ROCCO: A Robust Method for Detection of Open Chromatin via Convex Optimization

Nolan H. Hamilton Terrence S. Furey

University of North Carolina at Chapel Hill
Department of Genetics

Abstract

Motivation

Analysis of open chromatin regions across multiple samples from two or more distinct conditions can determine altered gene regulatory patterns associated with biological phenotypes and complex traits. The ATAC-seq assay allows for tractable genome-wide open chromatin profiling of large numbers of samples. Stable, broadly applicable genomic annotations of open chromatin regions are not available. Thus, most studies first identify open regions, or *peaks*, using peak calling methods for each sample independently. These are then heuristically combined to obtain a consensus peak set. Reconciling sample-specific peak results *post hoc* from a large cohort is challenging, and informative spatial features specific to open chromatin signals are not leveraged effectively.

Results

We propose a novel method, ROCCO, that determines consensus open chromatin regions across multiple samples simultaneously. ROCCO uses robust summary statistics across samples by solving a constrained optimization problem formulated to account for both enrichment and spatial features of open chromatin signal data. We show this formulation admits attractive theoretical and conceptual properties as well as superior empirical performance compared to current methodology.

Availability and Implementation

An implementation of ROCCO in Python, along with documentation and usage demos, is available on GitHub.

Contact

nolanh@email.unc.edu or tsfurey@email.unc.edu.

Supplementary Information

Supplementary material is available with this submission. Additional resources that may aid readers are available in the ROCCO GitHub repository.

1 Introduction

Nucleosomes, complexes of DNA and histone proteins, comprise the initial stage of chromatin compaction of the genome, reducing its occupying volume and enabling it to fit in cell nuclei (Li and Reinberg, 2011). Most nucleosomal DNA is inaccessible for binding by transcription factors (TFs) that regulate gene expression. Genome-wide annotations of non-nucleosomal DNA, or open chromatin, therefore delineate where TFs can readily bind and effectively characterize the current gene regulatory program in a sample. Open chromatin landscapes vary across cell types and conditions, including in disease (Corces and Granja, 2018) reflecting cell and condition specific gene regulation. To better understand this dynamic nature of gene regulation, identification of open chromatin across cell and conditions has become an important aspect of molecular studies of complex phenotypes.

Several assays have been developed for genome-wide measurement of open chromatin, including DNase-seq (Boyle et al., 2008) and ATAC-seq (Buenrostro et al., 2015). These assays generate DNA fragments enriched for open chromatin regions that are then sequenced using short-read sequencers. Resulting reads are aligned to a reference genome, and regions with an enrichment of reads, or *peaks*, are identified as open chromatin. Peak calling is a necessary step as, unlike genes for which comprehensive annotations are available for many species, there do not exist pre-defined standard databases of open chromatin regions.

Studies focused on determining changes in chromatin associated with differing cellular conditions or complex traits typically include multiple samples. For these studies, it is necessary to define a common

set of open chromatin regions, or *consensus peaks*, to facilitate comparisons across sample groups. Typically, consensus peaks are determined by first annotating peaks independently in each sample. Then, these sample-specific peaks are merged based on one of several heuristics including: (1) Simply take the maximal set across all samples. This method, though, is particularly vulnerable to anomalous data since peaks from a single sample satisfy the inclusion criterion; (2) Include only peak regions that occur in *all* samples. This is usually too stringent, especially when there is an expectation that there will be differences; (3) Require that peaks be present in at least $M = 1 \dots K$ samples, where the boundaries of the consensus peaks allow for some tolerance, T, for disparity in nucleotide position. Protocols in the spirit of this general method have been utilized in many open chromatin studies (Bao et al., 2015; Wang et al., 2018; Ming et al., 2020; Bentsen et al., 2020). A difficulty in applying such methods is choosing appropriate M and T—a task manifesting a rigid criteria that may ignore some open regions or may include spurious regions and that may not define well-supported peak boundaries.

More statistically sound methods have been developed for handling multiple samples. For the specific case of $K = 2$ samples, the Irreproducible Discovery Rate (Li et al., 2011) can be controlled to mitigate calling of irreproducible peaks. However, since most experimental designs include $K >> 2$ samples per group, it is difficult to apply this method. Alternatively, Genrich[1] offers a method for multiple samples in which p-values are combined using Fisher's Method (Fisher, 1925). While Genrich has been used successfully in several studies (Hofvander et al., 2019; Salavati et al., 2021; Tsaryk et al., 2022), the independence assumption of Fisher's Method may be problematic for large numbers of samples, especially in cases involving multiple technical replicates (Roy et al., 2019). It also does not explicitly account for specific peak boundaries.

Here, we propose a novel method for identification of open chromatin regions across multiple samples, `ROCCO`: "**R**obust **O**pen **C**hromatin detection via **C**onvex **O**ptimization." This method offers several favorable features:

- Accounts for both enrichment *and* spatial characteristics of open chromatin signals, the latter of which is an informative but often ignored aspect of ATAC-seq data that can be used to not only better detect regions but also improves on annotating peak boundaries;
- Leverages data from multiple samples without imposing arbitrary "hard" thresholds on a minimum number of samples declaring peaks;
- Is efficient for large numbers of samples;
- Does not require training data or a heuristically determined set of initial candidate regions, which are hard to define given the the lack of a priori sets of open chromatin regions;
- Employs a mathematically tractable model granting useful performance guarantees.

We formally describe an algorithm implemented by `ROCCO` for the consensus peak problem and present a theoretical analysis. We then conduct several experiments to investigate `ROCCO`'s efficacy empirically, including on a set of 56 samples from human lymphoblastoid cell lines.

2 Algorithm

We begin by introducing notation used throughout this manuscript and describing the structure of signal data used by `ROCCO` to detect accessible chromatin.

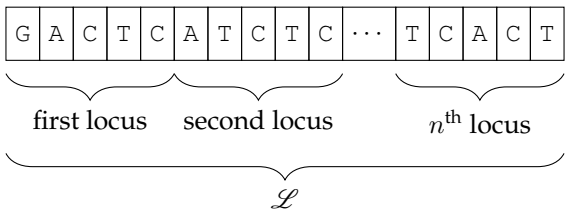

Figure 1: Genomic region $\mathscr{L}$ consisting of n fixed-width loci containing $L = 5$ nucleotides each.

2.1 Notation and Definitions

Let $\mathscr{L}$ be a contiguous genomic sequence, e.g., a chromosome, divided into n fixed-width loci, each consisting of L nucleotides as in Figure 1. For each sample j, we assume access to a signal, s_{i_j}, computed as a function

[1] Available at `https://github.com/jsh58/Genrich`

Symbol	Description	Default Value
b	budget threshold on selected loci	.035
c_1, c_2, c_3	weights for score function $\mathcal{S}(i)$	1.0
f	objective function	N/A
γ	penalty weight in (3)	1.0
K	number of input samples	N/A
ℓ_i	decision variable for i^{th} locus	N/A
$\boldsymbol{\ell}$	n-dimensional vector of decision variables	N/A
L	fixed interval size of input signals	50
$\mathscr{L}$	contiguous genomic region	N/A
n	number of loci	N/A
N	max. randomized rounding iterations	50
$\mathcal{S}(\cdot)$	locus score function	N/A
$\mathbf{S}_{\mathscr{L}}$	$K \times n$ signal matrix over $\mathscr{L}$	N/A
τ	median enrichment threshold in $\mathcal{S}(\cdot)$	0

Table 1: Notation Reference

of observed enrichment based on sequence reads at the ith locus. For K total samples, this yields a $K \times n$ signal matrix used as input to ROCCO:

$$\mathbf{S}_{\mathscr{L}} = \begin{pmatrix} s_{1_1} & s_{2_1} & \dots & s_{n_1} \\ s_{1_2} & s_{2_2} & \dots & s_{n_2} \\ \dots & \dots & \dots & \dots \\ s_{1_k} & s_{2_k} & \dots & s_{n_K} \end{pmatrix} = \begin{pmatrix} \mathbf{s_1} & \mathbf{s_2} & \dots & \mathbf{s_n} \end{pmatrix} \tag{1}$$

with $\mathbf{s_i} \in \mathbb{R}^K$ denoting the column vector of signal values among samples at the ith locus. This matrix can be generated with a variety of methods, but we offer a context-specific tool, `prep_bams.py` in the GitHub Repository, to generate $\mathbf{S}_{\mathscr{L}}$ by calling PEPATAC's (Smith et al., 2021) `bamSitesToWig.py` script on each sample's BAM file.

2.2 Scoring Loci

To determine consensus open chromatin regions, we first score each locus while accounting for enrichment (g_1), dispersion among samples (g_2), and a measure of local volatility in enrichment (g_3).

Specifically, we take $g_1(i)$ to be the median, and $g_2(i)$ to be the median absolute deviation (Pham-Gia and Hung, 2001) of the K signal values at the ith locus:

$$g_1(i) = \text{med}\left\{s_{i_1}, s_{i_2}, \dots, s_{i_K}\right\}$$
$$g_2(i) = \text{med}\left\{|s_{i_1} - g_1(i)|, \dots, |s_{i_K} - g_1(i)|\right\}.$$

Large g_1 and low g_2 correspond to regions of high enrichment with little dispersion among samples—a favorable combination of traits to emphasize when predicting accessibility. We also leverage the disparities (L_1-norm) between enrichment signal values at adjacent loci, normalized by the current locus's enrichment,

$$g_3(i) = \frac{1}{g_1(i) + 1} \max\Big\{|g_1(i) - g_1(i+1)|, |g_1(i) - g_1(i-1)|\Big\}.$$

A fundamental aim of g_3 is to more precisely annotate the edges and immediately adjacent regions of peaks where signals may be low before or after an abrupt shift in enrichment characterizing the nearby peak. See Figure 2 for a visual demonstration on an idealized, continuous enrichment signal. Note, g_3 is redefined at the chromosome ends (that is, $i = 1$, $i = n$) to consider only the subsequent or previous locus, respectively.

In (2), we define the score piece-wise and take a simple linear combination of g_1, g_2, g_3—or 0 if median enrichment is below a defined threshold. This enrichment threshold can encourage sparsity that may have favorable implications for computational efficiency during optimization and mitigate consideration of regions that are unlikely accessible.

$$\mathcal{S}(i) = \begin{cases} c_1 g_1(i) - c_2 g_2(i) + c_3 g_3(i) & \text{if } g_1(i) \geq \tau \\ 0 & \text{if } g_1(i) < \tau \end{cases} \tag{2}$$

where $\tau \geq 0$ is the threshold placed on the median signal, and $c_1, c_2, c_3 \geq 0$ are coefficients for each term. By default, each scoring term has the following weights: $c_1, c_2, c_3 = 1$, and $\tau = 0$. These defaults provide strong performance but can be modified in the software implementation of ROCCO to suit users' specific needs. For example, users desiring more conservative peak predictions may wish to set $\tau > 0$.

$\mathcal{S}(i) \leq 0$ does not necessarily preclude the corresponding locus from being included within a consensus peak since the objective function (See Eqn. 3) is not completely dependent on the locus score. Note that for default parameters, $\mathcal{S}(i) \geq 0$.

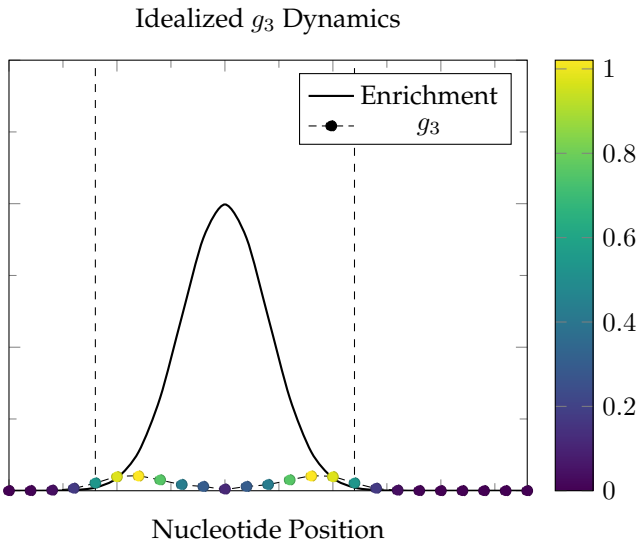

Figure 2: g_3 is designed to more accurately determine peak edges to capture enriched regions in their whole. This function is greatest (yellow) near the ends of the enriched region in this idealized example.

2.3 Optimization

We address open chromatin detection as a constrained optimization problem. Let $\boldsymbol{\ell} \in \{0,1\}^n$ be a vector of binary decision variables, where we label $\ell_i = 1$ if the ith locus is present in open chromatin, and $\ell_i = 0$ otherwise.

We impose an explicit *budget constraint* controlling the total number of selected loci in a given input chromosome. Let $b \in [0,1]$ be the maximum percent of loci that can be selected, i.e.,

$$\sum_{i=1}^{n} \ell_i \leq \lfloor nb \rfloor.$$

This constraint controls sensitivity in peak predictions and prevents unrealistic solutions in which an excessive fraction of chromatin is declared open. With estimates for the fraction of accessible chromatin hovering around $3-4\%$ of the human genome (Song et al., 2011; Sahinyan et al., 2022), we accordingly set $b = .035$ as the default value. Since the budget applies to each chromosome independently, and the accessibility for each chromosome can vary, the software implementations allows for chromosome-specific parameters to be defined.

To optimize selection of accessible regions, the following objective function is *minimized*:

$$f(\boldsymbol{\ell}) = \underbrace{\sum_{i=1}^{n} -(\mathcal{S}(i) \cdot \ell_i)}_{\text{reward high locus scores}} + \underbrace{\gamma \sum_{i=1}^{n-1} |\ell_i - \ell_{i+1}|}_{\gamma \text{ controls influence of adjacent loci}} \tag{3}$$

The first term rewards loci with high $\mathcal{S}$ scores, e.g., those with consistently high enrichment across samples or those on the edges of greatly enriched regions. The second term is introduced to account for spatial proximity of loci during optimization and controls the influence of signals in adjacent loci: For a given budget b, as γ is increased (greater influence of adjacent regions), fewer but longer *distinct* regions are annotated as open, yielding simpler solutions in a topological sense. This pattern is exhibited in Figure 3.

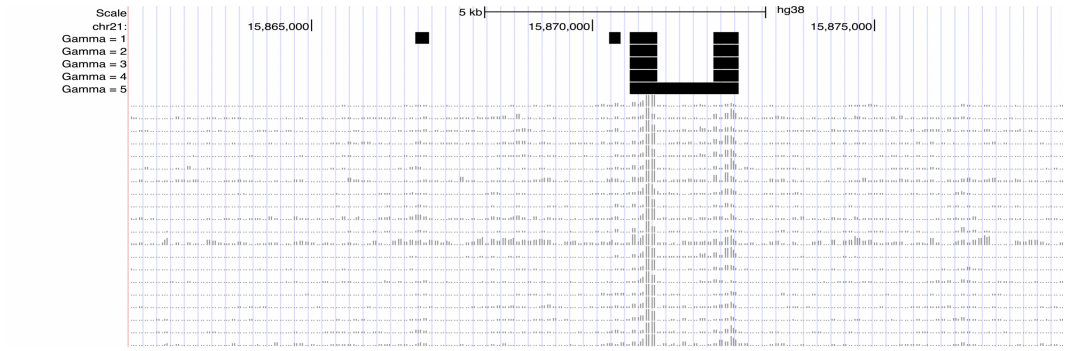

Figure 3: Example behavior of ROCCO in the UCSC Genome Browser (Karolchik et al., 2003) as γ is increased. The black bars in each track correspond to ROCCO's predictions given the replicate signal data below. In the last row, we see that two distinct regions of enrichment are merged due to the strong influence of adjacent loci imposed by the $\gamma = 5$ parameter.

To incorporate the described objective and budget constraint, we pose the following constrained optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\ell}}{\text{Minimize:}} \quad & f_{\text{IP}}(\boldsymbol{\ell}) = \sum_{i=1}^{n} -(\mathcal{S}(i) \cdot \ell_i) + \gamma \sum_{i=1}^{n-1} |\ell_i - \ell_{i+1}| \\ \text{Subject To:} \quad & \text{(i)} \ \sum_{i=1}^{n} \ell_i \leq \lfloor nb \rfloor \\ & \text{(ii)} \ \ell_i \in \{0, 1\}, \ \forall i = 1 \ldots n. \end{aligned} \tag{4}$$

Constraint (ii) restricts the feasible region to integer solutions. In general, such constraints yield difficult optimization problems, for example, because gradients are not defined for functions over the integers and convexity cannot be leveraged. Indeed, general integer programming is known to be NP-hard (Korte and Vygen, 2012). A common remedy is to convert the original, integer-constrained formulation to an analogous problem with convenient analytic properties. Accordingly, we substitute the constraints

$$\ell_i \in \{0, 1\} \rightarrow \ \ell_i \in \mathbb{R} : 0 \leq \ell_i \leq 1$$

to obtain the following convex optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\ell}}{\text{Minimize:}} \quad & f_{\text{CP}}(\boldsymbol{\ell}) = \sum_{i=1}^{n} -(\mathcal{S}(i) \cdot \ell_i) + \gamma \sum_{i=1}^{n-1} |\ell_i - \ell_{i+1}| \\ \text{Subject To:} \quad & \text{(i)} \ \sum_{i=1}^{n} \ell_i \leq \lfloor nb \rfloor. \\ & \text{(ii)} \ \ell_i \in [0, 1], \ \forall i = 1 \ldots n. \end{aligned} \tag{5}$$

As we will see, this formulation maintains the essence of the original problem in (4) and confers several useful properties. In general, convexity is a highly-valued feature in constrained optimization as it ensures every local minimum is also a global minimum, thereby preventing instances of "premature" convergence to suboptimal solutions (Boyd and Vandenberghe, 2004).

Theorem 1. *The problem in (5) can be solved in polynomial time for a globally optimal solution.*

Linear programs (LPs) are a special class of convex optimization problems in which both the objective and constraints are linear functions of the decision variables. Though general convex problems can often be solved efficiently in practice, there are certain intractable instances. In contrast, LPs can be solved in worst-case polynomial time with respect to the number of variables (Boyd and Vandenberghe, 2004). Accordingly, the proof of Theorem 1, deferred to Supplementary Material 1.1, relies on showing that an optimal solution to (5), $\boldsymbol{\ell}^{CP} \in \mathbb{R}^n$, is obtained from the n-dimensional truncation of the optimal solution to the following LP:

$$\begin{aligned} \underset{\boldsymbol{\ell}}{\text{Minimize:}} \quad & f_{\text{LP}}(\boldsymbol{\ell}) = \sum_{i=1}^{n} -(\mathcal{S}(i) \cdot \ell_i) + \gamma \sum_{j=n+1}^{2n-1} \ell_j \\ \text{Subject To:} \quad & \text{(i)} \ \sum_{i=1}^{n} \ell_i \leq \lfloor nb \rfloor \\ & \text{(ii)} \ \ell_i \in [0, 1], \ \forall i = 1 \ldots n \\ & \text{(iii)} \ \ell_j \geq -1 \cdot (\ell_i - \ell_{i+1}), \ \forall i < n, \ j = n + i \\ & \text{(iv)} \ \ell_j \geq +1 \cdot (\ell_i - \ell_{i+1}), \ \forall i < n, \ j = n + i \end{aligned} \tag{6}$$

which we denote as $\boldsymbol{\ell}^{LP} \in \mathbb{R}^{2n-1}$. Moreover, the optimal objective values for (5) and (6) are equal:

$$\min f_{\text{CP}} = \min f_{\text{LP}}.$$

The time complexity of interior-point methods for solving general LPs with n decision variables and a d-bit data representation is $O\left(n^3 d\right)$ (den Hertog, 1994). It is important to note that, in practice, modern solvers offer much greater efficiency than this cubic worst-case bound might suggest by exploiting problem structure (Koch et al., 2022). Indeed, (6) possesses a particularly sparse objective and constraint matrix that allows for surprising efficiency showcased in Supplementary Material 1.3.

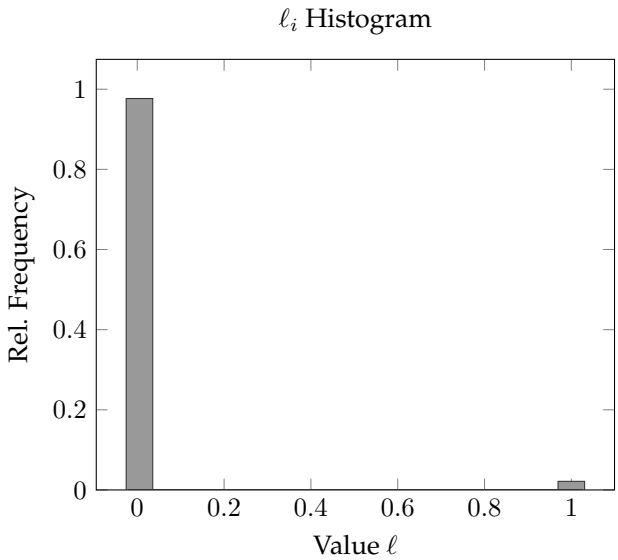

Figure 4: Observed distribution of decision variables after solving (6) with budgets $b \in \{.01, .025, .05, .075, .10\}$ on 50 random subsamples ($K = 40$) of the ATAC-seq data detailed in Section 5.2 and pooling the solutions from each run. More than 99.95% of decision variables are placed in the first and last bins with edges $.005$ and 1.0, respectively. See attached material for exact relative frequencies used to generate this plot.

After solving the relaxed form of an integer program, it is often necessary to refine the solution for feasibility in the original integer-constrained region. Exact combinatorial techniques, such as branch and bound, can be applied to find optimal integer solutions; However, such methods may incur prohibitive computational expense and do not offer polynomial runtime guarantees. It is often more practical to use an approximate rounding scheme (Williamson and Shmoys, 2011).

We find in our own experiments that solutions to (6) are nearly-integral (Figure 4) and can be rounded immediately, for instance, with the floor function, to obtain a feasible solution without a substantial sacrifice in performance. However, this cannot be guaranteed in the general case and a more robust rounding procedure is preferred, which we now describe.

Algorithm 1: Drawing $\boldsymbol{\ell}^{\,rand}$

Input: $\boldsymbol{\ell}^{LP}$
$\boldsymbol{\ell}^{rand} \leftarrow \mathbf{0}$
for $\ell_h \in \boldsymbol{\ell}^{\,rand}$ **do**
  Assign integer decision variable:

$$\ell_h^{rand} \sim \text{Bernoulli}(\ell_h^{LP})$$

end
Return: $\boldsymbol{\ell}^{rand}$

Given a solution to the relaxed formulation (6), we devise a procedure, denoted RR, based on randomized rounding (Raghavan and Tompson, 1987), to obtain a set of candidate integral solutions, $\mathbf{L}_N$. This set is generated by executing N iterations of Algorithm 1, after which RR picks the best observed solution (lowest objective value) satisfying the constraints.

Note that for a given solution space $\mathscr{F} \in \mathbb{R}^D$, $D \geq 1$, integral solutions in $\mathscr{F}$ cannot yield better performance than the best real-valued solution; The set of integral solutions is a proper subset of $\mathscr{F}$. For this reason, the

quality of integer solutions can be judged with reference to $f_{\text{LP}}(\boldsymbol{\ell}^{LP})$. In light of this, the construction of solutions $\boldsymbol{\ell}^{\,rand} \in \mathbf{L}_N$, with each ℓ_i^{rand} defined as a Bernoulli-distributed random variable with parameter $p_i = \ell_i^{LP}$, grants several convenient properties arising from linearity of expectation. Namely,

$$\mathbb{E}\left[f_{\text{LP}}(\boldsymbol{\ell}^{\,rand})\right] = f_{\text{LP}}(\boldsymbol{\ell}^{\,LP})$$

with constraints likewise satisfied in expectation by $\boldsymbol{\ell}^{\,rand}$.

We can use these expected values to make probabilistic assertions regarding the solutions present in $\mathbf{L}_N$.

Theorem 2. *Let $\mathbf{L}_N$ be a set of $N > 1$ random solutions generated with Algorithm 1, and let $c > 1$, $a > 0$ be real numbers satisfying $\frac{1}{c} + e^{-2n(ab)^2} < 1$ for n loci and budget $b \in (0, 1)$. Then with high probability, $\mathbf{L}_N$ contains at least one solution with both (a) an objective value no more than c-times $f_{\text{LP}}(\boldsymbol{\ell}^{LP})$ and (b) no more than $nb(1 + a)$ loci selected.*

The proof of Theorem 2, deferred to Supplementary Material, shows

$$\mathbb{P}\left[\exists\, \boldsymbol{\ell}^{\,rand} \in \mathbf{L}_N :\ f_{\text{LP}}(\boldsymbol{\ell}^{\,rand}) < c \cdot f_{\text{LP}}(\boldsymbol{\ell}^{\,LP}),\ \sum_i^n \ell_i^{rand} < nb(1+a)\right] \geq 1 - \left(\frac{1}{c} + e^{-2n(ab)^2}\right)^N$$

which quickly approaches 1 for increasing N. Note that this expression is a *lower bound* for the probability there exists a solution satisfying both criteria (a,b). We often observed multiple such solutions in $\mathbf{L}_N$ during the course of our experiments. But Theorem 2 allows us to make more general assertions under the supposition $\frac{1}{c} + e^{-2n(ab)^2} < 1$. Since n is on the order of millions for default locus size $L = 50$, this criterion is satisfied even for quite small $c > 1, a > 0$.

Note that RR runs in $\Theta(nN)$ time and does not require storing the whole of $\mathbf{L}_N$ in memory simultaneously since only the current, best-scoring solution needs to be referenced at each iteration for comparison. Users can therefore liberally increase N if desired without incurring a great deal of computational expense. The RR procedure technically renders ROCCO stochastic, but for the default $N = 50$ RR iterations, we observed only minor variation in solutions returned from independent runs of ROCCO. As seen in Supplementary Material (Table S2), a pairwise Jaccard similarity matrix for five independently-generated ROCCO solutions contains values no less than 0.9977.

Algorithm 2 offers a pseudocode representation of ROCCO as a whole.

Algorithm 2: ROCCO

Input: $\mathbf{S}_{chr} \in \mathbb{R}^{K \times n}$
Parameters: $b, \gamma, \tau, c_1, c_2, c_3$

1. Construct the row vector of scores given by (2):
   $\mathcal{S}(i) \leftarrow$ `score`$(\mathbf{S}_{chr}, \tau, c_1, c_2, c_3)$`[i]`
2. Solve the linear program in (6):
   $\boldsymbol{\ell}^{LP} \leftarrow$ `solve`$((6), b, \gamma)$;
3. Refine the LP solution:
   $\boldsymbol{\ell}^{*} \leftarrow$ `round_truncate`$(\boldsymbol{\ell}^{LP}$, `method=RR`$)$

Return: $\boldsymbol{\ell}^{*}$

The object returned by Algorithm 2 is an n-dimensional decision vector, $\boldsymbol{\ell}^{*}$, used to select loci as accessible. Note, contiguous selections (that is, sequences of loci such that $\ell_i^{*} = 1$) are merged into single peaks in the final BED file.

Remark. *ROCCO's scaling for large numbers of input samples.* Note that runtime for only the scoring step is directly affected by the number of input samples, running on the order of nK elementary operations using the median of medians algorithm. Practical scenarios usually satisfy $K << n$, making the number of input samples a minor contributor to computational expense during calculation of $\mathcal{S}(\cdot)$. Moreover, the scoring step altogether is dominated in computational complexity by ROCCO's optimization step, which is independent of K.

3 Results

We performed several experiments to assess ROCCO's detection performance using ATAC-seq data from 56 human lymphoblast samples (see Methods). Additional analyses and details are available in the Supplementary Material.

3.1 Detection Performance

A noteworthy limitation in experiments comparing performance of open chromatin detection methods is a lack of high-confidence open chromatin annotations against which to test. However, to gauge performance and ensure viability, some proxy for ground truth is needed. Following (Zhao and Boyle, 2021), we constructed a *union set*, GT, of conservative IDR (Li et al., 2011) peaks from ENCODE *transcription factor* ChIP-seq experiments in the GM12878 lymphoblast cell line. The $\mathcal{F}_\beta$-score, defined below, was then used to assess the ability to recover the regions in GT using ATAC-seq data from the 56 independent samples. Details regarding construction of the GT data set can be found in Supplementary Material 1.7.

For each method, we generated consensus peaks using previously determined alignments for the $K = 56$ samples. We then computed precision as

$$\mathcal{P} = \frac{|D_{\mathrm{X}} \cap D_{\mathrm{GT}}|}{|D_{\mathrm{X}}|}$$

and recall as

$$\mathcal{R} = \frac{|D_{\mathrm{X}} \cap D_{\mathrm{GT}}|}{|D_{\mathrm{GT}}|},$$

where D_X denotes the consensus peaks obtained from method X and set intersections in the numerators are computed using `bedtools intersect`. The F_β-score was then calculated as the harmonic mean of precision and recall where recall is weighed β times as much as precision, that is:

$$\mathcal{F}_\beta = (1 + \beta^2)\frac{\mathcal{P} \cdot \mathcal{R}}{\beta^2 \mathcal{P} + \mathcal{R}}.$$

As in (Zhao and Boyle, 2021), we use the $\mathcal{F}_\beta$-score as the primary metric for comparison of methods since it intuitively combines both precision and recall and is less affected by extreme regions of the precision-recall curve that do not correspond to realistic use-cases.

3.1.1 Detection Performance: Benchmark Methods

Most methods start with sample-specific peaks. For this step, we employed the widely-used MACS2 software (Gaspar, 2018) using parameters commonly specified for ATAC-seq experiments (see Methods). With these, we used a common heuristic to specify consensus peaks (Yang et al., 2014). Namely, `MACS2-Consensus` only retained merged peaks supported by a majority of samples with a 100bp tolerance in chromosome position across samples. Genrich is another method for consensus peak calling we tested that analyzes samples separately, calculating p-values for each. It then applies Fisher's method to combine p-values at each genomic region. We also generated peak sets with `MACS2-Pooled`, which combined alignments from all samples into one BAM file and then used MACS2 to call peaks on this combined alignment file.

See Supplementary Material 1.8 for exact configurations used to produce results for these MACS2-based methods and Genrich.

3.1.2 Detection Performance: Results

For an initial visual comparison of methods, Figure 5 displays peak calls from each in 100kb and 20kb regions on chromosome 19 in the UCSC Genome Browser (Karolchik et al., 2003). We also include ATAC-seq signals from 25 of the lymphoblast samples being evaluated. As expected, all methods identify regions with consistently strong signal across all samples. They vary, though, in the contiguity and boundaries of these regions. There are also method-specific regions, as well as ones called by multiple, but not all, methods.

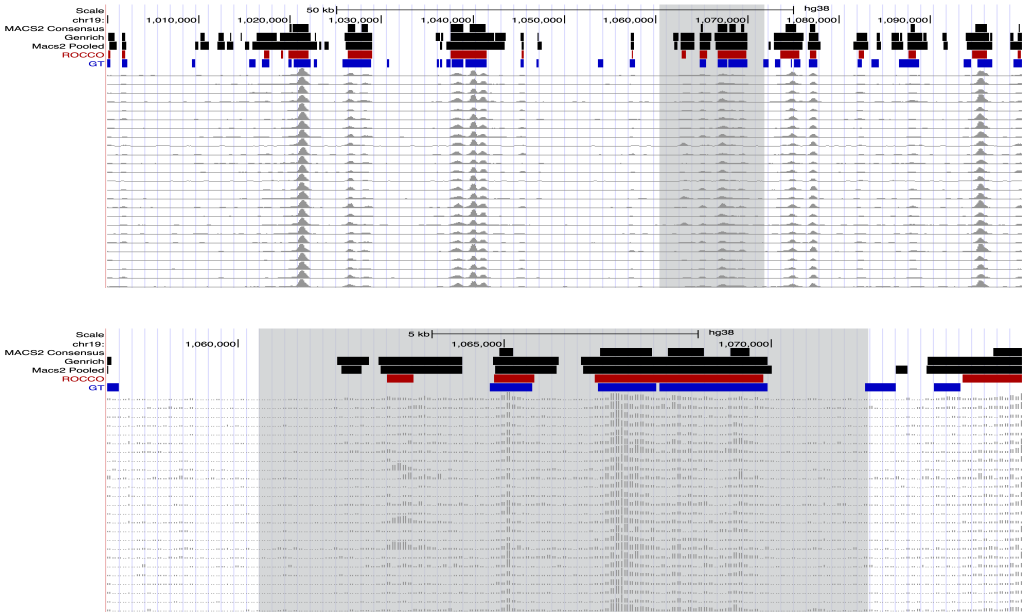

Figure 5: Consensus peak calls from each method tuned using the $\mathcal{F}_\beta$ score ($\beta = 1.0$), displayed at resolutions of 100,000 nucleotides (top) and 20,000 nucleotides (bottom) in a region within chromosome 19.

Detection Performance

β	ROCCO	Genrich	MACS2-Pooled	MACS2-Consensus
0.50	**0.651**	0.460	0.390	0.643
0.75	**0.596**	0.489	0.427	0.586
1.0	**0.579**	0.520	0.465	0.545
1.25	**0.580**	0.545	0.501	0.517
1.50	**0.582**	0.566	0.531	0.498
2.0	**0.603**	0.595	0.577	0.475

Table 2: Performance for each method is recorded after tuning for the $\mathcal{F}_\beta$ value in the leftmost column.

We evaluated across several values

$$\beta \in \{0.5, 0.75, \ldots, 1.50, 2.0\}$$

to address a plausible but encompassing range of recall/precision prioritizations. The most extreme cases $\beta = 0.5, \beta = 2.0$ were included for completeness but may not be particularly well-motivated by realistic usage since the corresponding $\mathcal{F}_\beta$ score can be unduly improved by simply rejecting any uncertain predictions or accepting all plausible predictions, respectively.

For each F_β-score, we tuned each method over a range of significance thresholds deemed reasonable given the methods' underlying models to maximize their performance. For Genrich, we tested

$$p \in \{10^{-6}, 10^{-5}, \ldots, 10^{-1}\}.$$

For the MACS2-based methods, we tested:

$$q \in \{.001, .005, .01, .05, .10, .20\}.$$

For ROCCO, the budget parameter is most fundamental and bounds the fraction of genomic region $\mathscr{L}$ that can be selected. We thus use b as the tuning parameter for ROCCO, leaving $\gamma, \tau, c_{1,2,3}$ as their default values, and evaluated:

$$b \in \{.02, .025, \ldots, .06\}.$$

For each β listed above, given a method X and parameter r, we computed tuned performance as

$$\max_r \mathcal{F}_\beta(X_r),$$

that is, the best observed performance of the method while sweeping its most fundamental parameter. These values are recorded in Table 2. ROCCO matched or exceeded the best performance of every benchmark method for all six β values.

The performance disparity between ROCCO and the second best-scoring method was smallest for the most recall-dependent case $\beta = 2$, which we have stated above is only partially informative and particularly vulnerable to spurious predictions.

As mentioned above, ROCCO allows for specifying chromosome specific parameters to account for differences in accessibility. For a cursory investigation into the effects of this practice, we tuned the budget parameter for $\mathcal{F}_1$ for each chromosome. We observed a non-trivial increase in performance ($F_1 = .620$) compared to a constant budget for all chromosomes ($F_1 = .579$ as in Figure 2), but we expect additional improvements from a more rigorous, technically sound approach in which budgets are not restricted to an arbitrary set of values.

A comparison of methods using their default significance thresholds with no tuning is included in the Supplementary Material S1, where ROCCO offers the greatest $\mathcal{F}_\beta$-score in all but the $\beta = 0.5, \beta = 2.0$ experiments.

3.2 Differential Accessibility Testing with ROCCO

ATAC-seq data is commonly generated from multiple samples within two or more distinct groups. Using these data, a key question is where are regions where accessibility is significantly different, potentially revealing insights into regulatory mechanisms responsible for phenotypic differences between groups. To offer a template for differential analysis with ROCCO, we ran a simple experiment comparing the accessibility landscapes of males and females in the lymphoblast data described in Section 5.2.

To ensure that peaks that may be specific to a particular group are considered, ROCCO can be run independently on each group's respective samples and then merged to produce the final set of consensus peaks. The software implementation of ROCCO offers a convenient means to separate samples by group for peak calling, the merging of peaks, and the generation of a count matrix across all samples that can be used for input to methods such as DESeq2 (Love et al., 2014). A step-by-step Jupyter notebook tutorial for differential analysis with ROCCO is available on the GitHub repository.

In this experiment, ROCCO was run independently on the 23 female and 33 male lymphoblastoid cell lines using default parameters, and the peaks were merged to create a final set of $172{,}933$ consensus peaks. We note that this merged set included $23{,}865$ peaks only detected in the male samples and $19{,}165$ peaks only detected in the female samples. Further, there were $\sim 17{,}000$ peaks in each of these sets that were not included in the consensus set derived from all samples combined. This underscores the importance for calling group-specific peaks.

DESeq2 was then used to detect differentially accessible regions across the merged consensus peaks. At FDR-adjusted $p < 0.05$, $3{,}141$ regions spanning $2{,}275{,}100$ base pairs were identified as differentially accessible. About 93% ($2{,}916$) of these peaks were observed in chromosome X, which is unsurprising given the division of samples into male and female groups.

4 Discussion

In this manuscript, we introduced ROCCO, a novel method for identifying open chromatin regions in ATAC-seq data that simultaneously leverages information from multiple samples to determine a consensus set of peaks. ROCCO uses spatial features of enrichment signal data by formulating the problem with an appropriate convex model that can be solved with provable efficiency and performance guarantees. Importantly, the model accounts for features common to the edges of accessible chromatin regions, which are often hard to determine based on independently determined sample peaks that can vary widely in their genomic locations. In addition to several attractive conceptual and theoretical features, ROCCO also exhibited improved detection performance based on ATAC-seq data from 56 lymphoblastoid samples evaluated against known transcription factor binding sites determined using ChIP-seq. ROCCO is especially suited for experimental designs that include samples from two or more distinct groups with one goal being to determine regions that are differentially accessible between these groups. A Jupyter notebook tutorial provides a step-by-step protocol for this with all necessary scripts provided on the GitHub repository.

For simplicity and to provide a conservative comparison with other methods, we ran ROCCO with the same genome-wide parameters for all chromosomes, including the budget which dictates the proportion of the chromosome that should be considered accessible. However, chromatin accessibility varies across chromosomes, and ROCCO's performance may be improved by exploiting properties specific to each chromosome. We found that optimizing the budget parameter for each chromosome for $\mathcal{F}_\beta$-score at $\beta = 1.0$ did show an improvement. These optimized budget parameters roughly reflected the differences in gene density and read density across chromosomes, as expected. In the future, we will focus on developing an efficient, robust method to derive reasonable chromosome-specific budgets based on the input signal data.

ROCCO's locus size parameter was set to $L = 50$ throughout experiments. While our results suggest this grants good performance and a resolution sufficient to identify both broad and concentrated regions of enrichment, it may prove beneficial to modify this parameter depending on the expected size of elements and desired granularity. We note, however, that decreasing L increases the number of loci, n, which may induce additional computational expense. By the same reasoning, computational burden can be reduced by increasing L, though some loss in the precision of predicted peaks may result.

Overall, ROCCO represents an efficient, mathematically sound, flexible, and easy to use method that is broadly applicable and addresses an important need in functional genomics analysis. While we focus on chromatin accessibility data here, ROCCO is broadly applicable to any genomics assay that requires simultaneous prediction of regions of enrichment along with a quantification of those regions across multiple samples, such as ChIP-seq and methyl-seq.

5 Methods

5.1 Computing Environment

Experiments were conducted using a stand-alone computer with an Intel Xeon CPU E5-2680 v3 @ 2.50GHz processor, 8 cores, and 64g RAM. The MOSEK solver (MOSEK, 2022), for which a free academic license can be readily obtained[2], is used to solve (6). We use PEPATAC's `bamSitesToWig` tool to process BAM files and create smooth enrichment signal tracks with constant interval size $L = 50$.

5.2 Data

The ATAC-seq data from 56 lymphoblastoid samples used to conduct experiments was obatined from the ENCODE Project (Luo et al., 2019). Specifically, we used alignments that had been determined according to the ENCODE ATAC-seq protocol. Note, we remove chromosome Y from consideration to ensure each sample contained data for the same set of chromosomes. A link to the metadata for this dataset is available in Supplementary Material 1.6.

5.3 Running ROCCO

Genome-wide results are generated by invoking the `ROCCO.py` script that executes the described Algorithm 2 for each chromosome with a call to `ROCCO_chrom.py` and combined with the `--combine` parameter into a single bed file. If desired, chromosome-specific parameters can be set for `ROCCO_chrom.py` in a CSV file, e.g., `params.csv`. Otherwise, genome-wide defaults (as given in Table 1) are used for all chromosomes.

References

Bao, X., Rubin, A. J., Qu, K., Zhang, J., Giresi, P. G., Chang, H. Y., and Khavari, P. A. (2015). A novel ATAC-seq approach reveals lineage-specific reinforcement of the open chromatin landscape via cooperation between BAF and p63. *Genome Biology*, 16(1).

Bentsen, M., Goymann, P., Schultheis, H., Klee, K., Petrova, A., Wiegandt, R., Fust, A., Preussner, J., Kuenne, C., Braun, T., Kim, J., and Looso, M. (2020). ATAC-seq footprinting unravels kinetics of transcription factor binding during zygotic genome activation. *Nature Communications*, 11(1).

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Boyle, A. P., Davis, S., Shulha, H. P., Meltzer, P., Margulies, E. H., Weng, Z., Furey, T. S., and Crawford, G. E. (2008). High-resolution mapping and characterization of open chromatin across the genome. *Cell*, 132(2):311–322.

Buenrostro, J. D., Wu, B., Chang, H. Y., and Greenleaf, W. J. (2015). Atac-seq: A method for assaying chromatin accessibility genome-wide. *Current Protocols in Molecular Biology*, 109(1):21.29.1–21.29.9.

Corces, M. R. and Granja, J. M. (2018). The chromatin accessibility landscape of primary human cancers. *Science*, 362(6413).

den Hertog, D. (1994). *Interior Point Approach to Linear, Quadratic and Convex Programming*. Springer Netherlands.

Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5.

Domahidi, A., Chu, E., and Boyd, S. (2013). ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076.

Fisher, R. (1925). *Statistical methods for research workers*. Edinburgh Oliver & Boyd.

[2] ROCCO can call any open-source LP solver offered within the CVXPY (Diamond and Boyd, 2016) platform, but runtimes may increase. ECOS (Domahidi et al., 2013) is a viable option installed with CVXPY by default.

Gaspar, J. M. (2018). Improved peak-calling with macs2. *bioRxiv*.

Hofvander, J., Puls, F., Pillay, N., Steele, C. D., Flanagan, A. M., Magnusson, L., Nilsson, J., and Mertens, F. (2019). Undifferentiated pleomorphic sarcomas with prdm10 fusions have a distinct gene expression profile. *The Journal of Pathology*, 249(4):425–434.

Karolchik, D., Baertsch, R., Diekhans, M., Furey, T. S., Hinrichs, A., Lu, Y., Roskin, K. M., Schwartz, M., Sugnet, C. W., Thomas, D. J., et al. (2003). The ucsc genome browser database. *Nucleic acids research*, 31(1):51–54.

Koch, T., Berthold, T., Pedersen, J., and Vanaret, C. (2022). Progress in mathematical programming solvers from 2001 to 2020. *EURO Journal on Computational Optimization*, 10:100031.

Korte, B. and Vygen, J. (2012). *Combinatorial Optimization: Theory and Algorithms*. Springer Publishing Company, Incorporated, 5th edition.

Li, G. and Reinberg, D. (2011). Chromatin higher-order structures and gene regulation. *Current opinion in genetics & development*, 21(2):175–186.

Li, Q., Brown, J. B., Huang, H., and Bickel, P. J. (2011). Measuring reproducibility of high-throughput experiments. *The Annals of Applied Statistics*, 5(3):1752 – 1779.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12).

Luo, Y., Hitz, B. C., Gabdank, I., Hilton, J. A., Kagda, M. S., Lam, B., Myers, Z., Sud, P., Jou, J., Lin, K., Baymuradov, U. K., Graham, K., Litton, C., Miyasato, S. R., Strattan, J. S., Jolanki, O., Lee, J.-W., Tanaka, F. Y., Adenekan, P., O'Neill, E., and Cherry, J. M. (2019). New developments on the encyclopedia of DNA elements (ENCODE) data portal. *Nucleic Acids Research*, 48(D1):D882–D889.

Ming, H., Sun, J., Pasquariello, R., Gatenby, L., Herrick, J. R., Yuan, Y., Pinto, C. R., Bondioli, K. R., Krisher, R. L., and Jiang, Z. (2020). The landscape of accessible chromatin in bovine oocytes and early embryos. *Epigenetics*, 16(3):300–312.

MOSEK (2022). *MOSEK Optimizer API for Python 10.0.39*.

Pham-Gia, T. and Hung, T. (2001). The mean and median absolute deviations. *Mathematical and Computer Modelling*, 34(7):921–936.

Raghavan, P. and Tompson, C. D. (1987). Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374.

Roy, A., Harrar, S. W., and Konietschke, F. (2019). The nonparametric behrens-fisher problem with dependent replicates. *Statistics in Medicine*, 38(25):4939–4962.

Sahinyan, K., Blackburn, D. M., Simon, M.-M., Lazure, F., Kwan, T., Bourque, G., and Soleimani, V. D. (2022). Application of atac-seq for genome-wide analysis of the chromatin state at single myofiber resolution. *eLife*, 11:e72792.

Salavati, M., Woolley, S. A., Araya, Y. C., Halstead, M. M., Stenhouse, C., Johnsson, M., Ashworth, C. J., Archibald, A. L., Donadeu, F. X., Hassan, M. A., and Clark, E. L. (2021). Profiling of open chromatin in developing pig (sus scrofa) muscle to identify regulatory regions. *G3 Genes|Genomes|Genetics*, 12(2).

Smith, J. P., Corces, M. R., Xu, J., Reuter, V. P., Chang, H. Y., and Sheffield, N. C. (2021). PEPATAC: an optimized pipeline for ATAC-seq data analysis with serial alignments. *NAR Genomics and Bioinformatics*, 3(4). lqab101.

Song, L., Zhang, Z., Grasfeder, L. L., Boyle, A. P., Giresi, P. G., Lee, B.-K., Sheffield, N. C., Gräf, S., Huss, M., Keefe, D., Liu, Z., London, D., McDaniell, R. M., Shibata, Y., Showers, K. A., Simon, J. M., Vales, T., Wang, T., Winter, D., Zhang, Z., Clarke, N. D., Birney, E., Iyer, V. R., Crawford, G. E., Lieb, J. D., and Furey, T. S. (2011). Open chromatin defined by DNaseI and FAIRE identifies regulatory elements that shape cell-type identity. *Genome Research*, 21(10):1757–1767.

Tange, O. (2018). *Gnu Parallel 2018*. Zenodo.

Tsaryk, R., Yucel, N., Leonard, E. V., Diaz, N., Bondareva, O., Odenthal-Schnittler, M., Arany, Z., Vaquerizas, J. M., Schnittler, H., and Siekmann, A. F. (2022). Shear stress switches the association of endothelial enhancers from ETV/ETS to KLF transcription factor binding sites. *Scientific Reports*, 12(1).

Wang, J., Zibetti, C., Shang, P., Sripathi, S. R., Zhang, P., Cano, M., Hoang, T., Xia, S., Ji, H., Merbs, S. L., Zack, D. J., Handa, J. T., Sinha, D., Blackshaw, S., and Qian, J. (2018). ATAC-seq analysis reveals a widespread decrease of chromatin accessibility in age-related macular degeneration. *Nature Communications*, 9(1).

Williamson, D. P. and Shmoys, D. B. (2011). *The Design of Approximation Algorithms.* Cambridge University Press.

Yang, Y., Fear, J., Hu, J., Haecker, I., Zhou, L., Renne, R., Bloom, D., and McIntyre, L. M. (2014). Leveraging biological replicates to improve analysis in chip-seq experiments. *Computational and Structural Biotechnology Journal*, 9(13):e201401002.

Zhao, N. and Boyle, A. P. (2021). F-Seq2: improving the feature density based peak caller with dynamic statistics. *NAR Genomics and Bioinformatics*, 3(1). lqab012.

1 Supplementary Material

1.1 Proof of Theorem I

For easy reference, the optimization problem (5) is restated here.

$$\begin{array}{ll} \underset{\boldsymbol{\ell}}{\text{Minimize:}} & f_{\text{CP}}\boldsymbol{\ell}) = \sum_{i=1}^{n} -(\mathcal{S}(i)\cdot \ell_i) + \gamma \sum_{i=1}^{n-1} |\ell_i - \ell_{i+1}| \\ \text{Subject To:} & \text{(i)} \ \sum_{i=1}^{n} \ell_i \leq \lfloor nb \rfloor. \\ & \text{(ii)} \ \ell_i \in [0,1], \ \forall i = 1 \ldots n. \end{array}$$

Theorem 1. *(5) can be solved in polynomial time for a globally optimal solution.*

Proof. The crux of the proof is to show that (5) can be expressed as a linear program (LP): That is, a constrained optimization problem with real-valued decision variables in which the objective function and constraints are linear. These conditions maintain a convex feasible region of solutions with a unique (global) minimum value. Efficiency then follows from the use of interior-point algorithms with well-known polynomial time complexities for LPs (den Hertog, 1994).

A quick remedy for modelling the non-linearity of f is to introduce additional variables, $\ell_j : \forall\, i < n, j = n + i$ to replace the absolute value functions in the objective

$$f_{\text{LP}}(\boldsymbol{\ell}) = \sum_{i=1}^{n} -(\mathcal{S}(i)\cdot \ell_i) + \gamma \sum_{j=n+1}^{2n-1} \ell_j$$

and add constraints such that:

$$\ell_j \geq -(\ell_i - \ell_{i+1}), \ \ell_j \geq (\ell_i - \ell_{i+1}).$$

Since we are *minimizing* f, in any optimal solution, the above constraints imply

$$\ell_j = \max\{\ell_i - \ell_{i+1}, \ell_{i+1} - \ell_i\} = |\ell_i - \ell_{i+1}|$$

for all ℓ_i. Thus, the truncated n-dimensional optimum of the following LP:

$$\begin{array}{ll} \underset{\boldsymbol{\ell}}{\text{Minimize:}} & f_{\text{LP}}(\boldsymbol{\ell}) = \sum_{i=1}^{n} -(\mathcal{S}(i)\cdot \ell_i) + \gamma \sum_{j=n+i}^{2n-1} \ell_j \\ \text{Subject To:} & \text{(i)} \ \sum_{i=1}^{n} \ell_i \leq \lfloor nb \rfloor \\ & \text{(ii)} \ \ell_i \in [0,1], \ \forall i = 1 \ldots n \\ & \text{(iii)} \ \ell_j \geq -1 \cdot (\ell_i - \ell_{i+1}), \ \forall i < n, \ j = n + i \\ & \text{(iv)} \ \ell_j \geq +1 \cdot (\ell_i - \ell_{i+1}), \ \forall i < n, \ j = n + i \end{array} \tag{7}$$

is an optimal solution for (5), which is what we needed to show. □

1.2 Proof of Theorem 2

The following two results are are useful in proving Theorem 2.

Lemma 1. *Let $\boldsymbol{\ell}^{\,rand}$ be a solution generated with the procedure in in Algorithm 1 and let $\boldsymbol{\ell}^{LP}$ be an optimal solution to 6. Then $\boldsymbol{\ell}^{\,rand}$ is integral and possesses the following properties*

1. *The objective value of $\boldsymbol{\ell}^{\,rand}$ is equal in expectation to the objective value of $\boldsymbol{\ell}^{\,LP}$:*

$$\mathbb{E}[f_{\text{LP}}(\boldsymbol{\ell}^{\,rand})] = f_{\text{LP}}(\boldsymbol{\ell}^{\,LP})$$

.

2. *The randomized solution satisfies the budget:*

$$\mathbb{E}\left[\sum_{i=1}^{n} \ell_i^{rand}\right] \leq \lfloor nb \rfloor$$

Both of the statements above follow as consequences of linearity of expectation.

Theorem 2. *Let $\mathbf{L}_N$ be a set of $N > 1$ random solutions generated with Algorithm 1, and let $c > 1$, $a > 0$ be real numbers satisfying $\frac{1}{c} + e^{-2n(ab)^2} < 1$ for n loci and budget $b \in (0, 1)$. Then with high probability, $\mathbf{L}_N$ contains at least one solution with both (a) an objective value no more than c-times $f_{\text{LP}}(\boldsymbol{\ell}^{LP})$ and (b) no more than $nb(1 + a)$ loci selected.*

Proof. To prove the result, we find a lower bound for the probability that these two criteria occur simultaneously for some solution in $\mathbf{L}_N$, and that this probability approaches 1 as $N \to \infty$. Note, for concision, we let $S_n = \sum_{i=1}^{i=n} \ell_i^{rand}$.

Using the expected value for the objective given in Lemma 1, applying Markov's inequality yields:

$$\mathbb{P}\left[\underbrace{f_{\text{LP}}(\boldsymbol{\ell}^{\,rand}) \geq c \cdot f_{\text{LP}}(\boldsymbol{\ell}^{LP})}_{\text{Event A}}\right] \leq \frac{1}{c}.$$

Likewise, recalling that each ℓ_i^{rand} is generated as an independent random variable, we can apply Hoeffding's inequality and Lemma 1 to obtain:

$$\mathbb{P}\left[\underbrace{S_n \geq (1 + a) \cdot nb}_{\text{Event B}}\right] \leq e^{-2n(ab)^2}.$$

Combining these two (possibly dependent) events with the union bound, we get

$$\mathbb{P}\left[A \cup B\right] \leq \mathbb{P}[A] + \mathbb{P}[B] \leq \frac{1}{c} + e^{-2n(ab)^2}.$$

By construction, each solution in $\mathbf{L}_N$ is independent and identically distributed. Since the joint probability of N independent events $\{E_1, E_2, \ldots, E_N\}$, each satisfying $\mathbb{P}[E_i] \leq m_p$, is no more than $(m_p)^N$, we can obtain an upper bound for the probability that at least one of the events, A or B, occurs in *every* solution in $\mathbf{L}_N$:

$$\mathbb{P}\left[\underbrace{\forall\, \boldsymbol{\ell}^{\,rand} \in \mathbf{L}_N :\ A \cup B}_{\text{Event C}}\right] \leq \left(\frac{1}{c} + e^{-2n(ab)^2}\right)^N$$

We are interested in the event that,

$$\exists\, \boldsymbol{\ell}^{\,rand} \in \mathbf{L}_N :\ \neg(A \cup B),$$

or, equivalently,

$$\exists\, \boldsymbol{\ell}^{\,rand} \in \mathbf{L}_N :\ \neg A \ \cap\ \neg B,$$

where at least one solution exists in $\mathbf{L}_N$ such that neither A nor B occur. This is the complement of Event C and occurs with probability at least

$$1 - \left(\frac{1}{c} + e^{-2n(ab)^2}\right)^N.$$

More explicitly, we have:

$$\mathbb{P}\left[\exists\, \boldsymbol{\ell}^{\,rand} \in \mathbf{L}_N :\ f_{\text{LP}}(\boldsymbol{\ell}^{\,rand}) < c \cdot f_{\text{LP}}(\boldsymbol{\ell}^{\,LP}),\ S_n < nb(1 + a)\right]$$
$$\geq 1 - \left(\frac{1}{c} + e^{-2n(ab)^2}\right)^N$$

which rapidly approaches 1 with respect to increasing N. □

1.3 Computational Efficiency

In this experiment, we aim to estimate the average cpu-time to execute Algorithm 2 using the computing environment in Section 5.1 and default parameters.

After 50 trials in which we run the algorithm on the data set described in Section 5.2, we the plot the sample mean runtime for each chromosome and shade the 95%-confidence interval[3] thereof to account for uncertainty arising from stochastic factors at the operating system and processor-level in Figure S1.

[3]The confidence interval is computed according to a t-distribution with $\nu = 49$.

Chromosome size is a clear factor influencing runtime. For example, chr1—the largest of human chromosomes with 250+ million nucleotides—yields the greatest average cpu-time. In contrast, the smallest chromosome, chr21, requires roughly one-tenth the cpu-time of chr1.

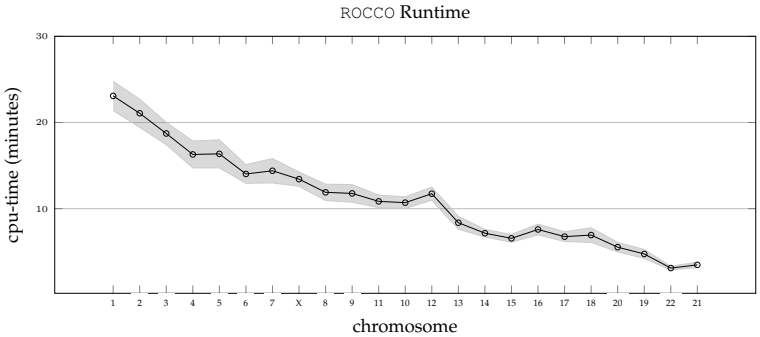

Figure S1: Average cpu-time required to construct and run Algorithm 2 with default parameters using the computing environment in Section 5.1. Chromosomes are ordered by size.

ROCCO's design allows for straightforward parallelization. Given adequate computational resources, a separate process can be created simultaneously for each chromosome by manually calling ROCCO_chrom.py, yielding genome-wide results in a matter of minutes.

ROCCO.py also offers a --jobs parameter that invokes GNU Parallel (parallel) (Tange, 2018) to run chromosome-specific jobs in parallel while controlling memory use with the --mem parameter. This parameter is equivalent to GNU Parallel's --memfree.

The *default* behavior of ROCCO.py, however, is to sequentially evaluate each chromosome. This requires less memory than running in parallel and still grants genome-wide results in roughly 3 hours in our computing environment.

1.4 Default Performance

Detection Performance—No Tuning

β	ROCCO	Genrich	MACS2-Pooled	MACS2-Consensus
0.50	0.597	0.277	0.329	**0.642**
0.75	**0.587**	0.317	0.369	0.565
1.0	**0.578**	0.363	0.413	0.511
1.25	**0.572**	0.410	0.455	0.501
1.50	**0.568**	0.454	0.493	0.452
2.0	0.562	0.530	**0.565**	0.424

Table S1: $\mathcal{F}_\beta$-score obtained from each method using their default significance threshold.

Because parameter tuning can prove difficult in practice due to a lack of general ground truth data, we also compare each method's performance using their *default* significance thresholds: $b = .035,\ q = .05,\ p = .01$ for ROCCO, MACS2, and Genrich, respectively. In Table S1, we see that ROCCO outperforms the benchmarks in four-of-six experiments. The two experiments in which ROCCO does not achieve best performance are for the minimum and maximum β values, where recall is weighed half and twice as much as precision, respectively. Users with a strong preference for either recall or precision are advised to tune accordingly.

1.5 Consistency of Solutions

$\mathcal{J}$	ℓ_1^*	ℓ_2^*	ℓ_3^*	ℓ_4^*	ℓ_5^*
ℓ_1^*	1	0.997758	0.997745	0.997765	0.997743
ℓ_2^*	.	1	0.997734	0.997745	0.997714
ℓ_3^*	.	.	1	0.997721	0.997731
ℓ_4^*	.	.	.	1	0.997738
ℓ_5^*	.	.	.	.	1

Table S2: Pairwise Jaccard similarities $\mathcal{J}$ = `bedtools jaccard -a i.bed -b j.bed` of solutions ℓ^* from five separate genome-wide runs of `ROCCO` with default parameters using the samples discussed in Section 5.2.

1.6 ATAC-seq Data

The 56 ATAC-seq samples used for experiments in this paper are available from the ENCODE Project. These BAM files result from ENCODE's standard ATAC-seq protocol
`https://www.encodeproject.org/pipelines/ENCPL344QWT/`.

Download Metadata
We select the BAM files corresponding to rows with `Output Type='alignments'`.

1.7 Conservative IDR TF Chip-Seq Union Peaks

Available from the F-Seq2 supplementary materials.

We then used the the `liftOver` UCSC binary tool `genome.ucsc.edu/cgi-bin/hgLiftOver` to convert to `hg38`.

1.8 Alternative Methods - Configurations

1.8.1 MACS2-Based Methods

The following configuration of `MACS2` is used to call peaks in each sample for each significance threshold.

```
macs2 callpeak -t {BAM file} -f BAMPE  --nomodel  --nolambda \
--keep-dup all -q {significance threshold}
```

1.8.2 Genrich

Genrich is called with the following configuration for each significance threshold

```
./Genrich -t {comma-separated list of name-sorted BAM files} -j \
-p {significance threshold}
```

The `-j` option specifies ATAC-seq mode