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 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 larger cohorts is particularly 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 employs robust summary statistics and solves a constrained optimization problem formulated to account for both enrichment and spatial dependence 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

Source code, documentation, and usage demos for ROCCO are available on GitHub at: `https://github.com/nolan-h-hamilton/ROCCO`. ROCCO can also be installed as a standalone binary utility using pip/PyPI.

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.

Keywords: ATAC-seq, open chromatin, consensus peaks, convex optimization, peak calling

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, the identification of open chromatin regions 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 annotations are available for many species, there are not comprehensive, predefined standard databases of open chromatin regions.

Studies focused on determining changes in chromatin associated with differing cellular conditions or complex traits normally include many 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 due to variability in data quality across samples, especially when there is an expectation of 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 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 broadly. 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, Guerin et al., 2021], the independence assumption of Fisher's Method may be problematic for large numbers of samples and/or 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 lack of a priori sets of validated open chromatin regions;
- Employs a mathematically tractable model granting useful performance guarantees.

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

[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
$\boldsymbol{\ell}$	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

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.

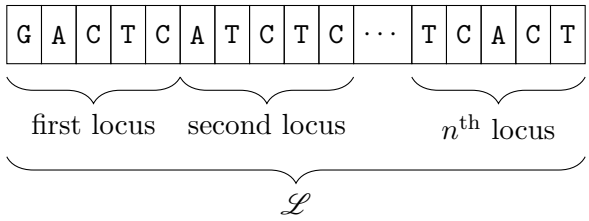

Figure 1: Genomic region $\mathscr{L}$ consisting of n fixed-width loci containing $L = 5$ nucleotides each.

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 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 a context-specific tool, `rocco prep` is included as a subcommand in the software implementation for convenience: Given a directory of samples' BAM files, $\mathbf{S}_{\mathscr{L}}$ is generated with multiple calls to PEPATAC's [Smith et al., 2021] `bamSitesToWig.py` tool.

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:

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

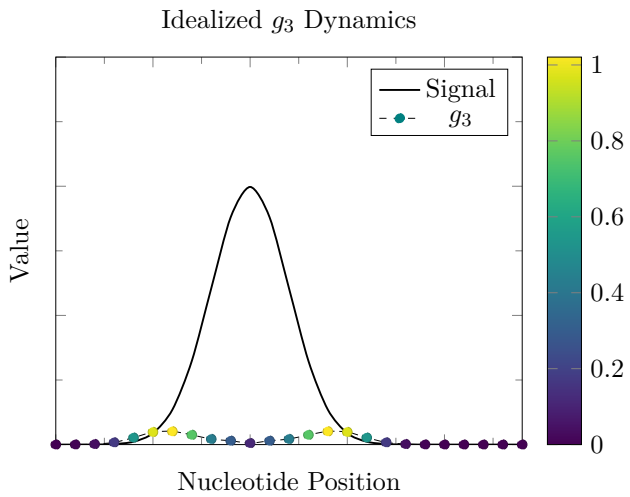

Figure 2: g_3 is designed to more accurately determine peak edges to capture enriched regions in their whole. This function is greatest (yellow marks) near the ends of the enriched region and lowest (dark blue marks) at peak centers in this idealized example.

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 between enrichment signal values at adjacent loci, normalized by the current locus's enrichment,

$$g_3(i) = \frac{1}{g_1(i)+1}\begin{cases} |g_1(i) - g_1(i+1)|, & i = 1 \\ \max\{|g_1(i) - g_1(i+1)|, |g_1(i) - g_1(i-1)|\}, & 1 < i < n \\ |g_1(i) - g_1(i-1)|, & i = n \end{cases}$$

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.

In (2), we define the score piece-wise and take a simple linear combination of g_1, g_2, g_3—or set this score to 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 minimum enrichment 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 selected within a consensus peak, since the objective function (See Eqn. 3) is not completely dependent on the locus score. Note that for default parameters, $\mathcal{S}(i) \geq 0$.

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.

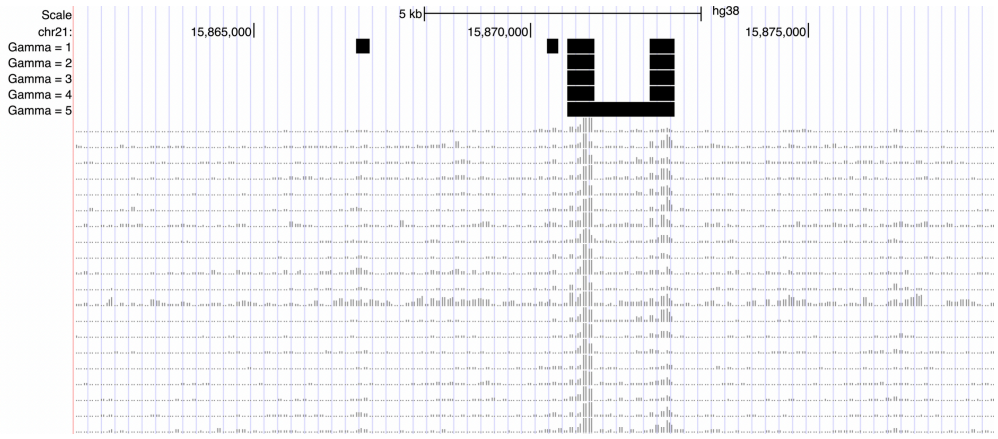

Figure 3: Example behavior of ROCCO in the UCSC Genome Browser [Karolchik et al., 2003] as γ is increased. The black bars in each track correspond to ROCCO's predictions given the replicate signal data below. In the last row, we see that two distinct regions of enrichment are merged due to the strong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

We impose a budget constraint *upper-bounding* the proportion 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 implementation allows for chromosome-specific parameters to be defined.

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

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

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

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

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:

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

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:

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

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

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

The time complexity of standard interior-point methods for solving LPs with n decision variables and a d-bit data representation is $\mathcal{O}\left(n^3 d\right)$, though several methods with improved worst-case bounds have been proposed [Karmarkar, 1984, Vaidya, 1989, den Hertog, 1994]. It is important to note that, in practice, modern solvers offer much greater efficiency than this worst-case bound might suggest by exploiting problem structure [Koch et al., 2022, Boyd and Vandenberghe, 2004]. Indeed, (6) possesses a particularly sparse objective and constraint matrix that allows for surprising efficiency showcased in Supplementary Material 1.3.

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 approximation scheme [Williamson and Shmoys, 2011].

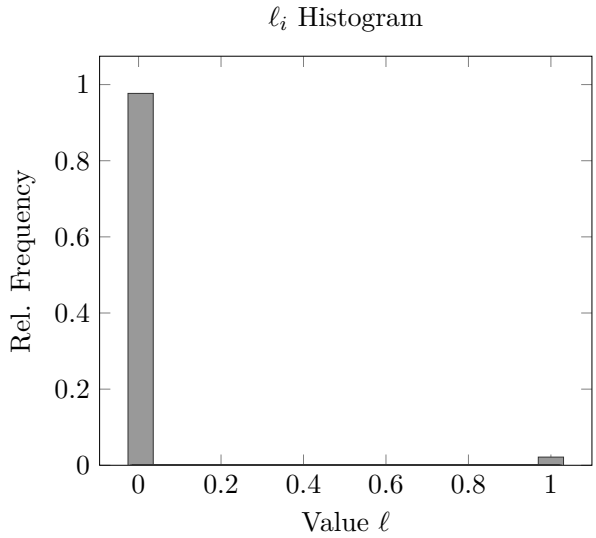

Figure 4: Observed distribution of decision variables after solving (6) with budgets $b \in \{.01, .025, .05, .075, .10\}$ on 50 random subsamples ($K = 40$) of the ATAC-seq data detailed in Section 5.2 and pooling the solutions from each run. See attached material for exact relative frequencies used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

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 practical performance. However, this near-integrality cannot be guaranteed in general, and a more robust 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 feasible solution with the lowest objective value.

Note that in a given solution space $\mathscr{F} \in \mathbb{R}^D, D \geq 1$, integral solutions 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 $\ell_i^{\,rand}$ defined as a Bernoulli-distributed random variable with parameter $p_i = \ell_i^{\,LP}$, grants convenient properties arising from linearity of expectation. Namely,

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

with constraints satisfied in expectation by $\boldsymbol{\ell}^{\,rand}$. We can use these expected values and leverage concentration inequalities to make probabilistic assertions regarding the solutions present in $\mathbf{L}_N$.

Theorem 2. *Let $\mathbf{L}_N$ be a set of $N \geq 1$ 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* $c \cdot \mathsf{OPT}$ *and (b) no more than* $nb(1+a)$ *loci selected.*

In short, Theorem 2 is proven[2] using Markov's and Hoeffding's inequalities to show that the probability of one or more $\boldsymbol{\ell}^{rand} \in \mathbf{L}_N$ satisfying both criteria is at least

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

which quickly approaches 1 for increasing N. We emphasize that this expression is a lower bound on the probability of a satisfying solution, and 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 $c > 1, a > 0$.

The `RR` procedure has linear time and space complexity, making it a minor contributor to overall computational expense. Though this random procedure technically renders `ROCCO` stochastic, 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 S3), 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. The object returned by

Algorithm 2: `ROCCO`

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

1. *Data Fusion and Scoring.* Compute (2)
   $\mathcal{S}(i) \leftarrow$ `score`$(\mathbf{S}_{chr}, \tau, c_1, c_2, c_3)$`[i]`

2. *Optimization.* Solve LP (6)
   $\boldsymbol{\ell}^{LP} \leftarrow$ `solve`$((6), b, \gamma)$;

3. *Solution Refinement.* Obtain integral solution from $\boldsymbol{\ell}^{LP}$
   $\boldsymbol{\ell}^* \leftarrow$ `round_truncate`$(\boldsymbol{\ell}^{LP}$, `method=RR`$)$

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

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 1. *Large Sample Sizes.* Note that only the scoring step is directly affected in runtime by the number of input samples K, running on the order of nK elementary operations using the median of medians algorithm. Practical scenarios satisfy $K << n$, making the number of input samples a minor contributor to computational expense during the calculation of $\mathcal{S}(\cdot)$. Perhaps more importantly, because the scoring step is asymptotically dominated in worst-case computational expense by the optimization step, which is independent of K, the worst-case time complexity of `ROCCO` is likewise independent of the sample size, K.

3 Results

We performed several experiments to assess `ROCCO`'s detection performance using ATAC-seq data from 56 human lymphoblast samples (see Section 5). Additional analyses and details are available in the Supplementary Material.

[2]Supplementary Material 1.2

3.1 Detection Performance

A noteworthy limitation in experiments comparing performance of open chromatin detection methods is a lack of high-confidence 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 Irreproducible Discovery Rate (IDR) [Li et al., 2011] peaks from ENCODE *transcription factor* ChIP-seq experiments in the `GM12878` lymphoblast cell line. We assume that the majority of annotated transcription factor binding sites will correspond to open chromatin regions, but we note that variability in binding at a snapshot in time, the incomplete annotation of all transcription factor binding, and cases where factors can bind to non-accessible chromatin introduce notable limitations. But we argue that this data is sufficient to compare the relative performance of distinct methods. The $\mathcal{F}_\beta$-score, defined below, was then used to assess the ability to recover and bound regions in `GT` using ATAC-seq data from the 56 independent samples. Details regarding the 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

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

and recall as

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

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 to determine consensus peaks begin by identifying sample-specific peaks. For this step, we employed the widely-used MACS2 software [Gaspar, 2018] using parameters commonly specified for ATAC-seq experiments (see Section 5). 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 for Genrich.

3.1.2 Detection Performance: Results

For an initial visual comparison of methods, Figure 5 displays peak calls from each in 100`kb` and 20`kb` 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 signals 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.

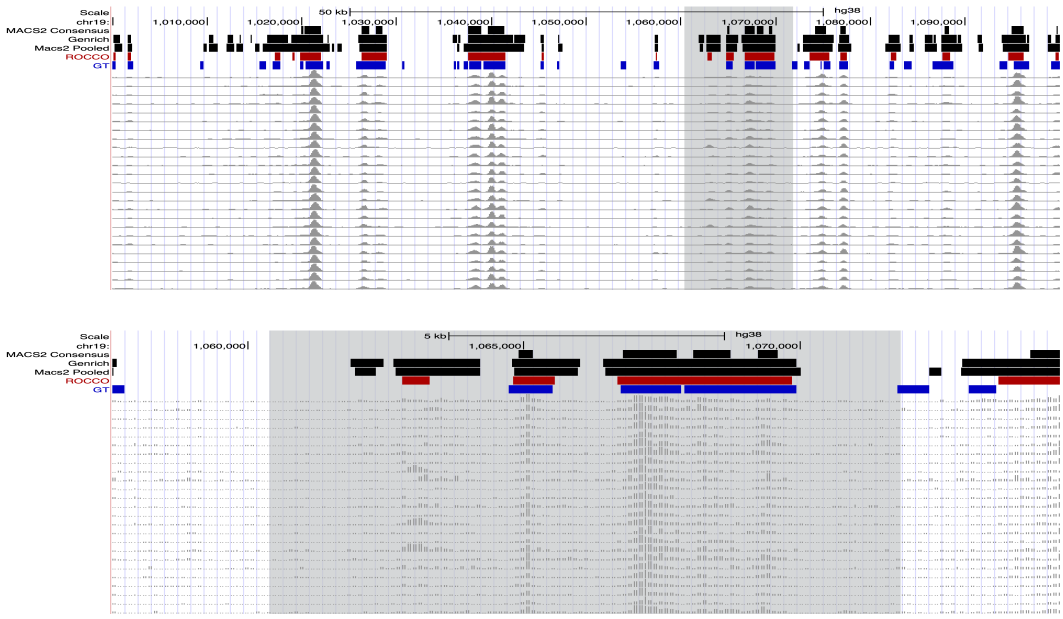

Figure 5: *Example Behavior over* `chr19:`$1000000 - 1100000$. Consensus peak calls from each method tuned using the $\mathcal{F}_\beta$ score ($\beta = 1.0$), displayed at resolutions of 100,000 nucleotides (top) and 20,000 nucleotides (bottom)

To quantify genome-wide detection performance of the methods, 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 their 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 upper-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:

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

Genome-Wide 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.

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 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 across the genome. For a cursory investigation into the effects of this practice, we tuned the budget parameter for $\mathcal{F}_1$ via grid search 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 Table 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 without tuning is included in the Supplementary Material S2, where ROCCO offers the greatest $\mathcal{F}_\beta$-score in all but the $\beta = 0.5, \beta = 2.0$ experiments. Likewise, Supplementary Material 1.3 includes experiments assessing ROCCO's computational efficiency in both theory and practice.

3.2 Variation in Sample Size/Quality

Ideally, a consensus peak calling procedure will

- Effectively leverage data presented by multiple samples.
- Yield robust results in the presence of varying sample quality and size.
- Scale efficiently for large sample sizes.

In this section, we conduct several analyses to consider these aspects for ROCCO.

In the first experiment, ROCCO is repeatedly executed using random subsamples of $K_{\mathsf{sub}} \in \{5, 10, 15, \ldots, 50\}$ ATAC-seq alignments from the data set in Section 5.2 as input. The subsamples' respective output peak sets are then compared to the peak set obtained by running ROCCO on the full set of $K = 56$ ATAC-seq alignments. To compute similarity between the subsamples' peak sets and the entire sample's, we measure the Jaccard index between their respective BED files using `bedtools` [Quinlan and Hall, 2010]. The average Jaccard indexes for each K_{sub} are recorded in Figure 6 along with 95% confidence intervals. Notably, with only $K_{\mathsf{sub}} = 5$ samples, ROCCO generated peak sets roughly 70%-similar to the ROCCO's peak set generated using all $K = 56$ samples. Moreover, ROCCO produced strictly increasing Jaccard indexes for increasing subsample sizes, indicating an effective utilization of additional samples. Though the purpose of this experiment is to evaluate ROCCO's approximation of the full-sample-derived results with respect to varying subsample sizes, we note that detection performance as measured in Section 3.1 likewise improved with respect to increasing K_{sub} from $\mathcal{F}_1 = 0.546 \pm 0.011$ to $\mathcal{F}_1 = 0.5783 \pm .001$.

Regarding efficiency, we note that the cpu-time required to execute ROCCO genome-wide was affected negligibly by K_{sub}, with the average runtime for $K_{\mathsf{sub}} = 5$ and $K_{\mathsf{sub}} = 50$ differing by less than 20% despite the 1000% increase in samples. This result is informed theoretically by Remark 1, where the time complexity of ROCCO is shown to be asymptotically independent of sample size K.

The second experiment compares the effect of data quality on consensus peak sets generated by executing ROCCO independently on the ten best and worst samples as measured by the transcription start site (TSS) enrichment score [Smith et al., 2021]. The data from the 56 lymphoblastoid samples are of relatively good quality (Supplementary Material 1.6), as evidenced by minimum TSS enrichment score of 4.95. Nonetheless, the distribution of scores reflects appreciable differences in sample quality between the left and right tails. With this considered, the relatively small disparities in ROCCO's detection performance shown in Figure 7 indicate robustness to variation in sample quality. In comparison, `MACS2-Consensus`, the best-performing alternative method in this experimental setting, returns lower $\mathcal{F}_1$-scores for both the worst ten samples ($\mathcal{F}_1 = 0.492$) and best ten samples ($\mathcal{F}_1 = 0.530$) and a larger disparity between performance in each case.

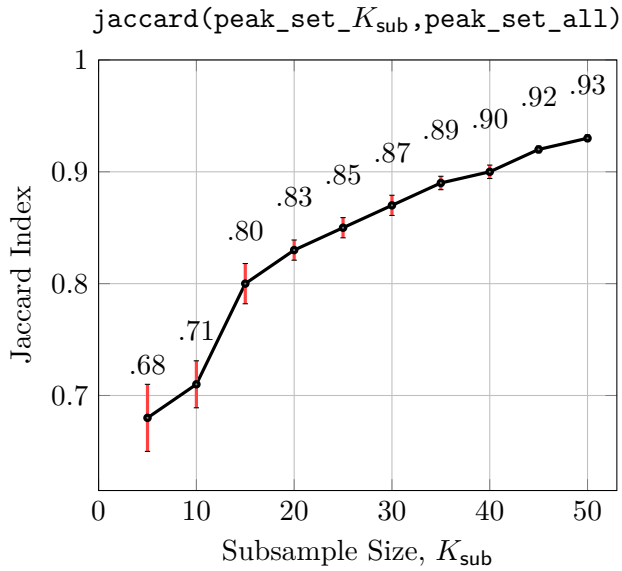

Figure 6: In fifty experiments for each $K_{\text{sub}} \in \{5, 10, 15, \ldots, 50\}$, K_{sub} ATAC-seq alignments are randomly subsampled and supplied as input to ROCCO. The fifty resulting output BED files are used to compute the average Jaccard index (95% C.I.) to ROCCO's results obtained using all $K = 56$ samples.

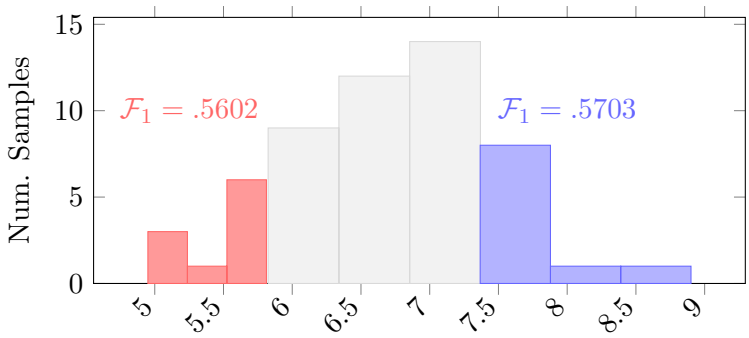

Figure 7: Histogram of TSS enrichment scores as defined by ENCODE for the $K = 56$ ENCODE ATAC-seq samples used in experiments. ROCCO was run twice with default parameters—once using the $K = 10$ *worst* samples (red) and again using the $K = 10$ *best* samples (blue) as measured by TSS score.

3.3 Differential Accessibility Testing with ROCCO

A key motivation for the development of ROCCO was for experimental designs where ATAC-seq data is generated from multiple samples within two or more distinct groups. Using these data, a key question regards the location of genomic regions over which accessibility differs significantly between groups. Knowledge of such regions may yield insights into regulatory mechanisms responsible for phenotypic differences. 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 specific to a particular group are considered, ROCCO was 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 differential detection methods such as DESeq2 [Love et al., 2014]. Note, a Jupyter Notebook tutorial addresses steps for differential analysis with ROCCO and is available on the GitHub repository.

In this demonstration 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 ∼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 of calling group-specific peaks in the context of differential analysis between imbalanced samples.

Despite the seemingly large number of phenotype-specific peaks, our previous experience with these analyses has shown that most of these are due to low signal peaks that are not significantly different across groups. Therefore, DESeq2 was then used to detect significant differentially accessible regions across the merged consensus peaks. At FDR-adjusted $p < 0.05$, 3,141 peaks 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 initially formulating the problem with a 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 *maximum* 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 a scalable, effective, and mathematically sound method that is broadly applicable and addresses an important need in functional genomics analysis.

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[3], is used to solve (6). We run `rocco prep` to process BAM files and create enrichment signal tracks with $L = 50$.

5.2 Data Availability Statement

The ATAC-seq data from 56 lymphoblastoid samples used to conduct experiments was obtained 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.

Competing interests

No competing interest is declared.

References

Xiaomin Bao, Adam J. Rubin, Kun Qu, Jiajing Zhang, Paul G. Giresi, Howard Y. Chang, and Paul A. Khavari. A novel ATAC-seq approach reveals lineage-specific reinforcement of the open chromatin landscape via cooperation between BAF and p63. *Genome Biology*, 16(1), December 2015. doi: 10.1186/s13059-015-0840-9. URL https://doi.org/10.1186/s13059-015-0840-9.

Mette Bentsen, Philipp Goymann, Hendrik Schultheis, Kathrin Klee, Anastasiia Petrova, René Wiegandt, Annika Fust, Jens Preussner, Carsten Kuenne, Thomas Braun, Johnny Kim, and Mario Looso. ATAC-seq footprinting unravels kinetics of transcription factor binding during zygotic genome activation. *Nature Communications*, 11(1), August 2020. doi: 10.1038/s41467-020-18035-1. URL https://doi.org/10.1038/s41467-020-18035-1.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. ISBN 0521833787. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike-20&path=ASIN/0521833787.

Alan P. Boyle, Sean Davis, Hennady P. Shulha, Paul Meltzer, Elliott H. Margulies, Zhiping Weng, Terrence S. Furey, and Gregory E. Crawford. High-resolution mapping and characterization of open chromatin across the genome. *Cell*, 132 (2):311–322, January 2008. doi: 10.1016/j.cell.2007.12.014. URL https://doi.org/10.1016/j.cell.2007.12.014.

Jason D. Buenrostro, Beijing Wu, Howard Y. Chang, and William J. Greenleaf. Atac-seq: A method for assaying chromatin accessibility genome-wide. *Current Protocols in Molecular Biology*, 109(1):21.29.1–21.29.9, 2015. doi: https://doi.org/10.1002/0471142727.mb2129s109. URL https://currentprotocols.onlinelibrary.wiley.com/doi/abs/10.1002/0471142727.mb2129s109.

M. Ryan Corces and Jeffrey M. Granja. The chromatin accessibility landscape of primary human cancers. *Science*, 362 (6413), October 2018. doi: 10.1126/science.aav1898. URL https://doi.org/10.1126/science.aav1898.

D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming*. Springer Netherlands, 1994. doi: 10.1007/978-94-011-1134-8. URL https://doi.org/10.1007/978-94-011-1134-8.

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

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

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

John M. Gaspar. Improved peak-calling with macs2. *bioRxiv*, 2018. doi: 10.1101/496521. URL https://www.biorxiv.org/content/early/2018/12/17/496521.

Lindsey N. Guerin, Kelly R. Barnett, and Emily Hodges. Dual detection of chromatin accessibility and DNA methylation using ATAC-me. *Nature Protocols*, 16(12):5377–5397, October 2021. doi: 10.1038/s41596-021-00608-z. URL https://doi.org/10.1038/s41596-021-00608-z.

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

Jakob Hofvander, Florian Puls, Nischalan Pillay, Christopher D Steele, Adrienne M Flanagan, Linda Magnusson, Jenny Nilsson, and Fredrik Mertens. Undifferentiated pleomorphic sarcomas with prdm10 fusions have a distinct gene expression profile. *The Journal of Pathology*, 249(4):425–434, August 2019. doi: 10.1002/path.5326. URL https://doi.org/10.1002/path.5326.

N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, page 302–311, New York, NY, USA, 1984. Association for Computing Machinery. ISBN 0897911334. doi: 10.1145/800057.808695. URL https://doi.org/10.1145/800057.808695.

Donna Karolchik, Robert Baertsch, Mark Diekhans, Terrence S Furey, Angie Hinrichs, YT Lu, Krishna M Roskin, Matt Schwartz, Charles W Sugnet, Daryl J Thomas, et al. The ucsc genome browser database. *Nucleic acids research*, 31 (1):51–54, 2003.

Thorsten Koch, Timo Berthold, Jaap Pedersen, and Charlie Vanaret. Progress in mathematical programming solvers from 2001 to 2020. *EURO Journal on Computational Optimization*, 10:100031, 2022. doi: 10.1016/j.ejco.2022.100031. URL https://doi.org/10.1016/j.ejco.2022.100031.

Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer Publishing Company, Incorporated, 5th edition, 2012. ISBN 3642244874.

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

Qunhua Li, James B. Brown, Haiyan Huang, and Peter J. Bickel. Measuring reproducibility of high-throughput experiments. *The Annals of Applied Statistics*, 5(3):1752 – 1779, 2011. doi: 10.1214/11-AOAS466. URL https://doi.org/10.1214/11-AOAS466.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12), December 2014. doi: 10.1186/s13059-014-0550-8. URL https://doi.org/10.1186/s13059-014-0550-8.

Yunhai Luo, Benjamin C Hitz, Idan Gabdank, Jason A Hilton, Meenakshi S Kagda, Bonita Lam, Zachary Myers, Paul Sud, Jennifer Jou, Khine Lin, Ulugbek K Baymuradov, Keenan Graham, Casey Litton, Stuart R Miyasato, J Seth Strattan, Otto Jolanki, Jin-Wook Lee, Forrest Y Tanaka, Philip Adenekan, Emma O'Neill, and J Michael Cherry. New developments on the encyclopedia of DNA elements (ENCODE) data portal. *Nucleic Acids Research*, 48(D1): D882–D889, November 2019. doi: 10.1093/nar/gkz1062. URL https://doi.org/10.1093/nar/gkz1062.

Hao Ming, Jiangwen Sun, Rolando Pasquariello, Lauren Gatenby, Jason R Herrick, Ye Yuan, Carlos R Pinto, Kenneth R Bondioli, Rebecca L Krisher, and Zongliang Jiang. The landscape of accessible chromatin in bovine oocytes and early embryos. *Epigenetics*, 16(3):300–312, July 2020. doi: 10.1080/15592294.2020.1795602. URL https://doi.org/10.1080/15592294.2020.1795602.

MOSEK. *MOSEK Optimizer API for Python 10.0.39*, 2022. URL https://docs.mosek.com/latest/pythonapi/index.html.

T. Pham-Gia and T.L. Hung. The mean and median absolute deviations. *Mathematical and Computer Modelling*, 34(7):921–936, 2001. ISSN 0895-7177. doi: https://doi.org/10.1016/S0895-7177(01)00109-1. URL https://www.sciencedirect.com/science/article/pii/S0895717701001091.

Aaron R. Quinlan and Ira M. Hall. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, January 2010. doi: 10.1093/bioinformatics/btq033. URL https://doi.org/10.1093/bioinformatics/btq033.

Prabhakar Raghavan and Clark D. Tompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, December 1987. doi: 10.1007/bf02579324. URL https://doi.org/10.1007/bf02579324.

Akash Roy, Solomon W. Harrar, and Frank Konietschke. The nonparametric behrens-fisher problem with dependent replicates. *Statistics in Medicine*, 38(25):4939–4962, 2019. doi: https://doi.org/10.1002/sim.8343. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.8343.

Korin Sahinyan, Darren M Blackburn, Marie-Michelle Simon, Felicia Lazure, Tony Kwan, Guillaume Bourque, and Vahab D Soleimani. Application of atac-seq for genome-wide analysis of the chromatin state at single myofiber resolution. *eLife*, 11:e72792, feb 2022. ISSN 2050-084X. doi: 10.7554/eLife.72792. URL https://doi.org/10.7554/eLife.72792.

Mazdak Salavati, Shernae A Woolley, Yennifer Cortés Araya, Michelle M Halstead, Claire Stenhouse, Martin Johnsson, Cheryl J Ashworth, Alan L Archibald, Francesc X Donadeu, Musa A Hassan, and Emily L Clark. Profiling of open chromatin in developing pig (sus scrofa) muscle to identify regulatory regions. *G3 Genes|Genomes|Genetics*, 12(2), December 2021. doi: 10.1093/g3journal/jkab424. URL https://doi.org/10.1093/g3journal/jkab424.

Jason P Smith, M Ryan Corces, Jin Xu, Vincent P Reuter, Howard Y Chang, and Nathan C Sheffield. PEPATAC: an optimized pipeline for ATAC-seq data analysis with serial alignments. *NAR Genomics and Bioinformatics*, 3(4), 11 2021. ISSN 2631-9268. doi: 10.1093/nargab/lqab101. URL https://doi.org/10.1093/nargab/lqab101. lqab101.

Lingyun Song, Zhancheng Zhang, Linda L. Grasfeder, Alan P. Boyle, Paul G. Giresi, Bum-Kyu Lee, Nathan C. Sheffield, Stefan Gräf, Mikael Huss, Damian Keefe, Zheng Liu, Darin London, Ryan M. McDaniell, Yoichiro Shibata, Kimberly A. Showers, Jeremy M. Simon, Teresa Vales, Tianyuan Wang, Deborah Winter, Zhuzhu Zhang, Neil D. Clarke, Ewan Birney, Vishwanath R. Iyer, Gregory E. Crawford, Jason D. Lieb, and Terrence S. Furey. Open chromatin defined by DNaseI and FAIRE identifies regulatory elements that shape cell-type identity. *Genome Research*, 21(10): 1757–1767, July 2011. doi: 10.1101/gr.121541.111. URL https://doi.org/10.1101/gr.121541.111.

Roman Tsaryk, Nora Yucel, Elvin V. Leonard, Noelia Diaz, Olga Bondareva, Maria Odenthal-Schnittler, Zoltan Arany, Juan M. Vaquerizas, Hans Schnittler, and Arndt F. Siekmann. Shear stress switches the association of endothelial enhancers from ETV/ETS to KLF transcription factor binding sites. *Scientific Reports*, 12(1), March 2022. doi: 10.1038/s41598-022-08645-8. URL https://doi.org/10.1038/s41598-022-08645-8.

P.M. Vaidya. Speeding-up linear programming using fast matrix multiplication. In *30th Annual Symposium on Foundations of Computer Science*. IEEE, 1989. doi: 10.1109/sfcs.1989.63499. URL https://doi.org/10.1109/sfcs.1989.63499.

Jie Wang, Cristina Zibetti, Peng Shang, Srinivasa R. Sripathi, Pingwu Zhang, Marisol Cano, Thanh Hoang, Shuli Xia, Hongkai Ji, Shannath L. Merbs, Donald J. Zack, James T. Handa, Debasish Sinha, Seth Blackshaw, and Jiang Qian. ATAC-seq analysis reveals a widespread decrease of chromatin accessibility in age-related macular degeneration. *Nature Communications*, 9(1), April 2018. doi: 10.1038/s41467-018-03856-y. URL https://doi.org/10.1038/s41467-018-03856-y.

David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, 2011. ISBN 978-0-521-19527-0.

Yajie Yang, Justin Fear, Jianhong Hu, Irina Haecker, Lei Zhou, Rolf Renne, David Bloom, and Lauren M McIntyre. Leveraging biological replicates to improve analysis in chip-seq experiments. *Computational and Structural Biotechnology Journal*, 9(13):e201401002, 2014. ISSN 2001-0370. doi: https://doi.org/10.5936/csbj.201401002. URL https://www.sciencedirect.com/science/article/pii/S2001037014600106.

Nanxiang Zhao and Alan P Boyle. F-Seq2: improving the feature density based peak caller with dynamic statistics. *NAR Genomics and Bioinformatics*, 3(1), 02 2021. ISSN 2631-9268. doi: 10.1093/nargab/lqab012. URL https://doi.org/10.1093/nargab/lqab012. 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 solution to the following LP:

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

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 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 in expectation:*

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

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

Note, without loss of generality, we can treat the objective function f as non-negative in order to satisfy the condition for Markov's inequality, used in proof of Theorem 2. Because f is bounded below, we can always add a constant to ensure $\min f \geq 0$ without affecting optimality of solutions since

$$\arg\min_{\boldsymbol{\ell}} \; f(\boldsymbol{\ell}) = \arg\min_{\boldsymbol{\ell}} \; f(\boldsymbol{\ell}) + C.$$

Theorem 2. *Let $\mathbf{L}_N$ be a set of $N \geq 1$ 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 $c \cdot \mathsf{OPT}$ 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 $S_n = \sum_{i=1}^{i=n} \ell_i^{\,rand}$.
Now, 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 $\ell_i^{\,rand}$ 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 approaches 1 for increasing N. □

1.3 Computational Efficiency

In this section, we discuss `ROCCO`'s computational efficiency from both theoretical and practical perspectives. Unless otherwise stated, `ROCCO`'s default parameters and the computing environment discussed in Section 5.1 are used.

The time complexity of `ROCCO` is equivalent to the time complexity required to solve a linear program (LP) with n variables and a d-bit data representation, since this step dominates the others asymptotically (See Algorithm 2). However, the $\mathcal{O}(n^3 d)$ bound of standard interior-point solvers is of little insight for the vast majority of practical problems using current methodology and is only realized in pathological cases with completely dense matrices. A detailed survey of the improved capabilities of modern solvers and the multitude of algorithmic and hardware-based advances experienced in the past two decades is out of scope for this paper, but readers are referred to [Koch et al., 2022].

To gain a practical sense of `ROCCO`'s computational expense, an empirical analysis is warranted. To this end, we ran `ROCCO` genome-wide in 50 independent trials on the data set described in Section 5.2. The observed mean runtime and maximum resident set size (MRSS) for each chromosome were then plotted, with the 95%-confidence intervals[4] thereof shaded to account for uncertainty arising from stochastic factors at the operating system and processor-level in Figure S1.

Table S1: Chromosome Sizes (bp) in `hg38`

Chromosome	Base pairs (Millions)
1	248.96
2	242.19
3	198.30
4	190.21
5	181.54
6	170.81
7	159.35
8	145.14
9	138.39
10	133.80
11	135.09
12	133.28
13	114.36
14	107.04
15	101.99
16	90.34
17	81.28
18	78.08
19	59.13
20	63.03
21	48.13
22	50.96
X	155.27
Y	59.37

Chromosome size proved to be a fundamental contributor to computational expense. For example, `chr1`—the largest of human chromosomes—yielded the greatest average runtime and MRSS. In contrast,

[4]The confidence intervals were computed according to a t-distribution with $\nu = 49$.

the smallest chromosome, `chr21`, required roughly one-tenth the time of `chr1`. See Table S1 for the number of base pairs in each human chromosome, scaled to millions.

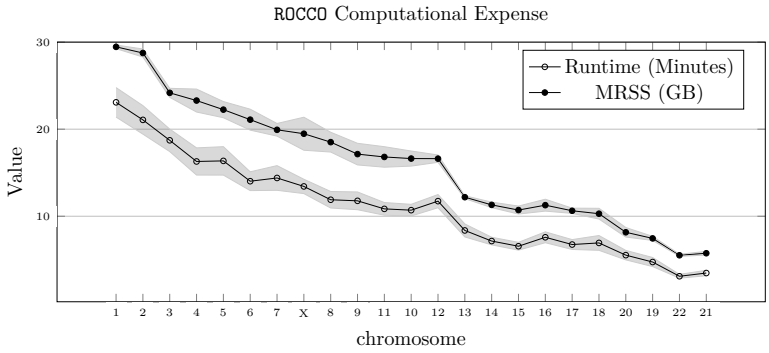

Figure S1: Average runtime and peak memory (MRSS) expended by `ROCCO` for each chromosome.

We note that `ROCCO`'s design allows for straightforward parallelization, since each chromosome can be processed independently. An option for simultaneous execution of `ROCCO` on multiple chromosomes is available in the software implementation via `rocco gwide --multi`. This feature can dramatically reduce the runtime of `ROCCO` if multiple cores are available. However, to accommodate users in limited computing environments, the default behavior of `rocco gwide` is to run the method sequentially on each chromosome, leading to an average genome-wide runtime of 4.3 hours in this experiment[5].

1.4 `ROCCO` Parameters

`ROCCO` is designed to require minimal fine-tuning. Many of the parameters are included only for completeness and can be ignored for common use-cases. The default parameters provide generally strong performance as shown in Table S2, but we highlight several potentially consequential dynamics for users wishing to configure `ROCCO` optimally for a particular setting. In particular, the budget parameter may non-trivially affect behavior of the method.

1.4.1 Constant Budgets

The budget, b, is a fundamental parameter of `ROCCO` as it determines an *upper-bound* on the proportion of loci that can be selected. A genome-wide, constant budget can yield strong detection performance as demonstrated in Tables 2, S2. The best constant budget depends on users' preferences regarding recall/precision, as seen in Figure S2. Intuitively, for increasing β, that is, an increased emphasis on recall than precision, the best-scoring budget b is increasing.

[5]In a personal computing environment, `rocco gwide` successfully returned genome-wide results on a MacBook (Model A2485) in under five hours.

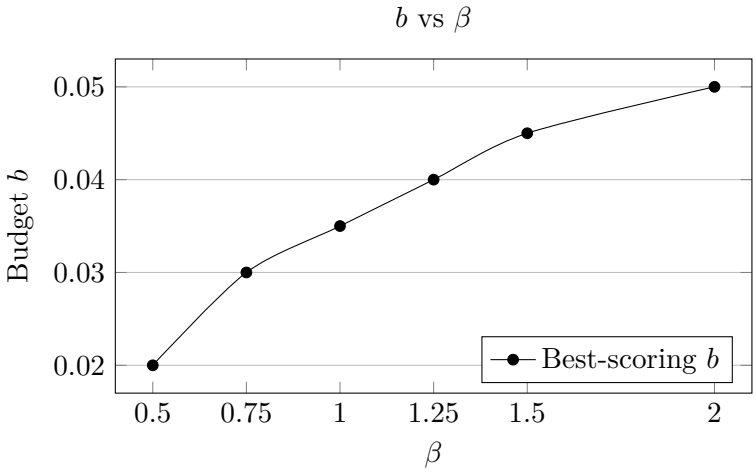

Figure S2: As greater importance is placed on recall than precision (β is increased), the $\mathcal{F}_\beta$-tuned budget increases.

1.4.2 Chromosome-Specific Budgets

A notion of dynamic, region-specific budgets is naturally appealing given chromatin's variable dynamics throughout the genome. `ROCCO`'s software implementation allows users to specify budgets (and other parameters as well) at the chromosome level using `rocco gwide`'s `-p/--param_file` parameter. See `hg38_params.csv` for suggested budgets for each chromosome in the `hg38` assembly. Using these chromosome-specific budgets to run `ROCCO` genome-wide on the ATAC-seq data in Section 5.2, $\mathcal{F}_1 = .592$ was obtained as compared to $\mathcal{F}_1 = .579$ with an invariant, genome-wide budget (Table 2).

In the software implementation, `rocco budgets` can also be used to compute chromosome-specific budgets. In this procedure, estimates for the relative read densities of each chromosome are scaled such that they average to a user-specified value, e.g., $b = .035$. The ranking of chromosomes' budgets according to read density is preserved under this operation, and the scaling step produces interpretable budget values as we have defined them. See the API reference and corresponding script for more details.

1.4.3 Locus Size, L and γ

At higher resolutions, it is natural to expect less variance between proximal elements, as the absolute distance between them is lesser by definition. Consequently, for smaller locus sizes L, users may wish to enforce a stronger dependence between adjacent loci by increasing γ. As a brief demonstration in Figure S3, all parameters except γ, L are fixed at their defaults, and we observe greater fragmentation in `ROCCO`'s solution with $(\gamma = 1, L = 25)$ than the $(\gamma = 2, L = 25)$, $(\gamma = 1, L = 50)$, and $(\gamma = 2, L = 50)$ cases.

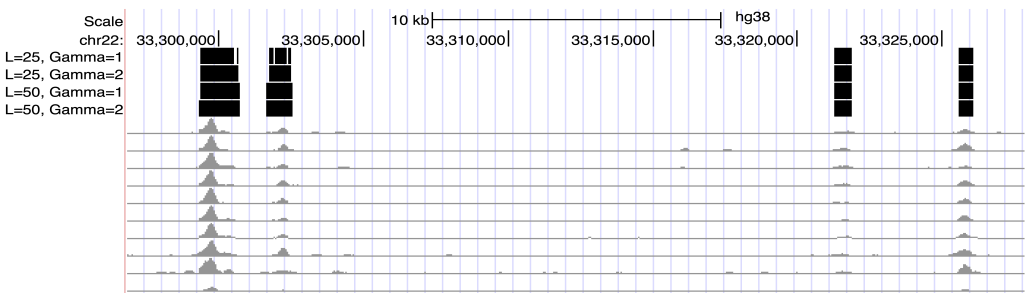

Figure S3: *Relationship between γ and L.* `ROCCO`'s annotations on a demonstrative 10kb subset of chromosome 22 are displayed. `ROCCO` is run four times on a set of ten ATAC-seq alignments with all parameters default/fixed except γ, L.

1.4.4 Default Performance

Genome-Wide 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	**0.579**	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 S2: $\mathcal{F}_\beta$-score obtained from each method using their default significance threshold.

To measure "out-of-box" performance of ROCCO, we also compare each method using their *default* significance thresholds: $b = .035$, $q = .05$, $p = .01$ for ROCCO, MACS2, and Genrich, respectively. In Table S2, 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 Randomness of RR Procedure

Using the data set described in Section 5.2, ROCCO is run five times independently using the default $N = 50$ RR iterations. The pairwise Jaccard statistics between resulting peak sets are then computed in Table S3.

$\mathcal{J}$	$\boldsymbol{\ell}_1^*$	$\boldsymbol{\ell}_2^*$	$\boldsymbol{\ell}_3^*$	$\boldsymbol{\ell}_4^*$	$\boldsymbol{\ell}_5^*$
$\boldsymbol{\ell}_1^*$	1	0.997758	0.997745	0.997765	0.997743
$\boldsymbol{\ell}_2^*$	.	1	0.997734	0.997745	0.997714
$\boldsymbol{\ell}_3^*$	.	.	1	0.997721	0.997731
$\boldsymbol{\ell}_4^*$	.	.	.	1	0.997738
$\boldsymbol{\ell}_5^*$	.	.	.	.	1

Table S3: Pairwise Jaccard indexes of peak sets obtained during five separate genome-wide runs of ROCCO with default parameters and $N = 50$ RR iterations.

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 for ATAC-seq samples:
https://www.encodeproject.org/metadata/?control_type%21=%2A&status=released&perturbed=false&assay_title=ATAC-seq&biosample_ontology.cell_slims=lymphoblast&files.file_type=bam&type=Experiment

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

1.7 Conservative IDR TF Chip-Seq Union Peaks

Available from:
https://github.com/Boyle-Lab/F-Seq2-Paper-Supplementary/blob/d9c5357e615535240b09fac6567874bf6
ATAC-seq%20benchmarking/data_process/download_preprocess_data.ipynb

We then used the textttliftOver 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

1.9 Peak Width Distributions

Each method is tuned for the $\mathcal{F}_1$-score as in Table 2 Row 3, and their respective peak-width distributions plotted in Figure S4. For each method, there is a consistent majority of peaks less than 500 bps with strictly decreasing frequency past this length.

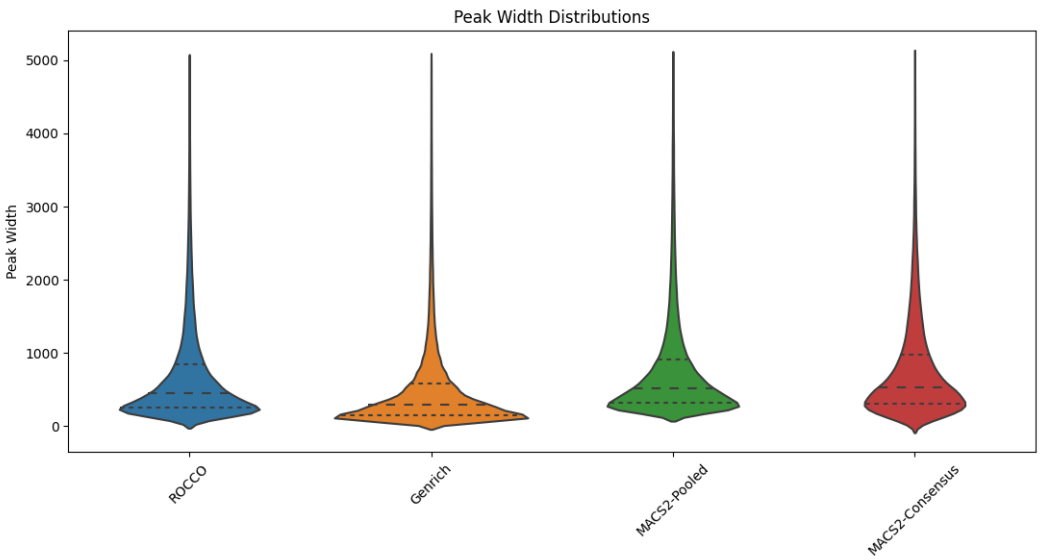

Figure S4: Violin plots of the peak width distributions for each method.