they have about 5 to 20 labels for 2 reasons. First, most can only articulate that many known distinctions between classes of genomic elements. Second, even if one could find meaningful distinctions between a large number of labels, few using the resulting annotations could keep fine distinctions between such a large number of patterns in their working memory [73]. Even if a statistical approach supported the use of 50 labels, the complexity of such an annotation would make it impractical for most users.

Understanding labels

SAGA methods are unsupervised. The labels they produce usually begin with integer designations without any essential meaning. Ideally, each label corresponds to a particular category of genomic element. To make this correspondence explicit, we must assign a biological interpretation, such as "Enhancer" or "Transcribed gene," to each label.

Usually, one makes assignments of labels to biological interpretations in a postprocessing step. In postprocessing, a researcher compares each label to known biological phenomena and assigns an interpretation that matches the researcher's understanding of molecular biology. For example, a label characterized by the histone modification H3K36me3 (associated with transcription) and enriched in annotated gene bodies might have the interpretation "Transcribed." A label characterized by H3K27ac and H3K4me1, both histone modifications canonically associated with active enhancers, might have the interpretation "Enhancer" [31].

The interpretation process provides an opportunity to discover new categories of genomic elements. For example, one SAGA study found that their model consistently produces a label corresponding to transcription termination sites. Previously, none had described a distinctive epigenetic signature for transcription termination [9].

Manual interpretation proves time-consuming for human analysis. Applying SAGA to multiple cell types independently exacerbates this problem (see "Annotating multiple cell types").

Two existing methods automate the label interpretation process: expert rules and machine learning. In both cases, an interpretation program considers the information that a researcher would use for interpretation. This includes examining the relationship between labels and individual input data properties. It also includes reviewing colocalization of labels with features in previously created annotations. These annotations may have come from SAGA approaches or other manual or automated methods.

In the expert rule approach, an analyst designs rules about what properties a given label must have to receive a particular interpretation. The analyst then applies these rules to assign interpretations to labels from all models [18].

In the machine learning approach, one trains a classifier on previous manual annotations. The classifier then learns a model that assigns interpretations to labels given their properties

[14]. One analysis [14] found that automatic interpretation agreed with manual for 77% of labels, compared to 19% expected by chance.

Annotating multiple cell types

There now exist epigenomics datasets describing hundreds of biological samples (Fig 3A). Researchers have correspondingly adapted SAGA methods to work with many samples simultaneously.

We use the term "sample" to refer to some population of cells on which one can perform an epigenomic assay. A sample could correspond to a primary tissue sample, a cell line, cells affected by some perturbation such as drug or disease, or even cells from different species.

The simplest approach for annotating multiple samples involves simply training a separate model on each sample [14] (Fig 3B).

The large number of models produced by this approach necessitates using an automated label interpretation process (see "Label interpretation").

Two categories of approaches aim to share information across samples. The first, "horizontal sharing" or "concatenated" approaches, share information between samples to inform the label-training process. The second, "vertical sharing" or "stacked" approaches, share position-specific information to inform the label assignment of each position.

Horizontal sharing: Emphasizing similarities across samples for learning labels

The simplest way to remove the need for interpreting multiple models is to apply a single model across many samples. To do this, one can treat each sample as referring to separate

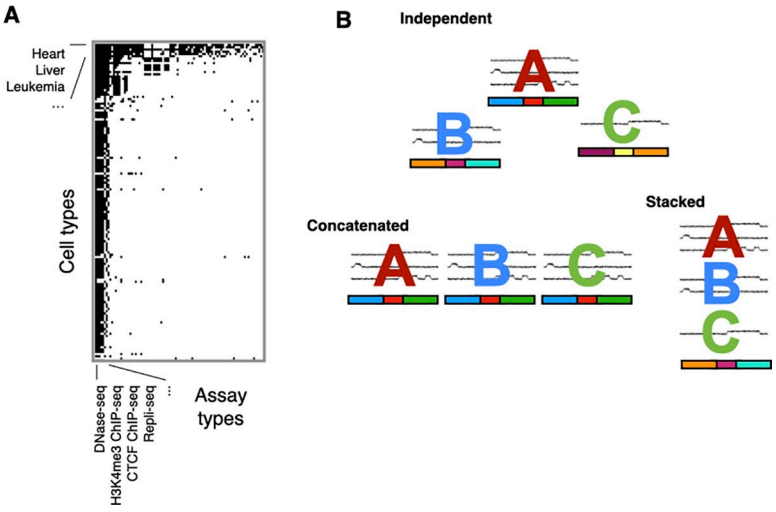

Fig 3. Annotating multiple cell types. (A) Datasets generated by the ENCODE and Roadmap Epigenomicsconsortia as of 2019. The black cells represent the datasets actually generated out of a larger number of potential combinations of cell type and assay type. (B) Annotating 6 datasets from 3 different samples: 3 from cell type A, 2 from cell type B, and 1 from cell type C. Colored letters over signal data indicate data associated with those samples. One can use 3 different SAGA strategies with this collection of datasets: Independent: performing training and inference completely independently on each sample. This yields a different annotation for each sample. Concatenated (horizontal sharing): training a single model across all cell types. This yields 1 annotation per sample with a shared label set. Each sample must have the same datasets, necessitating imputation of any missing datasets. Stacked (vertical sharing): performing training and inference on datasets from all samples. This yields a single pan–cell-type annotation. ChIP-seq, chromatin immunoprecipitation-followed by sequencing; DNase-seq, sequencing DNase I hypersensitive sites sequencing; ENCODE, Encyclopedia of DNA Elements; SAGA, segmentation and genome annotation.

https://doi.org/10.1371/journal.pcbi.1009423.g003

copies of a longer genome added horizontally after the first one in a "concatenated" approach (Fig 3B). One performs concatenated training and inference little differently than if the data from different samples pertained to different chromosomes in the same genome. Because all samples share a single concatenated model, researchers need only perform postprocessing interpretation once.

The concatenated approach has wide usage [9,10,74] but has 2 downsides. First, concatenated SAGA requires that every sample has data from the same assays. In practice, this criterion often does not hold true. This means that—unless these assays are imputed or treated as missing (see "Vertical sharing: emphasizing similarities across samples in positional information")—one must exclude data for an assay conducted in even all but one samples. In a simple concatenated approach, one cannot annotate a sample that lacks even 1 dataset present in the others.

Second, data from different samples can have artifactual differences or batch effects. Applying the same model across multiple cell types assumes that all datasets from the same assay type have similar statistical properties. This can result in label distributions that vary wildly across samples and biologically implausible sample-specific labels. Data normalization can help abate the problem of different statistical properties between samples but usually does so incompletely. This problem is particularly significant when using continuous signal. In contrast, binarizing the data (see "Input data") can cover up some experimental biases.

One might expect that concatenated annotation would benefit training by increasing the amount of training data. As it turns out, multiplying the amount of training data does not significantly aid the training process, as the types of labels vary little across samples. Most complex eukaryotic organisms studied with SAGA have very large genomes, and just 1 sample provides plenty of training data. In fact, for computational efficiency, researchers often train on just a fraction of the available samples [10], a fraction of the genome from a given sample [12] or both.

Vertical sharing: Emphasizing similarities across samples in positional information

Another class of multisample SAGA methods shares position-specific information across samples as part of the annotation process. These methods take advantage of the nonindependence of biological activity across samples at a genomic position. For example, if a given position has an active enhancer label in many samples, it is more likely to act as an active enhancer in a new sample.

The simplest type of vertical sharing approach learns a model on data from all samples jointly (Fig 3B). One can implement this "stacked" approach by adding datasets from all samples vertically into a single combined model. A stacked model captures patterns of activity across multiple cell types. For example, a stacked model, unlike an independent model, can find a label for an enhancer active in cell type A and cell type B but inactive in cell type C.

Although conceptually simple, the stacked approach tends not to work well for more than several cell types. Stacking fails with larger number of cell types because each pattern of activity requires its own label. Therefore, the number of labels must grow exponentially in the number of samples. A simple stacked model that treats all assays as independent also ignores the relationship between assays on the same cell type or the same assay type on different cell types.

A second approach uses a concatenated model that additionally learns a position-specific preference over the labels for each position. Through this preference, data from 1 sample can influence inference on another. Two implementations have applied variants of this hybrid horizontal–vertical sharing approach. First, TreeHMM [15] uses a cellular lineage tree as part of

its input. For each genomic position, TreeHMM models statistical dependency between the label of a parent cell type and that of a child cell type. Second, IDEAS [21] uses a similar approach to TreeHMM but dynamically identifies groups of related samples rather than using a fixed developmental structure. The IDEAS model allows these sample groups to vary across positions, which allows for different relationships between samples in different genomic regions.

A third approach for vertical sharing uses a pairwise prior to transfer position-specific information between cell types [3,20]. In other words, if position *i* and position *j* received the same label in many other samples, then they should be more likely to receive the same label in an additional sample. In contrast to the other methods in this section, the pairwise prior approach does not require the use of concatenated annotation. Therefore, the pairwise approach has the advantage of not requiring the same available data in all cell types.

A fourth approach imputes missing datasets in the target cell type, then applies any of the above annotation methods to the imputed data [55]. Imputation provides 3 advantages. First, it ensures that all target cell types have the same set of datasets. Second, one can conduct imputation entirely as a preprocessing step, allowing the use of any SAGA method afterward. Third, the imputation process can normalize some artifactual differences between datasets, making concatenated annotation more appropriate.

Vertical sharing approaches have the downside that one cannot understand the annotation of each sample in isolation. This arises from the influence on label assignments in 1 sample by data from other samples. In particular, vertical sharing tends to mask differences between samples. For example, if some position has an enhancer label in many samples, vertical sharing approaches will annotate that position as an enhancer in a target cell type, too, even with no enhancer-related data in the target cell type.

Evaluating SAGA annotations

Researchers use 2 categories of approaches to evaluate SAGA annotations. The first comprises qualitative approaches, in which a researcher assesses how well various statistics of an annotation match their expectations. These statistics might include the genomic coverage of each label, the distribution of segment lengths, the emission and transition parameters of the underlying probabilistic model, and the enrichment of each label for previously annotated genomic elements. Such analysis can show whether an annotation captures the expected parts of genome biology. Unfortunately, there currently are no generally agreed upon statistics that must hold for a high-quality annotation.

The second category of evaluation approaches comprises quantitative metrics. These metrics usually take the form of a prediction problem. For example, how accurately can one predict the RNA-seq expression of a gene given just the annotation label at the gene's promoter? One might intuit that a high-quality annotation would separate high-expression and low-expression genes better than a poor annotation. Researchers define similar evaluation metrics based on enhancer RNA expression or identifying previously annotated elements [14,22]. Prediction performance is usually poor in absolute terms because annotation labels are discrete. Such prediction tasks are useful for the purpose of comparing different annotations but do not serve as a realistic application as the annotations.

Several challenges complicate evaluation of SAGA methods. The unsupervised nature of these methods makes defining a single standard for quality impossible. Moreover, an annotation with more labels and shorter segments than another will have better performance according to most quantitative prediction metrics, but the former annotation is more complex and therefore less understandable. Therefore, there exists a trade-off between some quality metrics

and interpretability, and better quantitative metrics might mean a less useful annotation. In part for these reasons, no one has published a comprehensive benchmarking of the relative performance of different SAGA methods and the effect of the modeling choices described in this review.

Using and visualizing SAGA annotations

A number of resources can aid in the application of SAGA algorithms and annotations. Reference annotations exist for many cell types. These obviate the need for a user of the annotation to actually run a SAGA method. Alternatively, if the user must run a SAGA algorithm on their own data, standardized protocols describe how to perform this process [11,75].

Most users of SAGA annotations view them through 1 of 3 visualization strategies. The first, and most common, strategy displays individual annotations as individual rows or "tracks" on a genome browser (Fig 4A). In each row, the browser displays the segments of that annotation for a region of 1 chromosome, usually indicating the label by color. Popular genome browsers for displaying segmentations include the University of California, Santa Cruz (UCSC) Genome Browser [41], the Washington University in St. Louis (WashU) Epigenome Browser [78], and Ensembl [79].

A second visualization strategy integrates annotations of all samples (Fig 4B). This visualization stacks all labels for a given position on top of one another and scales the vertical axis by an estimate of functional importance of that position. This importance can be estimated using the CAAS, which measures activity that is correlated with evolutionary conservation [14]. Calculating CAAS comprises 2 steps. First, for each annotation, one calculates a horizontal label-wise CAAS, the label's genome-wide correlation with evolutionary conservation. Second, for

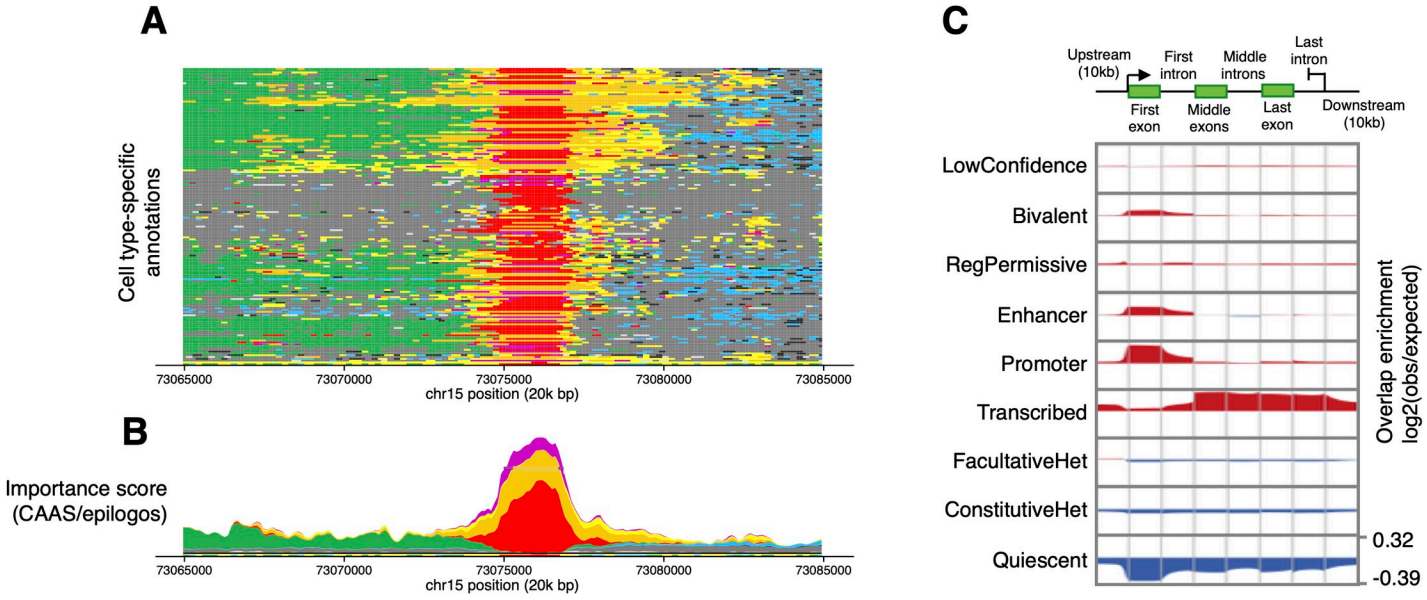

Fig 4. Visualizations of SAGA annotations. (A) Genome browser display showing 164 cell type annotations for a 20-kbp region on human chromosome 15 (GRCh37/hg19) [76]. Each annotation has 8 labels: Promoter (red), Enhancer (orange), Transcribed (green), Permissive regulatory (yellow), Bivalent (purple), Facultative heterochromatin (light blue), Constitutive heterochromatin (black), Quiescent (gray), and Low Confidence (light gray). (B) Importance score (CAAS) for the same region. Total height at each position indicates the position's estimated importance. Height of a given color band denotes the contribution toward importance of the associated label. (C) Genome-wide visualization of the SAGA annotation for 164 samples aggregated over GENCODE [77] protein-coding gene components. Rows: the 9 labels of the annotation. Columns: gene components, including 10 kbp flanking regions upstream and downstream. Each cell shows the enrichment of the row's label with a position along the column's component. Figures derived from [14]. CAAS, conservation-associated activity score; SAGA, segmentation and genome annotation.

https://doi.org/10.1371/journal.pcbi.1009423.g004

each position, one calculates a vertical position-specific CAAS, the average label-wise CAAS across the label at that position for all annotations.

A third visualization strategy aggregates information about each label across the entire genome. This shows the enrichment of each label at positions of known significance, such as gene components (Fig 4C) or curated enhancers. Tools such as Segtools [80] and deepTools [81] can create these visualizations.

SAGA annotations can provide valuable reference datasets to other analyses and tools. The assignment of one and only one label from a small set to every mappable part of the genome greatly eases downstream analyses. SAGA annotations summarize genomic activity in a much simpler way than the individual input datasets, and even than processed versions of the input datasets such as peak calls.

Most SAGA annotations are in the tab-delimited BED format, using the "name" column for the annotation label (https://genome.ucsc.edu/FAQ/FAQformat.html#format1). This makes it easy to remix SAGA annotations with other datasets using powerful software such as BEDTools [82]. SAGA annotations form building blocks for methods for integrative analysis of genomic data such as CADD [83].

Conclusions and outlook for future work

SAGA methods provide a powerful and flexible tool for analyzing genomic datasets. These methods promise to continue to play an important role as researchers generate more datasets. Despite the large existing literature, future work could still address many challenges.

Alternate scales and data types

Nucleosome-scale annotations (100 bp to1,000 bp segments) of histone modifications have wide usage. While annotations of different data types or at different length scales exist, they are less widely used. Currently, there exist reference domain annotations with segments of length 10^5 bp to 10^6 bp for only a small number of samples [3,7,47,84] and few or no annotations at other scales.

Data preprocessing

Genome annotations would improve with better processing and normalization of input datasets. Representations such as fold enrichment used by existing methods seem primitive compared to more rigorous quantification schemes used in RNA-seq analysis such as transcripts per million (TPM). One could also improve SAGA preprocessing by more frequently incorporating information from multimapping reads [85].

Confidence estimates

Most methods do not report any measure of confidence in their predictions. Two types of confidence would prove useful. First, one would often like to know the level of confidence that a position in some sample has label X and not label Y. Second, in many cases, one would like to have confidence in a differential labeling between 2 samples—that cell type A and cell type B have different labels. Two methods work toward a solution for the second problem [86,87], but there remains much room for further work.

Determining the number of labels and discovering new element types

As we discuss, researchers do not agree on a consensus number of labels. While data-driven methods for making this choice exist, they are not widely used. These methods are seldom

used in part because they often suggest larger numbers of labels than a human might easily interpret.

Novel categories of genomic element might be hiding in poorly characterized labels only visible when using a large number of labels. Investigation of such labels may be a fruitful line of research. If data-driven methods consistently suggest the same number of labels, this may provide insight into a true number of biologically distinct recurring epigenetic states.

Continuous representations

Existing SAGA methods output a discrete annotation, assigning a single label to each position. In this discrete approach, annotations cannot easily represent varying strength in activity of genomic elements or elements that simultaneously exhibit multiple types of activity. A continuous annotation approach analogous to the topic models used for text document classification might address this limitation [88].

Single-cell data

Existing SAGA methods use data from bulk samples of cells. Increasing availability of data from single-cell assays necessitates the development of methods that can leverage this additional information.

Pan–cell-type annotation

The semantics of genome annotations correspond poorly to the way most molecular biologists conceptualize genomic elements. Most existing annotations are cell-type-specific—the annotation states that a given locus acts as an active enhancer in cell type A. In contrast, researchers often state that a given locus "is an enhancer."

In contrast, other annotations—such of those of protein-coding genes—serve as a pan–cell-type characterization. Each gene has a fixed location, and only its expression varies across samples.

There exists a need for pan–cell-type epigenome annotations. Such an annotation would define fixed intervals for regulatory elements such as promoters, enhancers, and insulators, and it would specify in which samples each element is active. Specifically targeting this task in the SAGA model could improve results over pan–cell-type annotations assembled from multiple cell-type-specific SAGA models [14].

References

1. Day N, Hemmaplardh A, Thurman RE, Stamatoyannopoulos JA, Noble WS. Unsupervised segmentation of continuous genomic data. Bioinformatics. 2007; 23(11):1424–6. https://doi.org/10.1093/bioinformatics/btm096 PMID: 17384021
2. Ernst J, Kellis M. ChromHMM: automating chromatin-state discovery and characterization. Nat Methods. 2012; 9(3):215–6. https://doi.org/10.1038/nmeth.1906 PMID: 22373907
3. Libbrecht MW, Ay F, Hoffman MM, Gilbert DM, Bilmes JA, Noble WS. Joint annotation of chromatin state and chromatin conformation reveals relationships among domain types and identifies domains of cell-type-specific expression. Genome Res. 2015; 25(4):544–57. https://doi.org/10.1101/gr.184341.114 PMID: 25677182
4. Arneson A, Ernst J. Systematic discovery of conservation states for single-nucleotide annotation of the human genome. Commun Biol. 2019; 2(1):248. https://doi.org/10.1038/s42003-019-0488-1 PMID: 31286065
5. Poulet A, Li B, Dubos T, Rivera-Mulia JC, Gilbert DM, Qin ZS. RT States: systematic annotation of the human genome using cell type-specific replication timing programs. Bioinformatics. 2019; 35(13):2167–76. https://doi.org/10.1093/bioinformatics/bty957 PMID: 30475980

6. Mendez M FANTOM Consortium Main Contributors, Scott MS, Hoffman MM. Unsupervised analysis of multi-experiment transcriptomic patterns with SegRNA identifies unannotated transcripts. bioRxiv. 2020;2020.07.28.225193. https://doi.org/10.1101/2020.07.28.225193
7. Filion GJ, van Bemmel JG, Braunschweig U, Talhout W, Kind J, Ward LD, et al. Systematic protein location mapping reveals five principal chromatin types in *Drosophila* cells. Cell. 2010; 143(2):212–24. https://doi.org/10.1016/j.cell.2010.09.009 PMID: 20888037
8. Ernst J, Kellis M. Discovery and characterization of chromatin states for systematic annotation of the human genome. Nat Biotechnol. 2010; 28(8):817–25. https://doi.org/10.1038/nbt.1662 PMID: 20657582
9. Hoffman MM, Ernst J, Wilder SP, Kundaje A, Harris RS, Libbrecht M, et al. Integrative annotation of chromatin elements from ENCODE data. Nucleic Acids Res. 2012; 41(2):827–41. https://doi.org/10.1093/nar/gks1284 PMID: 23221638
10. Kundaje A, Meuleman W, Ernst J, Bilenky M, Yen A, Heravi-Moussavi A, et al. Integrative analysis of 111 reference human epigenomes. Nature. 2015; 518(7539):317–30. https://doi.org/10.1038/nature14248 PMID: 25693563
11. Ernst J, Kellis M. Chromatin-state discovery and genome annotation with ChromHMM. Nat Protoc. 2017; 12(12):2478–92. https://doi.org/10.1038/nprot.2017.124 PMID: 29120462
12. Hoffman MM, Buske OJ, Wang J, Weng Z, Bilmes JA, Noble WS. Unsupervised pattern discovery in human chromatin structure through genomic segmentation. Nat Methods. 2012; 9(5):473–6. https://doi.org/10.1038/nmeth.1937 PMID: 22426492
13. Chan RC, Libbrecht MW, Roberts EG, Bilmes JA, Noble WS, Hoffman MM. Segway 2.0: Gaussian mixture models and minibatch training. Bioinformatics. 2018; 34(4):669–71. https://doi.org/10.1093/bioinformatics/btx603 PMID: 29028889
14. Libbrecht MW, Rodriguez OL, Weng Z, Bilmes JA, Hoffman MM, Noble WS. A unified encyclopedia of human functional DNA elements through fully automated annotation of 164 human cell types. Genome Biol. 2019; 20:180. https://doi.org/10.1186/s13059-019-1784-2 PMID: 31462275
15. Biesinger J, Wang Y, Xie X. Discovering and mapping chromatin states using a tree hidden Markov model. BMC Bioinformatics. 2013; 14(Suppl 5):S4. https://doi.org/10.1186/1471-2105-14-S5-S4 PMID: 23734743
16. Song J, Chen KC. Spectacle: fast chromatin state annotation using spectral learning. Genome Biol. 2015; 16:33. https://doi.org/10.1186/s13059-015-0598-0 PMID: 25786205
17. Sohn KA, Ho JW, Djordjevic D, Jeong H, Park PJ, Kim JH. hiHMM: Bayesian non-parametric joint inference of chromatin state maps. Bioinformatics. 2015; 31(13):2066–74. https://doi.org/10.1093/bioinformatics/btv117 PMID: 25725496
18. Zerbino DR, Wilder SP, Johnson N, Juettemann T, Flicek PR. The Ensembl regulatory build. Genome Biol. 2015; 16:56. https://doi.org/10.1186/s13059-015-0621-5 PMID: 25887522
19. Mammana A, Chung HR. Chromatin segmentation based on a probabilistic model for read counts explains a large portion of the epigenome. Genome Biol. 2015; 16:151. https://doi.org/10.1186/s13059-015-0708-z PMID: 26206277
20. Libbrecht MW, Hoffman MM, Bilmes JA, Noble WS. Entropic graph-based posterior regularization. In: Proceedings of the International Conference on Machine Learning; 2015. p. 1992–2000.
21. Zhang Y, An L, Yue F, Hardison RC. Jointly characterizing epigenetic dynamics across multiple human cell types. Nucleic Acids Res. 2016; 44(14):6721–31. https://doi.org/10.1093/nar/gkw278 PMID: 27095202
22. Zhang Y, Hardison RC. Accurate and reproducible functional maps in 127 human cell types via 2D genome segmentation. Nucleic Acids Res. 2017; 45(17):9823–36. https://doi.org/10.1093/nar/gkx659 PMID: 28973456
23. Zhang Y, Mahony S. Direct prediction of regulatory elements from partial data without imputation. PLoS Comput Biol. 2019; 15(11):e1007399. https://doi.org/10.1371/journal.pcbi.1007399 PMID: 31682602
24. Xiang G, Keller CA, Heuston E, Giardine BM, An L, Wixom AQ, et al. An integrative view of the regulatory and transcriptional landscapes in mouse hematopoiesis. Genome Res. 2020; 30(3):472–84. https://doi.org/10.1101/gr.255760.119 PMID: 32132109
25. Zacher B, Michel M, Schwalb B, Cramer P, Tresch A, Gagneur J. Accurate promoter and enhancer identification in 127 ENCODE and roadmap epigenomics cell types and tissues by GenoSTAN. PLoS ONE. 2017; 12(1):e0169249. https://doi.org/10.1371/journal.pone.0169249 PMID: 28056037
26. Marco E, Meuleman W, Huang J, Glass K, Pinello L, Wang J, et al. Multi-scale chromatin state annotation using a hierarchical hidden Markov model. Nat Commun. 2017; 8:15011. https://doi.org/10.1038/ncomms15011 PMID: 28387224

27. Girimurugan SB, Liu Y, Lung PY, Vera DL, Dennis JH, Bass HW, et al. iSeg: an efficient algorithm for segmentation of genomic and epigenomic data. BMC Bioinformatics. 2018; 19:131. https://doi.org/10.1186/s12859-018-2140-3 PMID: 29642840

28. Coetzee SG, Ramjan Z, Dinh HQ, Berman BP, Hazelett DJ. StateHub-StatePaintR: rapid and reproducible chromatin state evaluation for custom genome annotation. F1000Res. 2020; 7(214):214.

29. Benner P, Vingron M. ModHMM: A modular supra-Bayesian genome segmentation method. J Comput Biol. 2020; 27(4):442–57. https://doi.org/10.1089/cmb.2019.0280 PMID: 31891534

30. Wang Y, Zhang Y, Zhang R, van Schaik T, Zhang L, Sasaki T, et al. SPIN reveals genome-wide landscape of nuclear compartmentalization. bioRxiv. 2020;2020.03.09.982967. https://doi.org/10.1101/2020.03.09.982967

31. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012; 489(7414):57–74. https://doi.org/10.1038/nature11247 PMID: 22955616

32. Zitnik M, Nguyen F, Wang B, Leskovec J, Goldenberg A, Hoffman MM. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. Inf Fusion. 2019; 50:71–91. https://doi.org/10.1016/j.inffus.2018.09.012 PMID: 30467459

33. ENCODE Project Consortium et al. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature. 2007; 447(7146):799–816. https://doi.org/10.1038/nature05874 PMID: 17571346

34. Wold B, Myers RM. Sequence census methods for functional genomics. Nat Methods. 2007; 5(1):19–21. https://doi.org/10.1038/nmeth1157 PMID: 18165803

35. Barski A, Cuddapah S, Cui K, Roh TY, Schones DE, Wang Z, et al. High-resolution profiling of histone methylations in the human genome. Cell. 2007; 129(4):823–37. https://doi.org/10.1016/j.cell.2007.05.009 PMID: 17512414

36. Skene PJ, Henikoff S. An efficient targeted nuclease strategy for high-resolution mapping of DNA binding sites. elife. 2017; 6:e21856. https://doi.org/10.7554/eLife.21856 PMID: 28079019

37. Boyle AP, Davis S, Shulha HP, Meltzer P, Margulies EH, Weng Z, et al. High-resolution mapping and characterization of open chromatin across the genome. Cell. 2008; 132(2):311–22. https://doi.org/10.1016/j.cell.2007.12.014 PMID: 18243105

38. Hesselberth JR, Chen X, Zhang Z, Sabo PJ, Sandstrom R, Reynolds AP, et al. Global mapping of protein-DNA interactions in vivo by digital genomic footprinting. Nat Methods. 2009; 6(4):283–9. https://doi.org/10.1038/nmeth.1313 PMID: 19305407

39. Buenrostro JD, Giresi PG, Zaba LC, Chang HY, Greenleaf WJ. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. Nat Methods. 2013; 10(12):1213–8. https://doi.org/10.1038/nmeth.2688 PMID: 24097267

40. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The sequence alignment/map format and SAMtools. Bioinformatics. 2009; 25(16):2078–9. https://doi.org/10.1093/bioinformatics/btp352 PMID: 19505943

41. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, et al. The human genome browser at UCSC. Genome Res. 2002; 12(6):996–1006. https://doi.org/10.1101/gr.229102 PMID: 12045153

42. Pohl A, Beato M. bwtool: a tool for bigWig files. Bioinformatics. 2014; 30(11):1618–9. https://doi.org/10.1093/bioinformatics/btu056 PMID: 24489365

43. Schuettengruber B, Ganapathi M, Leblanc B, Portoso M, Jaschek R, Tolhuis B, et al. Functional anatomy of polycomb and trithorax chromatin landscapes in *Drosophila* embryos. PLoS Biol. 2009; 7(1):e1000013. https://doi.org/10.1371/journal.pbio.1000013 PMID: 19143474

44. Kharchenko PV, Alekseyenko AA, Schwartz YB, Minoda A, Riddle NC, Ernst J, et al. Comprehensive analysis of the chromatin landscape in *Drosophila melanogaster*. Nature. 2011; 471(7339):480–5. https://doi.org/10.1038/nature09725 PMID: 21179089

45. Xiang G, Keller CA, Giardine B, An L, Li Q, Zhang Y, et al. S3norm: simultaneous normalization of sequencing depth and signal-to-noise ratio in epigenomic data. Nucleic Acids Res. 2020; 48(8):e43. https://doi.org/10.1093/nar/gkaa105 PMID: 32086521

46. Bayat F, Libbrecht M. Variance-stabilized units for sequencing-based genomic signals. bioRxiv. 2020;2020.01.31.929174. https://doi.org/10.1101/2020.01.31.929174

47. Larson JL, Huttenhower C, Quackenbush J, Yuan GC. A tiered hidden Markov model characterizes multi-scale chromatin states. Genomics. 2013; 102(1):1–7. https://doi.org/10.1016/j.ygeno.2013.03.009 PMID: 23570996

48. Taudt A, Nguyen MA, Heinig M, Johannes F, Colome-Tatche M. chromstaR: Tracking combinatorial chromatin state dynamics in space and time. bioRxiv. 2016;038612. https://doi.org/10.1101/038612

49. Zehnder T, Benner P, Vingron M. Predicting enhancers in mammalian genomes using supervised hidden Markov models. BMC Bioinformatics. 2019; 20:157. https://doi.org/10.1186/s12859-019-2708-6 PMID: 30917778

50. Hamada M, Ono Y, Fujimaki R, Asai K. Learning chromatin states with factorized information criteria. Bioinformatics. 2015; 31(15):2426–33. https://doi.org/10.1093/bioinformatics/btv163 PMID: 25810430

51. Zhou J, Troyanskaya OG. Probabilistic modelling of chromatin code landscape reveals functional diversity of enhancer-like chromatin states. Nat Commun. 2016; 7:10528. https://doi.org/10.1038/ncomms10528 PMID: 26841971

52. Derrien T, Estellé J, Sola SM, Knowles DG, Raineri E, Guigó R, et al. Fast computation and applications of genome mappability. PLoS ONE. 2012; 7(1):e30377. https://doi.org/10.1371/journal.pone.0030377 PMID: 22276185

53. Karimzadeh M, Ernst C, Kundaje A, Hoffman MM. Umap and Bismap: quantifying genome and methylome mappability. Nucleic Acids Res. 2018; 46(20):e120. https://doi.org/10.1093/nar/gky677 PMID: 30169659

54. Lian H, Thompson WA, Thurman R, Stamatoyannopoulos JA, Noble WS, Lawrence CE. Automated mapping of large-scale chromatin structure in ENCODE. Bioinformatics. 2008; 24(17):1911–6. https://doi.org/10.1093/bioinformatics/btn335 PMID: 18591192

55. Ernst J, Kellis M. Large-scale imputation of epigenomic datasets for systematic annotation of diverse human tissues. Nat Biotechnol. 2015; 33(4):364–76. https://doi.org/10.1038/nbt.3157 PMID: 25690853

56. Durham TJ, Libbrecht MW, Howbert JJ, Bilmes J, Noble WS. PREDICTD parallel epigenomics data imputation with cloud-based tensor decomposition. Nat Commun. 2018; 9(1):1402. https://doi.org/10.1038/s41467-018-03635-9 PMID: 29643364

57. Schreiber J, Durham T, Bilmes J, Noble WS. Avocado: a multi-scale deep tensor factorization method learns a latent representation of the human epigenome. Genome Biol. 2020; 21:81. https://doi.org/10.1186/s13059-020-01977-6 PMID: 32228704

58. Dean T, Kanazawa K. A model for reasoning about persistence and causation. Comput Intell. 1989; 5 (2):142–50. https://doi.org/10.1111/j.1467-8640.1989.tb00324.x

59. Bilmes JA. What HMMs can do. IEICE Trans Inf Syst. 2006; 89(3):869–91. https://doi.org/10.1093/ietisy/e89-d.3.869

60. Yoon BJ. Hidden Markov models and their applications in biological sequence analysis. Curr Genomics. 2009; 10(6):402–15. https://doi.org/10.2174/138920209789177575 PMID: 20190955

61. Ferguson JD. Variable duration models for speech. Proceedings of Symposium on the Application of Hidden Markov Models to Text and Speech. 1980. p. 143–79.

62. Levinson SE. Continuously variable duration hidden Markov models for automatic speech recognition. Comput Speech Lang. 1986; 1(1):29–45.

63. Viterbi A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans Inf Theory. 1967; 13(2):260–9.

64. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Series B Stat Methodol. 1977; 39(1):1–22.

65. Baum LE, Petrie T, Soules G, Weiss N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat. 1970; 41(1):164–71.

66. Chan RC, McNeil M, Roberts EG, Mendez M, Libbrecht MW, Hoffman MM. Semi-supervised segmentation and genome annotation. bioRxiv. 2020;2020.01.30.926923. https://doi.org/10.1101/2020.01.30.926923

67. Dixon JR, Selvaraj S, Yue F, Kim A, Li Y, Shen Y, et al. Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature. 2012; 485(7398):376–80. https://doi.org/10.1038/nature11082 PMID: 22495300

68. Mitra S, Zhong J, MacAlpine D, Hartemink AJ. RoboCOP: Jointly computing chromatin occupancy profiles for numerous factors from chromatin accessibility data. bioRxiv. 2020;2020.06.03.132001. https://doi.org/10.1101/2020.06.03.132001

69. Rhee HS, Pugh BF. Comprehensive genome-wide protein-DNA interactions detected at single-nucleotide resolution. Cell. 2011; 147(6):1408–19. https://doi.org/10.1016/j.cell.2011.11.013 PMID: 22153082

70. He Q, Johnston J, Zeitlinger J. ChIP-nexus enables improved detection of *in vivo* transcription factor binding footprints. Nat Biotechnol. 2015; 33(4):395. https://doi.org/10.1038/nbt.3121 PMID: 25751057

71. Codogno M, Fissore L. Duration modelling in finite state automata for speech recognition and fast speaker adaptation. In: ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 12. IEEE; 1987. p. 1269–72.

72. Fujimaki R, Morinaga S. Factorized Asymptotic Bayesian Inference for Mixture Modeling. In: Lawrence ND, Girolami M, editors. Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics. vol. 22 of Proceedings of Machine Learning Research. La Palma, Canary Islands; 2012. p. 400–8.

73. Cowan N. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. Behav Brain Sci. 2001; 24(1):87–114. https://doi.org/10.1017/s0140525x01003922 PMID: 11515286

74. Ernst J, Kheradpour P, Mikkelsen TS, Shoresh N, Ward LD, Epstein CB, et al. Mapping and analysis of chromatin state dynamics in nine human cell types. Nature. 2011; 473(7345):43–9. https://doi.org/10.1038/nature09906 PMID: 21441907

75. Roberts EG, Mendez M, Viner C, Karimzadeh M, Chan RC, Ancar R, et al. Semi-automated genome annotation using epigenomic data and Segway. bioRxiv. 2016;080382. https://doi.org/10.1101/080382

76. Church DM, Schneider VA, Graves T, Auger K, Cunningham F, Bouk N, et al. Modernizing reference genome assemblies. PLoS Biol. 2011; 9(7):e1001091. https://doi.org/10.1371/journal.pbio.1001091 PMID: 21750661

77. Frankish A, Diekhans M, Ferreira AM, Johnson R, Jungreis I, Loveland J, et al. GENCODE reference annotation for the human and mouse genomes. Nucleic Acids Res. 2019; 47(D1):D766–73. https://doi.org/10.1093/nar/gky955 PMID: 30357393

78. Zhou X, Maricque B, Xie M, Li D, Sundaram V, Martin EA, et al. The human epigenome browser at Washington University. Nat Methods. 2011; 8(12):989–90. https://doi.org/10.1038/nmeth.1772 PMID: 22127213

79. Zerbino DR, Achuthan P, Akanni W, Amode MR, Barrell D, Bhai J, et al. Ensembl 2018. Nucleic Acids Res. 2017; 46(D1):D754–61.

80. Buske OJ, Hoffman MM, Ponts N, Le Roch KG, Noble WS. Exploratory analysis of genomic segmentations with Segtools. BMC Bioinformatics. 2011; 12:415. https://doi.org/10.1186/1471-2105-12-415 PMID: 22029426

81. Ramírez F, Dündar F, Diehl S, Grüning BA, Manke T. deepTools: a flexible platform for exploring deep-sequencing data. Nucleic Acids Res. 2014; 42(W1):W187–91. https://doi.org/10.1093/nar/gku365 PMID: 24799436

82. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010; 26(6):841–2. https://doi.org/10.1093/bioinformatics/btq033 PMID: 20110278

83. Kircher M, Witten DM, Jain P, O'Roak BJ, Cooper GM, Shendure J. A general framework for estimating the relative pathogenicity of human genetic variants. Nat Genet. 2014; 46(3):310–5. https://doi.org/10.1038/ng.2892 PMID: 24487276

84. Rao SS, Huntley MH, Durand NC, Stamenova EK, Bochkov ID, Robinson JT, et al. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell. 2014; 159(7):1665–80. https://doi.org/10.1016/j.cell.2014.11.021 PMID: 25497547

85. Zeng X, Li B, Welch R, Rojo C, Zheng Y, Dewey CN, et al. Perm-seq: mapping protein-DNA interactions in segmental duplication and highly repetitive regions of genomes with prior-enhanced read mapping. PLoS Comput Biol. 2015; 11(10):e1004491. https://doi.org/10.1371/journal.pcbi.1004491 PMID: 26484757

86. Yen A, Kellis M. Systematic chromatin state comparison of epigenomes associated with diverse properties including sex and tissue type. Nat Commun. 2015; 6:7973. https://doi.org/10.1038/ncomms8973 PMID: 26282110

87. Ebert P, Schulz MH. Fast detection of differential chromatin domains with SCIDDO. Bioinformatics. 37.9 (2021): 1198–1205 https://doi.org/10.1093/bioinformatics/btaa960

88. Chen B, Kenari NS, Libbrecht MW. Continuous chromatin state feature annotation of the human epigenome. bioRxiv. 2018;473017. https://doi.org/10.1101/473017