Unsupervised contrastive peak caller for ATAC-seq

Ha T.H. Vu, Yudi Zhang, Geetu Tuteja, et al.

Genome Res. 2023 33: 1133-1144 originally published online May 22, 2023
Access the most recent version at doi:10.1101/gr.277677.123

Method

Unsupervised contrastive peak caller for ATAC-seq

Ha T.H. Vu,[1,2,4] Yudi Zhang,[3,4] Geetu Tuteja,[1,2] and Karin S. Dorman[1,2,3]

[1]*Bioinformatics and Computational Biology Program, Iowa State University, Ames, Iowa 50011, USA;* [2]*Department of Genetics, Development and Cell Biology,* [3]*Department of Statistics, Iowa State University, Ames, Iowa 50011, USA*

The assay for transposase-accessible chromatin with sequencing (ATAC-seq) is a common assay to identify chromatin accessible regions by using a Tn5 transposase that can access, cut, and ligate adapters to DNA fragments for subsequent amplification and sequencing. These sequenced regions are quantified and tested for enrichment in a process referred to as "peak calling." Most unsupervised peak calling methods are based on simple statistical models and suffer from elevated false positive rates. Newly developed supervised deep learning methods can be successful, but they rely on high quality labeled data for training, which can be difficult to obtain. Moreover, though biological replicates are recognized to be important, there are no established approaches for using replicates in the deep learning tools, and the approaches available for traditional methods either cannot be applied to ATAC-seq, where control samples may be unavailable, or are post hoc and do not capitalize on potentially complex, but reproducible signal in the read enrichment data. Here, we propose a novel peak caller that uses unsupervised contrastive learning to extract shared signals from multiple replicates. Raw coverage data are encoded to obtain low-dimensional embeddings and optimized to minimize a contrastive loss over biological replicates. These embeddings are passed to another contrastive loss for learning and predicting peaks and decoded to denoised data under an autoencoder loss. We compared our replicative contrastive learner (RCL) method with other existing methods on ATAC-seq data, using annotations from ChromHMM genomic labels and transcription factor ChIP-seq as noisy truth. RCL consistently achieved the best performance.

[Supplemental material is available for this article.]

The assay for transposase-accessible chromatin with sequencing (ATAC-seq) is widely used when studying chromatin biology (Grandi et al. 2022). ATAC-seq uses a hyperactive mutant Tn5 transposase to cleave double stranded DNA and to attach adapters for subsequent sequencing by high throughput technologies (Buenrostro et al. 2015). Since DNA is more easily cleaved where it is unwound and open, sequenced DNA fragments tend to arise from regions of open chromatin. A standard analysis for ATAC-seq starts with aligning the sequencing reads to a reference genome using BWA (Li 2013), Bowtie 2 (Langmead and Salzberg 2012), or other short read aligner (Musich et al. 2021). Then peak calling methods will identify the open regions (peaks) in the genome where aligned reads are enriched. Downstream analyses include motif detection, differential binding analysis, or footprint identification (Buenrostro et al. 2013; Grandi et al. 2022), all of which require accurate peak calls. Unfortunately, peaks of false enrichment may be called due to mapping errors or experimental noise (Park 2009). Such errors can be reduced by masking repetitive regions and using control samples (Zhang et al. 2008), but input controls for ATAC-seq are typically not used due to high sequencing costs (Yan et al. 2020).

ATAC-seq peaks are often called with the most popular general-purpose peak caller, MACS (Zhang et al. 2008), and there is an ATAC-seq-specific method called HMMRATAC (Tarbell and Liu 2019). MACS slides a fixed-width window across the genome to find candidate peaks. The number of reads aligned to the genome in the current window is modeled as a Poisson random variable, with a dynamic mean to capture local variation in background coverage rates. MACS calculates the *P*-value for each candidate peak as the probability of obtaining coverage at or above the observed coverage given the current background rate. HMMRATAC (Tarbell and Liu 2019) employs a hidden Markov model (HMM) with four-dimensional (4D) emissions of varying fragment sizes, nucleosome-free (NF), one nucleosome (1N), two nucleosome (2N), and three nucleosome (3N) fragments, from three possible hidden states: a "center" state (open chromatin), with high emissions in all four dimensions; a nucleosome state, with low NF fragment emission; and a background state, with low emissions in all dimensions. Once the HMM has been estimated, the Viterbi algorithm is used to classify every 10 bp window in the genome into one of the three states.

Traditional modeling methods tend to predict many false positive peaks in ChIP-seq applications (Hocking et al. 2017), and some investigations have shown humans to be superior "peak callers" (Rye et al. 2011; Hocking et al. 2017). Inspired by such human performance and recent successes in artificial intelligence, two new peak callers, CNN-Peaks (Oh et al. 2020) and LanceOtron (Hentges et al. 2021), take a deep learning approach. CNN-Peaks (Oh et al. 2020) uses supervised convolutional neural networks (CNN) to call ChIP-seq peaks. In addition to the read count information obtained from BAM files, it uses genome annotation information, such as protein-coding transcripts, to improve estimation of peak locations. In their CNN architecture, filters of various sizes are used to extract diverse features and a weighted cross-entropy loss is adopted to account for the imbalanced labels. LanceOtron (Hentges et al. 2021) is another supervised CNN-based deep learning method that can be used on ATAC-seq, ChIP-seq, and DNase-seq data. It feeds the output of a logistic regression, fit to 11 enrichment scores predicting labeled peaks, the output of a CNN, fit to fragment coverage in 2000 bp windows predicting

[4]**These authors contributed equally to this work.**
Corresponding author: kdorman@iastate.edu
Article published online before print. Article, supplemental material, and publication date are at https://www.genome.org/cgi/doi/10.1101/gr.277677.123. Freely available online through the *Genome Research* Open Access option.

labeled peaks, and the 11 enrichment scores to a multilayer perceptron to produce the overall peak score. Many of the false positive peaks generated by other peak callers are filtered out by these supervised deep learners, increasing precision by about 18% (Hentges et al. 2021). Unfortunately, these supervised methods require labeled data for model training, which are often hard or costly to obtain.

None of these methods consider biological replicates, and in fact most peak calling methods assess biological replicates separately (Goren et al. 2018). HMMRATAC and some users of MACS recommend combining multiple replicates to increase signal, but joint analysis of multiple biological replicates could improve the power to distinguish actual transcription factor binding events (Newell et al. 2021), since some weak or highly variable peak signals may only become evident across multiple replicates (Zhang et al. 2014). One common approach for assessing reproducibility from replicates uses the irreproducible discovery rate (IDR), which identifies reproducible peaks by measuring the consistency in peak ranks between replicates (Li et al. 2011). ChIP-R (Newell et al. 2021), which shows improvement over IDR and can handle more than two replicates, uses the rank product to evaluate the reproducibility across any number of ChIP-seq or ATAC-seq replicates.

We introduce a novel unsupervised learning method that uses contrastive learning (Le-Khac et al. 2020) across replicates to separate genomic regions into peaks and nonpeaks. The proposed peak calling framework combines signals from multiple replicates to identify chromatin accessible regions with ATAC-seq data, and overcomes excess noise and lack of labels to make better inferences than existing methods.

Results

The RCL algorithm

In this study, we developed a peak calling tool (RCL), which contrasts biological replicates to identify the shared signals of ATAC-seq peaks (Fig. 1). Peak calling is a difficult task, where the genomic extent and significance of enrichment, together the *peak*, must be inferred. Our proposed method separates these tasks, first liberally identifying candidate regions of possible enrichment, and then learning how to score and classify data extracted from the regions. The learner makes no attempt to learn peak boundaries, so its predictions are passed back to the original candidate regions, which become peak predictions if sufficiently high scoring.

Prediction region selection

In general, the RCL framework is applicable for replicated experiments. The individual BAM files of $R \geq 2$ replicates and a merged BAM file are required to identify candidate peak regions. Additionally, two user settable parameters, coverage threshold (t, default: "median," see Step 1 below) and input segment length (α, default: 1000), affect the number and length of the candidate regions. Given these inputs, candidate peak regions are identified as follows:

Step 1: Retain genome positions with coverage $> t$ in all R individual BAM files. Threshold t defaults to a chromosome-specific value obtained from the input data. Specifically, the read coverage on each chromosome is calculated using BEDTools genomecov (Quinlan and Hall 2010) for every replicate (bedtools genomecov -ibam bamFiles -pc -bga), then median coverage across all nonzero positions per chromosome is obtained. The minimum median observed across replicates for a chromosome is used as the threshold for that chromosome. Alternatively, t can be set as a single integer value to be used for every chromosome.

Step 2: Contiguous retained sites are aggregated into regions. Then, regions within 90 bp are merged, since DNA linkers are known to be 8–90 bp (Singh and Mueller-Planitz 2021). All regions longer than 100 bp are retained for Step 3. Define this set of regions as $\mathcal{A}$.

Step 3.1: If a region in set $\mathcal{A}$ is shorter than α, an α bp long genomic segment is obtained by extending $\frac{\alpha}{2}$ bp upstream of and downstream from its midpoint.

Step 3.2: For regions in set $\mathcal{A}$ longer than α bp, we first get positions with coverage summed across replicates ≥0.95 quantile of the region (obtained from the merged BAM file). Positions within α bp are merged, then we extended $\frac{\alpha}{2}$ bp upstream of and downstream from each merged region's midpoint.

Hereafter, "segment" refers to these selected α bp genomic fragments. Any segment overlapping with a blacklist region

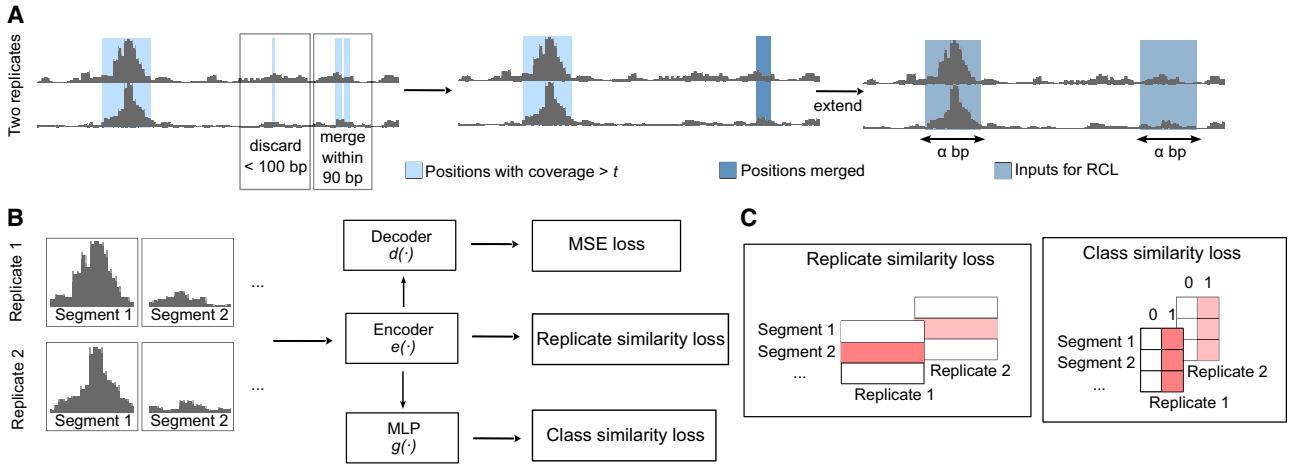

Figure 1. RCL model. (*A*) The raw input is processed to extract α-length input segments representing the same genomic region in all replicates. (*B*) The α-length input segments are fed to encoder $e(\cdot)$ to compute the cross-replicate contrastive loss. Then the embedding is fed to a multilayer perceptron (MLP), specifically a fully connected neural network, for class similarity loss and a decoder for the autoencoder (MSE) loss. The encoder/decoder has five ResNET blocks. (*C*) Shaded red boxes represent the elements contrasted in the respective losses.

(Amemiya et al. 2019) by at least 1 bp is removed. In the end, per-base coverage vectors for these length α bp segments from R replicates are the inputs to RCL. RCL assigns scores to the α bp segments as described in the next section. These scores are combined into a single score for each candidate region in $\mathcal{A}$ (see "RCL peak calling" in Methods section "Method comparison"). Since RCL does not estimate peak boundaries, we show in Supplemental Text S3.4 that transferring RCL scores to candidate regions in $\mathcal{A}$ better locates peaks than using the α bp segments directly.

Unsupervised learner

We use a neural network to assign a score to each segment. As illustrated in Figure 1, our method consists of three jointly learned components along with their respective losses, so the total minimized loss is

$$L = l_1 + l_2 + l_3,$$

shown without optional weights that can be tuned by standard cross-validation methods. The three components are a cross-replicate contrastive learner (Chen et al. 2020), a segment class (peak/nonpeak) learner (Zhong et al. 2020), and an autoencoder (Kramer 1991). The input to the contrastive learner and segment class learner is the output of the encoder network that maps the α bp coverage data to a lower-dimensional representation space.

Encoder. With R replicates of observed coverage in S segments, the input data are per-base coverage vectors $\boldsymbol{m}_{ri}$, $r \in \{1, \ldots, R\}$, $i \in \{1, \ldots, S\}$. We use ResNET (He et al. 2016) as the backbone of our encoder network. A ResNET module is composed of three basic blocks followed by one residual block. A basic block is composed of a 1D convolutional layer (default: dilation 8 and kernel size 31, more details in Supplemental Text S3.1.1; Supplemental Fig. S8), followed by a RELU activation function. Our whole encoder $e(\cdot)$ is made of five such ResNET modules, producing the lower-dimensional (default dimension: 50) representation $\boldsymbol{x}_{ri} = e(\boldsymbol{m}_{ri})$.

Replicate-wise contrastive learning. We use the latent space representations $\boldsymbol{x}_{ri}$ for computing the cross-replicate contrastive loss. We follow SimCLR (Chen et al. 2020), where the replicates are augmentations and the same segments across replicates are positive examples, otherwise they are negative examples. The pairwise replicate contrastive loss l_1,

$$-\frac{1}{S \times \binom{R}{2}} \sum_{i=1}^{S} \sum_{1 \le r' \le r \le R} \log \frac{\exp\left(\frac{\boldsymbol{x}_{ri}^{\top} \boldsymbol{x}_{r'i}}{\| \boldsymbol{x}_{ri} \| \, \| \boldsymbol{x}_{r'i} \|} / \tau_1\right)}{\sum_{j \ne i}^{S} \exp\left(\frac{\boldsymbol{x}_{ri}^{\top} \boldsymbol{x}_{r'j}}{\| \boldsymbol{x}_{ri} \| \, \| \boldsymbol{x}_{r'j} \|} / \tau_1\right)}, \quad (1)$$

where τ_1 is the temperature hyperparameter (default: 0.5), aims to learn lower-dimensional representations such that positive examples are close and negative examples are distant in the new space.

Segment class learning. Assuming the actual peak/nonpeak status of genomic segments is shared across replicates and there are underlying characteristics of coverage that define peaks and nonpeaks, we expect the low-dimensional representation of peak segments to cluster together and separate from the nonpeak segments in the new space. Therefore, we also require the representations to match in discrete (classification) space, which we achieve by requiring peak probabilities for each segment to be similar across replicates. The embedded representations $\boldsymbol{x}_{ri}$ are reduced to two dimensions via a fully connected neural network (multilayer perceptron, MLP, in Fig. 1) with one hidden layer the same dimension as $\boldsymbol{x}_{ri}$, followed by the softmax function, together denoted as $g(\cdot)$. Letting $\boldsymbol{q}_{ri} = g(\boldsymbol{x}_{ri})$ be the peak/nonpeak probabilities for segment i in replicate r and $\boldsymbol{p}'_{rk} = (q_{r1k}, q_{r2k}, \ldots, q_{rSk})$, $k \in \{1, 2\}$, vectors of peak/nonpeak probabilities across segments for the rth replicate, we maximize similarity in peak calls among replicates using loss l_2,

$$-\frac{1}{2 \times \binom{R}{2}} \sum_{k=1}^{2} \sum_{1 \le r' \le r \le R} \log \frac{\exp\left(\frac{\boldsymbol{p}_{rk}^{\top} \boldsymbol{p}_{r'k}}{\| \boldsymbol{p}_{rk} \| \, \| \boldsymbol{p}_{r'k} \|} / \tau_2\right)}{\exp\left(\frac{\boldsymbol{p}_{rk}^{\top} (\boldsymbol{1} - \boldsymbol{p}_{r'k})}{\| \boldsymbol{p}_{rk} \| \, \| \boldsymbol{1} - \boldsymbol{p}_{r'k} \|} / \tau_2\right)}, \quad (2)$$

where temperature hyperparameter $\tau_2 = \tau_1$ in our experiments. This loss strengthens the shared peak signal across replicates and provides a peak/nonpeak prediction for each segment of each replicate.

Autoencoder learning. We also want to produce cleaner data in the original genomic space, useful for purposes such as visualization or replicate merging. Therefore, we use decoder $d(\cdot)$, with structure symmetric to the encoder $e(\cdot)$, to map $\boldsymbol{x}_{ri}$ back to predicted data $\hat{\boldsymbol{m}}_{ri}$ in the genomic space. An autoencoder has good embedded feature representation capability (Baldi 2011), learned by minimizing the squared error loss l_3,

$$\frac{1}{S \times R} \sum_{i=1}^{S} \sum_{r=1}^{R} \mathrm{MSE}(\boldsymbol{m}_{ri}, \hat{\boldsymbol{m}}_{ri}), \quad (3)$$

between the original data $\boldsymbol{m}_{ri}$ and the reconstructed data $\hat{\boldsymbol{m}}_{ri}$.

Performance benchmarking using ChromHMM annotations

We compared the performance of RCL with both unsupervised (MACS, ChIP-R, and HMMRATAC) and pretrained supervised (LanceOtron) peak callers, where we used data from four human cell lines, MCF-7, A549, K652, and GM12878, and one data set generated from mouse placenta tissues at embryonic day 9.5. The data sets are summarized in Table 1.

The RCL method involves one important tunable parameter—the coverage threshold t (option -t) used to identify the candidate peak regions and segments for model training. By default, RCL uses a chromosome-specific threshold that depends on the median coverage (see section "Prediction region selection"). In all data sets, in addition to using this default setting, we also implemented RCL with -t 2 to explore the impact of this tuning parameter. In data sets with higher library size (MCF-7, K562, and A549), chromosome-specific thresholds generally exceed two (Supplemental Fig. S1); default thresholds for lower library size data sets (GM12878 and mouse placenta) for all chromosomes are one. We observed lowering of threshold t increases the number of candidate regions supplied to the RCL model.

Across all tested data sets of varying library size, RCL achieved the best overall performance (Table 2A, 2B). Sporadically, HMMRATAC, LanceOtron, or ChIP-R achieved higher precision at the cost of much lower recall. As the threshold t decreases, RCL predicts more peaks with lower precision and higher recall. Overall, the model trained with lower threshold (RCL-C2 for MCF-7, K562, and A549; RCL-MED for GM12878 and mouse placenta) achieved universally better F1 scores, suggesting that exposure to more low coverage regions can help RCL distinguish true peaks. In these comparisons, MACS and ChIP-R peaks were called with Q-value 0.05, but HMMRATAC, LanceOtron, and RCL peaks were called without false discovery control. HMMRATAC calls should be filtered by the score (Tarbell and Liu 2019), typically a measure of coverage, and it is similarly advisable to filter RCL calls for higher precision.

Table 1. Data sets used to compare methods on genome-wide annotation regions generated by ChromHMM

						ChromHMM labels		Number of called peaks in ChromHMM labels					
											RCL		
Data	Src.	Org.	*R*	Size (M)	Map rate (%)	+	–	MACS	ChIP-R	HMMR ATAC	MED	C2	Lance Otron
MCF-7	The ENCODE Project Consortium 2012	Human	2	37	96	148,531	116,729	65,440	93,860	113,362	92,333	116,367	90,050
A549	The ENCODE Project Consortium 2012	Human	3	157	98	154,274	120,726	116,654	133,121	80,991	148,725	213,263	94,284
GM12878	Buenrostro et al. 2013	Human	4	31	68	156,817	139,567	58,341	45,498	47,315	59,130	38,637	33,056
K562	The ENCODE Project Consortium 2012	Human	3	100	96	212,779	190,127	143,269	139,343	87,284	173,790	227,839	110,011
Placenta	Starks et al. 2019	Mouse	3	20	70	88,211	18,754	14,598	15,888	6,277	11,332	5719	4141

Src., literature source. Org., organism. *R*, number of biological replicates. Size, mean number of reads across replicates in the data set after filtering. Map rate, median proportion of aligned reads to autosomal and sex chromosomes using Bowtie 2. ChromHMM labels, number of positive and negative true regions annotated using ChromHMM. RCL MED (default threshold t based on median coverage) and C2 (threshold $t=2$) indicate two different coverage thresholds used to build training segments. Lower threshold results in larger input data sets and provides more predictions, specifically including predictions for lower coverage, harder-to-predict segments. For the total number of peaks including those outside of annotated regions, see Supplemental Table S1.

Precision recall (PR) curves are useful for comparing methods across all false discovery rates (Fig. 2; Supplemental Fig. S2). We applied a relaxed *Q*-value threshold (0.5) to generate candidate peaks for MACS and ChIP-R, and post hoc thresholded to plot the curves. Since MACS and ChIP-R are usually run with smaller *Q*-values and no post hoc thresholding, we also plot precision and recall point estimates for typical choices of *Q* (methods labeled "multiQ"). The linear portion of each PR curve from the black dot to 100% recall corresponds to the subset of ChromHMM-labeled regions with no score assigned by the method. The RCL PR curve, especially with lower threshold *t*, dominates the curves of other methods. RCL appears to use replicate information in the coverage data better than ChIP-R's post hoc comparison of peak calls across replicates, which is generally better than naive aggregation of MACS calls. HMMRATAC achieves intermediate performance (Table 2A, 2B), with higher achievable recall than MACS and ChIP-R of weak peaks, but sometimes lower achievable precision on strong peaks (Fig. 2). HMMRATAC also performs poorly on data with lower library size, probably because coverage data are too sparse, when partitioned by fragment length, to estimate this parameter-rich model. Despite the high number of predicted peaks for K562 and A549 data (Table 1), RCL maintained good precision out to much higher recall. In particular, RCL achieved nearly twice as many true predictions while maintaining higher precision than either MACS or LanceOtron. While ChIP-R, and sometimes HMMRATAC, can achieve near equal performance on the strongest peaks, only RCL can maintain high precision on the more difficult peaks. For lower library size GM12878 and mouse data, all methods called a limited number of peaks (Table 1), with low achievable recall. Nevertheless, RCL still obtained better performance, except at the highest achieved recall, where LanceOtron had higher precision (Table 2A, 2B). The slightly lower PRAUC of RCL on these data (Table 2A) may yet be overcome by allowing noninteger threshold values *t* on the average coverage across multiple neighboring sites.

Performance benchmarking using transcription factor ChIP-seq data

In addition to genome annotations obtained with ChromHMM, we used TF ChIP-seq data to evaluate method performance. These data mark potential binding sites of various TFs, which bind where DNA is accessible and should coincide with ATAC-seq peaks. Due to the lack of TF ChIP-seq data generated in matching conditions, tool performance on the mouse placenta data was not evaluated using this metric. In the data sets where suitable labels were available, RCL achieved the highest precision (Table 2C). As observed with ChromHMM labels, RCL precision improved upon lowering the threshold *t*.

Gene Ontology analysis

As ChromHMM and TF ChIP-seq labels do not cover the whole genome and all methods predicted peaks outside these labeled regions, we analyzed the biological functions of genes associated with peaks called by each method (see "Gene Ontology analysis" in Methods section "Method comparison"). We expect meaningful peaks to associate with genes that are related to the known functions of the cell types or tissues. For example, we expect MCF-7 peaks to be enriched for processes such as epithelial cell proliferation, migration and invasion, as well as angiogenesis (Comşa et al. 2015). We therefore checked for the enrichment of any Gene Ontology (GO) term containing words "epithelial," "epithelium," or "angiogenesis." The K562 cell line has antiapoptotic characteristics (Kuželová et al. 2004); therefore, we expect the enrichment of processes related to the negative regulation of apoptosis, and searched for terms that contained the words "apoptosis" or

Table 2. Precision (Prec.), recall, F1 scores, and AUC (area under the PR curve, PRAUC in the main text) for human (A) and mouse (B) data sets. (C) Precision using transcription factor (TF) ChIP-seq as labels

(A) Human data sets

	MCF-7				A549				GM12878				K562			
	Prec.	**Recall**	**F1**	**AUC**	**Prec.**	**Recall**	**F1**	**AUC**	**Prec.**	**Recall**	**F1**	**AUC**	**Prec.**	**Recall**	**F1**	**AUC**
RCL-C2	0.848	**0.637**	**0.728**	**0.858**	0.791	**0.739**	**0.764**	**0.855**	**0.931**	0.229	0.368	0.763	0.686	**0.948**	**0.796**	**0.914**
RCL-MED	**0.909**	0.422	0.576	0.818	0.778	0.636	0.700	0.830	0.882	**0.333**	**0.483**	0.761	0.782	0.754	0.768	0.874
HMMRATAC	0.781	0.596	0.676	0.808	**0.850**	0.349	0.495	0.783	0.850	0.256	0.393	0.738	0.827	0.435	0.571	0.799
MACS	0.779	0.441	0.563	0.748	0.746	0.383	0.506	0.695	0.880	0.218	0.350	0.739	0.740	0.473	0.577	0.746
ChIP-R	0.768	0.412	0.536	0.781	0.733	0.358	0.481	0.725	0.877	0.181	0.300	0.734	0.680	0.480	0.563	0.762
LanceOtron	0.842	0.510	0.635	0.809	0.842	0.435	0.574	0.783	0.911	0.192	0.317	**0.764**	**0.828**	0.506	0.628	0.791

(B) Mouse data set **Mouse placenta**	**Prec.**	**Recall**	**F1**	**AUC**
RCL-C2	0.996	0.0648	0.122	0.926
RCL-MED	**0.999**	**0.128**	**0.227**	**0.927**
HMMRATAC	0.993	0.071	0.132	0.915
MACS	**0.999**	0.078	0.144	0.923
ChIP-R	**0.999**	0.090	0.166	0.923
LanceOtron	0.991	0.047	0.089	0.921

(C) Precision against ChIP-seq labels	**MCF-7**	**A549**	**GM12878**	**K562**
RCL-C2	**0.656**	**0.929**	0.361	**0.649**
RCL-MED	0.541	0.753	**0.496**	0.562
HMMRATAC	0.593	0.560	0.357	0.227
MACS	0.571	0.682	0.422	0.329
ChIP-R	0.531	0.675	0.422	0.327
LanceOtron	0.547	0.549	0.269	0.257

To compute precision, recall, and F1 scores for MACS and ChIP-R, a *Q*-value of 0.05 was used. To compute PRAUC for MACS and ChIP-R, a Q-value of 0.5 was used and the scores post hoc thresholded to obtain a PR curve. All HMMRATAC results were obtained using scores > 0. All RCL and LanceOtron results were obtained using average scores across replicates > 0.5.

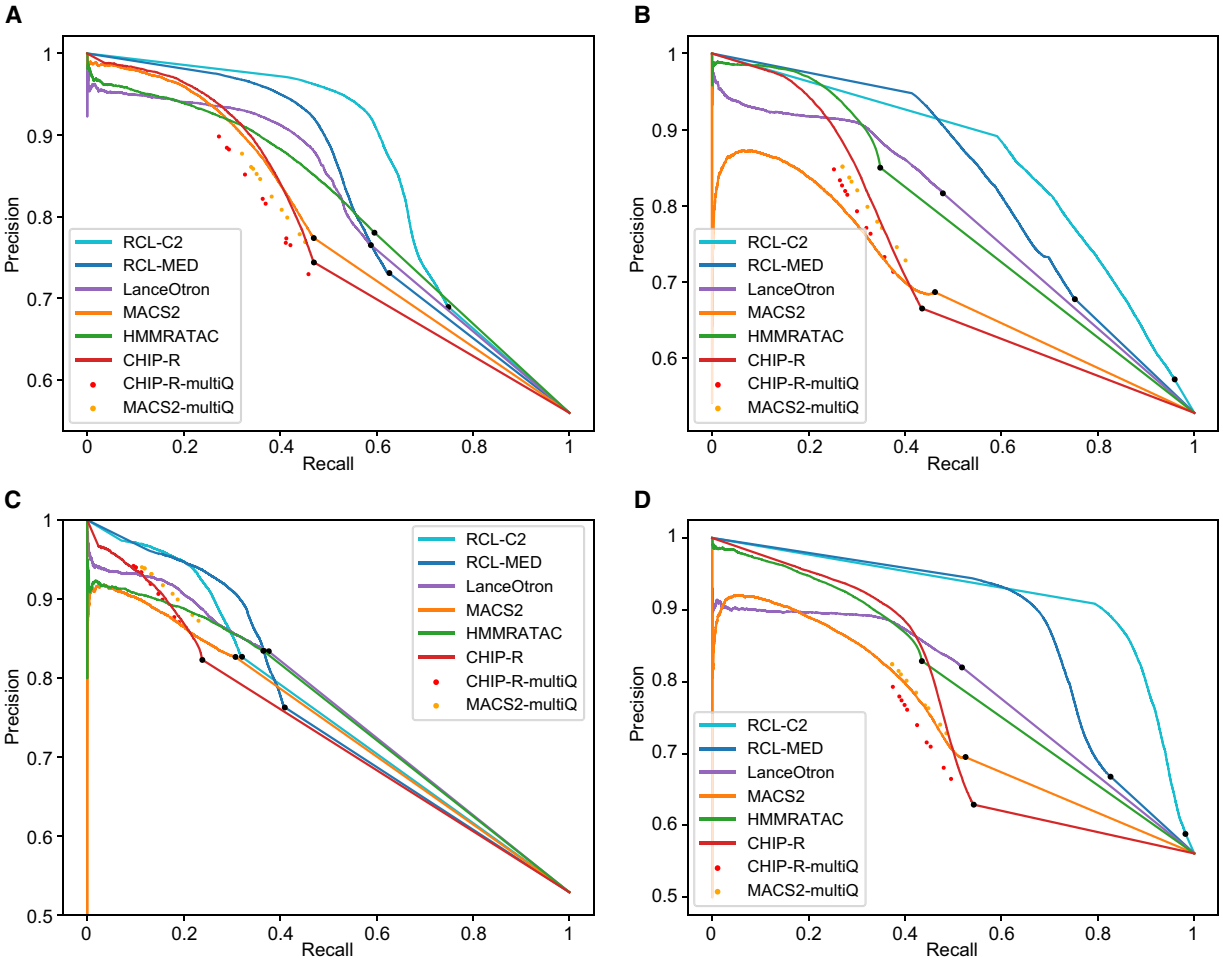

Figure 2. PR curves for ChromHMM-labeled regions, tested with data sets MCF-7 (*A*), K562 (*B*), GM12878 (C), and A549 (*D*). Black dot in each curve denotes the region with lowest score; all remaining ChromHMM-labeled regions are not scored by the method. RCL-C2, analysis with coverage threshold 2; RCL-MED, analysis with default "median" coverage threshold; MACS-multiQ and ChIP-R-multiQ dots are obtained by varying *Q*-value cutoffs.

"apoptotic." The cell line A549, a type of lung carcinoma epithelial cell, is an alveolar type II (ATII) cell that secretes surfactant protein to maintain homeostasis (Lee et al. 2018). Hence, processes underlying this cell type are related to terms that include "epithelial," "epithelium," and "surfactant." GM12878 is a human lymphoblastoid cell line generated by transforming primary B cells from peripheral blood with Epstein-Barr virus (EBV) (Bird et al. 1981; Anderson and Gusella 1984). Therefore, processes involving "B cell" should be enriched if biologically relevant peaks are supplied. Last, in the mouse placenta at day 9.5, the labyrinth layer is actively developing after chorioallantoic attachment finishes; as a result, a dense network of fetal blood vessels are forming within the layer where nutrients are exchanged (Cross et al. 2003; Watson and Cross 2005; Starks et al. 2021). In addition, the placenta is comprised mostly of trophoblast cells, which are epithelial-like cells. Thus, processes related to "placenta," "epithelium," "vasculature," "angiogenesis," "labyrinth," and "insulin" should be expected in meaningful peaks from day 9.5 mouse placenta tissue.

In general, we observed that only peaks uniquely called by RCL are enriched with relevant biological terms, with the exception of A549 data (Fig. 3; Supplemental Figs. S3–S7; Supplemental Tables S2–S6). For example, peaks that only RCL identified were associated with processes related to apoptosis in the K562 data set (Fig. 3). In case RCL benefitted from simply predicting a higher number of peaks, we randomly downsampled all peak sets and repeated the enrichment analysis. RCL continued to enrich for functionally relevant processes (Supplemental Figs. S3–S7; Supplemental Tables S2–S6). For relevant terms, RCL peaks are often associated with at least five genes and have higher than twofold enrichment (vertical line, Fig. 3; Supplemental Figs. S3–S7) unlike the other methods, suggesting RCL peaks are more likely to be associated with relevant genes than peaks identified by competing methods. In summary, there is evidence that unique peaks predicted by RCL, not just those overlapping ChromHMM- or TF-derived labels, are biologically relevant.

Discussion

We propose RCL, an unsupervised peak caller for ATAC-seq data using contrastive learning across biological replicates. In our model, three losses—replicate similarity loss, class similarity loss, and autoencoder loss—are learned simultaneously. We use ResNET as our backbone module with only five layers, making the network architecture shallow but efficient. On a server containing two Tesla V100 (16 GB) GPUs, the training time is 118 sec when there are 4828 1000 bp regions and four replicates. Empirical results

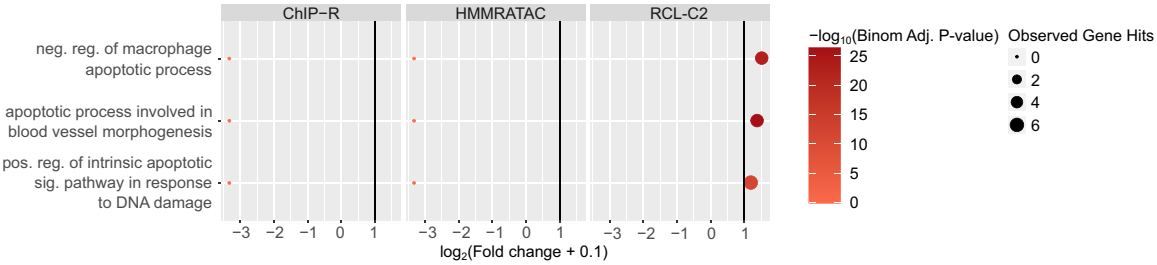

Figure 3. Gene Ontology analysis using unique peaks called by each method in K562 data. Only relevant terms enriched in at least one peak set are plotted. Colors correspond to $-\log_{10}$(Binomial Adjusted *P*-value) where the adjustment was done following the Benjamini–Hochberg procedure (Benjamini and Hochberg 1995); dot sizes correspond to the observed number of genes associated with the term; *x*-axis corresponds to $\log_2$(Fold change + 0.1) and vertical line is fold change of two. LanceOtron was not plotted since there was no unique peak called by the tool. (reg.) Regulation, (pos.) positive, (neg.) negative, (sig.) signaling.

indicate RCL training time is roughly linear in the number of segments. In theory, training time is quadratic in the number of replicates because of the contrastive loss calculation, but replicate numbers remain quite low. Further investigation on timing is warranted, but total run times were acceptable on all data sets tested here. For example, for the A549 data set (the largest data set and slowest to train), the training time (25 epochs) took about 62 min.

In practice, only a small proportion of the genome is accessible (The ENCODE Project Consortium 2012). As a result, data sets for peak calling tend to be highly imbalanced, making it challenging to separate peak and nonpeak regions. RCL showed no problems with class imbalance, probably because the region selection step effectively discards nonpeak regions and balances the data. If class imbalance proves to be a problem for calling data sets with sparser peaks or more widely across the genome in high coverage data sets, there are opportunities for improvement. For example, due to similarities to deep embedding clustering (Xie et al. 2016), cluster regularization methods proposed to avoid local optima or trivial solutions favoring predictions of the larger class (Tao et al. 2018; Zhong et al. 2020) may be applicable to contrastive learning and RCL.

Highly variable peaks or peaks in low coverage data may be difficult to find from single replicates, but their signal may become obvious when comparing across multiple replicates. HMMRATAC uses multiple replicates by combining them, which reduces the variance in the signal, but does not help the method learn what defines noise in a single replicate. ChIP-R, a post hoc method to combine peaks called by another method, can improve performance over MACS, but only when used with a liberal *Q*-value threshold followed by furthering filtering of ChIP-R-predicted peaks (red PR curves in Fig. 2). Although both MACS and RCL make predictions for individual replicates, RCL predicts after learning from all replicates, while MACS predicts after learning from only the replicate in question. Currently, we combine the RCL prediction scores by taking the mean across replicates, but one can imagine more sophisticated approaches to combine predictions across replicates, possibly assessing the quality of prediction from each replicate and weighting the mean.

Replicates are, by design, an essential component of our method. To demonstrate the value of biological replicates we conducted an ablation study (Supplemental Text S3.1.2). Contrasting real biological replicates gave the best predictions across chromosomes, which is not surprising given that biological replicates are fundamental for reproducibility and false signal reduction (Yan et al. 2020). In the absence of biological replicates, contrasting with an augmentation of the available data is better than contrasting with self. It could be that noise along the genome recapitulates some of the noise between biological replicates, but more study is necessary to understand RCL performance in the absence of replicates. Experiments varying the number of replicates available to RCL showed little effect on performance, even when the added replicate had substantially higher coverage (Supplemental Text S3.3; Supplemental Fig. S10). All the data examined in the current study were of high quality with minimal batch effects and mostly from cultured cells with likely little biological variation, all of which may explain the limited impact of additional replicates. It will be an interesting future direction to examine how contrastive learning and the RCL framework handle noisier replicates, batch effects, or the inclusion of low quality replicates.

We acknowledge that the labeled regions indicating the "ground truth" used for assessment are noisy. First, the annotations obtained using ChromHMM (Ernst and Kellis 2017) applied to several ChIP-seq data sets contain technical noise from data generation and model estimation. The TF ChIP-seq labels were specifically called by MACS2 (Zhang et al. 2008), which we know produces noisy, imperfect labels. Second, while we matched cell types and biological conditions, variation in the samples used to generate TF ChIP-seq or ChromHMM labels were not completely controlled. Third, our translations from ChromHMM states to open/closed regions were imperfectly determined to the best of our knowledge. There appear to be noisy truth labels in the MCF-7 data. Some negative ChromHMM regions were assigned high scores (logit-transformed scores > 10) (Supplemental Fig. S9). Although these score assignments could be due to the shortcomings of RCL, it is also possible that some labels are wrong. Further investigation will be enabled when the quality of labels is improved.

When there is noise in the labels, the observed performance metrics (precision, recall, F1, and PR curve) are not equal to the true performance metrics evaluated against the truth (Jiang et al. 2014). Moreover, the observed recall is a function of true recall *and* the true false positive rate. Specifically, let $\hat{y}$ be predicted labels, y unobserved true labels, and z observed noisy labels. Furthermore, suppose the labeling error rates $P(z = 0|y = 1) = P(z = 1|y = 0) = \epsilon$ are constant and independent of any signal in the data. Then, the observed recall is

$$P(\hat{y} = 1|z = 1) = P(\hat{y} = 1|y = 1)P(y = 1|z = 1) + P(\hat{y} = 1|y = 0)P(y = 0|z = 1),$$

where $P(\hat{y} = 1|y = 0)$ is the true false positive rate (FPR). Thus, observed recall is a contaminated measure of recall, and methods compared via observed recall (or F1 or PR curve) may not reveal

their actual ranking. Given this concern, it is possible to estimate method performance *in the context of label errors* (Raykar et al. 2009; Yan et al. 2014) or correct errored labels so traditional assessment metrics are more accurate (Sabetpour et al. 2021; Zheng et al. 2021). Alternatively, performance evaluation can be carried out with simulated data. However, there is no existing simulation method for ATAC-seq data, and the existing methods used for ChIP-seq, such as Zheng et al. (2022), are not applicable for ATAC-seq.

RCL learns and predicts on fixed-size segments (length α, default 1000 bp). We did not examine the impact of hyperparameter α on RCL performance, but it certainly complicates peak calling. We chose to transfer RCL prediction scores from the α bp segments to the variable-length candidate regions produced by the algorithm in section "Prediction region selection," because it works well (Supplemental Text S3.4). Using these candidate regions with mean coverage as a simple score already does well in MCF-7, but RCL learns additional signals, perhaps peak shape, that further improve the performance (Supplemental Fig. S11). Not only do the candidate regions work well, but they are not easily substituted. Using the α bp segments as peak predictions in A549 failed, probably because they lack the resolution to pinpoint narrow peaks, but a quick and dirty attempt to shrink the prediction regions to the relevant peak summit performed even worse (Supplemental Fig. S11). A better solution may be to learn and predict directly on the variable-sized candidate regions. We could pad variable-sized inputs to the same length or we could add a spatial pyramid pooling layer (He et al. 2015) before the first fully connected layer to remove the fixed-size constraint of the network. On the other hand, such an approach would still require data preprocessing to choose the candidate regions. An even better solution might be to predict at the nucleotide level, a one-step solution to identify peaks and their extent.

RCL can be extended and improved in other ways. First, we used simple read coverage as input, but HMMRATAC reports reproducible signal in the coverage of distinct fragment lengths around open regions (Tarbell and Liu 2019). RCL could be extended to take coverage vectors for multiple fragment lengths, the fragments themselves, or even annotation information, as used by the supervised method CNN-Peaks (Oh et al. 2020). Second, multiple hyperparameters in both data processing and model training can be further tuned. For example in input preparation, regions longer than 100 bp are kept in the current method. We have tried keeping regions longer than 147 bp, and it resulted in fewer inputs and fewer called peaks; however, we still obtained good predictions. Last, we have focused on ATAC-seq data, where peak calling has been particularly difficult because of the lack of control samples and good truth labels. Nevertheless, our model assumes nothing particular to ATAC-seq data and can be applied to ChIP-seq, CUT&RUN (Skene et al. 2018) and other techniques requiring peak calling.

There is clearly much left to learn about how RCL works to extract useful signal from replicates, but we can offer some preliminary recommendations. First, we recommend users follow established data preprocessing and quality control steps for ATAC-seq data (Yan et al. 2020). Where we have tested, few hyperparameters and inputs of RCL had much impact on performance, other than the coverage threshold *t*, option *-t*, and the candidate regions. The default threshold ("median") identified highly confident peaks with excellent precision (Fig. 2); therefore, this setting can be a good starting point for researchers to find the high confidence peaks. If a researcher wishes to predict more peaks accurately, it may be better to reduce threshold *t* and expose RCL to more and less obvious candidate peaks, a particularly good option for high coverage data sets, where RCL reproducibly outshines the competing methods. We recommend using all replicates under the assumption that replicates are still quite sparse because of cost. While additional replicates did not improve performance on data sets tested here, they also did not hurt performance. Finally, we provide no specific options to construct alternative prediction regions, but users may have good ideas for choosing candidate regions and they can try them in the RCL software package.

In summary, we have developed a novel peak calling framework for ATAC-seq data using contrastive learning techniques to extract signals shared across biological replicates and accurately identify open chromatin regions. Because RCL can predict more peaks with higher precision, it will facilitate future epigenome and chromatin accessibility studies in various biological contexts.

Methods

ATAC-seq data acquisition

ATAC-seq data sets of the following human cell lines and mouse tissues were obtained from public databases: MCF-7, A549, K562, GM12878, and mouse placenta. The MCF-7 data set, with two biological replicates, was accessed through the ENCODE experiment ID ENCSR422SUG (The ENCODE Project Consortium 2012). The A549 data set, with three biological replicates, was accessed through the ENCODE experiment ID ENCSR032RGS (The ENCODE Project Consortium 2012). The K562 data set, with three biological replicates, was accessed through the ENCODE experiment ID ENCSR868FGK (The ENCODE Project Consortium 2012). The GM12878 data set generated using 50,000 cells was obtained from four replicates with the NCBI Sequence Read Archive (SRA; https://www.ncbi.nlm.nih.gov/sra) accession numbers SRR891268, SRR891269, SRR891270, and SRR891271 (Buenrostro et al. 2013). Last, the mouse data generated from mouse placenta at day 9.5, with three biological replicates, was accessed from SRA under accession numbers SRR7912013, SRR7912014, and SRR7912015 (Starks et al. 2019).

ATAC-seq data processing

FASTQ files were assessed using FastQC (version 0.11.7) (Andrews 2010) to identify samples with overrepresented sequences or adapter contamination. Trimmomatic (Bolger et al. 2014) was used to remove adapter content and filter low quality base pairs and reads (`ILLUMINACLIP:overpresentedSeq.fa:2:30:10:2:keepBothReads LEADING:3 TRAILING:3 MINLEN:36`, other settings: default, version 0.39). Here, the `overpresentedSeq.fa` file contains the overrepresented sequences and adapter content identified with FastQC. Reads were aligned to the autosomal and sex chromosomes of human reference genome GRCh38 or mouse reference genome GRCm38 (release 98) (Cunningham et al. 2019) using Bowtie 2 (Langmead and Salzberg 2012) (`-X 1000 --no-discordant`, other settings: default, version 2.3.4.1). The genome reference versions were chosen to match those of the label files downloaded from public databases and used for performance assessment (see "Method comparison"). Matching reference versions between raw data and labels ensures performance metrics are not affected by technical differences in annotations. Picard (https://broadinstitute.github.io/picard/) was used to remove duplicate reads (`REMOVE_DUPLICATES = true`, version 2.17.0). Reads with low quality mapping (`MapQ < 20`) were removed before merging, sorting, and indexing the resulting BAM files with SAMtools (Danecek et al. 2021). Last, to assess sample quality after preprocessing, ataqv (Orchard et al. 2020) (`--ignore-read-groups`, other settings:

default, version 1.2.1) was used to check for fragment length distribution and transcription start site (TSS) enrichment. Samples used for downstream analyses must have a mononucleosome peak in the fragment length distribution and TSS enrichment ≥ 1.5.

Tuning RCL

We used dilation 8 and kernel size 31 to train our model. Other hyperparameters are set at default values (number of epochs = 25, batch size = 256, learning rate = 10^{-4}, and temperature $\tau_2 = \tau_1 = 0.5$). Details regarding choosing dilation 8, kernel size 31, and model development are discussed in Supplemental Text S3.1. Briefly, RCL was developed on the MCF-7 cell line data (see section "ATAC-seq data acquisition") using different truth labels than those presented in results. We will demonstrate that this *roughly* tuned RCL is already substantially superior to existing methods, not only on MCF-7 with a distinct truth, but on additional holdout data sets as well.

Method comparison

We compared RCL with MACS (Zhang et al. 2008), ChIP-R (Newell et al. 2021), HMMRATAC (Tarbell and Liu 2019), and LanceOtron (Hentges et al. 2021). Call performance was assessed using three analyses: comparisons using truth labels of genome annotation obtained with ChromHMM (Ernst and Kellis 2017) from independent data collected on the same cell lines and tissues, comparisons using truth labels of transcription factor ChIP-seq data collected on the same cell lines and tissues, and the association of peak prediction to biologically relevant genes.

MACS peak calling

MACS (version 2.1.1) (Zhang et al. 2008; Gaspar 2018) was used to call peaks with BAM files from individual replicates. Peaks were called with options `-g hg` (or `-g mm`) `-f BAMPE` `--bdg` `--keep-dup all`, with the following cutoffs for the *Q*-value `-q: 0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00002`, and `0.00001`, and other settings: default. Any peaks overlapping with a blacklist region (Amemiya et al. 2019) by at least 1 bp were removed. MACS was originally developed for calling peaks on transcription factor ChIP-seq data, so the default settings and model assumptions may not apply for ATAC-seq data. We have some evidence that altering shift and window sizes can improve MACS performance in some aspects (Supplemental Text S3.5; Supplemental Fig. S12), but settings to consistently improve MACS performance were elusive and beyond the scope of this work. Given peak calls from individual replicates, the peak union method was used to combine peaks across replicates. Specifically, a consensus peak set is the union of peaks overlapping with each other by ≥50% length in ≥2 replicates. Scores of consensus peaks were the mean $-\log_{10}$(Q-value) at the peak summit of the individual peaks observed in separate replicates. As MACS does not report scores of nonpeak regions, replicates not calling a peak in the region are not used when calculating scores of consensus peaks.

ChIP-R peak calling

We used ChIP-R (version 1.1.0) (Newell et al. 2021) as an additional, independent method for combining peaks called from MACS. Peaks were first called with MACS as described above. Then, ChIP-R was run with the following setting: `-m 2, -a 0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00002,` and `0.00001` (matching with `-q` in MACS), other settings: default. Any peaks overlapping with a blacklist region (Amemiya et al. 2019) by at least 1 bp were removed. The reported ChIP-R `score` was used as the final score of each ChIP-R peak.

HMMRATAC peak calling

HMMRATAC (version 1.2.4) (Tarbell and Liu 2019) was used to call peaks with a merged BAM file from all replicates, options `-Xmx128G`, `--window 250000`, other settings: default. A peak is a region in the open state with scores ≥ 0, reported by default with the `--peaks` option. By default, peak scores of HMMRATAC are the maximum read coverage of the called center state region. Any peaks overlapping with a blacklist region (Amemiya et al. 2019) by at least 1 bp were removed.

LanceOtron peak calling

To implement LanceOtron (version 1.0.8) (Hentges et al. 2021), the input bigWig files were obtained using deepTools (version 2.5.2) (Ramírez et al. 2016) with the following command: `bamCoverage -b bamFile -o bigWigFile --extendReads -bs 1 --normalizeUsing RPKM`. The inputs were then used to call peaks with default settings and LanceOtron's pretrained model. Any resulting peaks overlapping with a blacklist region (Amemiya et al. 2019) by at least 1 bp were removed. The peak union method, as defined above for MACS, was used to combine peaks across replicates into candidate consensus peaks. Scores of candidate consensus peaks were the mean `overall_peak_score` of the individual contributing peaks. Candidate consensus peaks with scores >0.5 were then defined as peaks by default.

RCL peak calling

RCL was used with coverage threshold *t* set to "median" and 2, other settings: default. By default, any segment overlapping with blacklist regions was excluded due to the segment selection procedure. Let peak prediction score given by RCL be $\xi_{ri} = q_{ri1}$ for the ith α bp segment in the rth replicate. We obtain a final peak prediction score for each region in $\mathcal{A}$ (see Step 2 in "Prediction region selection") by averaging over ξ_{ri} for all replicates $r = 1, 2, \ldots, R$ and segments i extracted from the region. A region in $\mathcal{A}$ is predicted to contain at least one peak if this score is >0.5.

Compilation of true positive– and true negative–labeled regions by ChromHMM

For human cell line data, we obtained genome annotations inferred with ChromHMM (Ernst and Kellis 2017) from ENCODE. Specifically, genome annotation for MCF-7 data was accessed via the experiment ID ENCSR579CCH, the A549 data via ENCSR283-FYU, and the GM12878 data via ENCSR988QYW. True positive regions are those marked "EnhA1," "EnhA2," "EnhG1," "EnhG2," "TssA," "TssFlnk," "TssFlnkD," "TssFlnkU," and "Tx." True negative regions are marked as "Het," "Quies," "ReprPC," and "ZNF/Rpts." Annotations not in these lists were not used, and regions overlapping with a blacklist region (Amemiya et al. 2019) by at least 1 bp were removed. For full definitions of the states, see Supplemental Table S7.

For the mouse placenta data, we obtained ChromHMM annotation from Starks et al. (2021). True positive regions are those belonging to States 8, 9, and 10, and true negative regions are those belonging to State 2. Detailed biological characterization of these states was described in Starks et al. (2021).

Compilation of true positives from transcription factor (TF) ChIP-seq data

For human cell line data, we obtained TF ChIP-seq data from matching cell lines from ENCODE (The ENCODE Project

Consortium 2012). BED files of IDR thresholded peaks were downloaded from all data sets that passed all quality control criteria of ENCODE and had "released" status, and their bio-samples were not perturbed. For mouse placenta data, no TF ChIP-seq data from matching conditions were available. Therefore, this analysis was not carried out for mouse placenta data.

True positive regions were defined as those with at least one TF ChIP-seq peak. Regions overlapping with a blacklist region (Amemiya et al. 2019) by at least 1 bp were removed. No true negative regions were defined using these data sets. For lists of data used, see Supplemental Table S7.

Calculation of evaluation metrics

Since labeled regions and called regions do not necessarily coincide, we defined a mapping function to transfer scores of called regions in the ATAC-seq data to predicted scores for annotated regions. Specifically, suppose there are n_i called regions overlapping with the ith labeled region, and c_j $(1 \le j \le n_i)$ is the predicted score that overlaps by o_j base pairs with the ith labeled region. Then the weighted prediction score for the ith labeled region is $\sum_{j=1}^{n_i} \frac{o_j c_j}{\sum_{v=1}^{n_i} o_v}$. In case $n_i = 0$, we assign the lowest weighted score observed for that method as the predicted score.

We used point estimates of precision, recall, and F1, as well as PR curves to compare the performance of the methods. Specifically, for MACS and ChIP-R, we calculated precision and recall for each `-q` (`-a`) cutoff. For plotting PR curves, we used MACS or ChIP-R with Q-value 0.5. Results from other Q-value settings were presented as point estimates on the plots. RCL can call more peaks by lowering the threshold t, but changing t will also change the fitted model and the peak calls made. We always ran RCL with defaults, but we also selected nondefault $t = 2$ to explore the impact.

Gene Ontology analysis

To assess the potential functional roles of the called peaks by all methods, we used Gene Ontology analysis. We examined the following sets of peaks. For the MCF-7, K562, and A549 data, we used ChIP-R, HMMRATAC, LanceOtron, and RCL peaks called when $t = 2$ peaks; for the GM12878 and mouse placenta data, ChIP-R, HMMRATAC, LanceOtron, and RCL peaks called when $t =$ *median* peaks; and for all data sets, peaks identified uniquely by each of these methods. Specifically, a peak is uniquely assigned to a method if it does not overlap with peaks predicted by any other method, as assessed using BEDTools `intersect -v` (Quinlan and Hall 2010).

We used the Genomic Regions Enrichment of Annotations Tool (GREAT) (version 4.0.4) (McLean et al. 2010; Tanigawa et al. 2022) implemented in R (Gu and Hübschmann 2023) to carry out GO enrichment using either the human GRCh38 or mouse GRCm38 annotations and the default basal plus extension association rule. For each analysis, we randomly selected peaks so that the number of input regions for GREAT was the smallest or second smallest peak set size amongst all tools (see number of peaks from each tool in Supplemental Table S1). For unique peaks, we also analyzed all peaks without down-sampling. A biological process term was considered enriched if its binomial q-value ≤ 0.05, binomial fold change ≥ 2, and the observed number of associated genes was ≥ 5.

Software availability

The entire pipeline is released under the GNU Public License to the community as a package named RCL, for Replicative Contrastive Learner, at GitHub (https://github.com/Tuteja-Lab/UnsupervisedPeakCaller). The source code can also be found as Supplemental Code. The pipeline uses SAMtools (Danecek et al. 2021), BEDTools (Quinlan and Hall 2010), parallel (Tange 2011), BEDOPS (Neph et al. 2012), R (R Core Team 2023), R package dplyr (Wickham et al. 2019), R package bedr (https://cran.r-project.org/package=bedr), R package doParallel (https://github.com/RevolutionAnalytics/doparallel), several Python packages including PyTorch (Paszke et al. 2019), numpy (Harris et al. 2020), pandas (McKinney 2010; doi:10.5281/zenodo.3509134), and scikit-learn (Pedregosa et al. 2011).

Competing interest statement

The authors declare no competing interests.

Acknowledgments

We acknowledge the Research IT group at Iowa State University (http://researchit.las.iastate.edu) for providing servers and IT support. We thank Dorman laboratory and Tuteja laboratory members for their discussion and support. This work was supported in part by the Eunice Kennedy Shriver National Institute of Child Health & Human Development of the National Institutes of Health under Award No. R01HD096083 (to G.T.). G.T. is Pew Scholar in the Biomedical Sciences, supported by The Pew Charitable Trusts. This work was supported in part by the United States Department of Agriculture (USDA) National Institute of Food and Agriculture (NIFA) Hatch Project No. IOW03717. The findings and conclusions in this publication are those of the author(s) and should not be construed to represent any official USDA or U.S. Government determination or policy.

References

Amemiya HM, Kundaje A, Boyle AP. 2019. The ENCODE blacklist: identification of problematic regions of the genome. *Sci Rep* **9:** 9354. doi:10.1038/s41598-019-45839-z

Anderson MA, Gusella JF. 1984. Use of cyclosporin A in establishing Epstein-Barr virus-transformed human lymphoblastoid cell lines. *In Vitro* **20:** 856–858. doi:10.1007/BF02619631

Andrews S. 2010. *FastQC: a quality control tool for high throughput sequence data. Version 0.11.7*. Babraham Institute, Cambridge, United Kingdom. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Baldi P. 2011. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of international conference on unsupervised and transfer learning workshop, vol. 27, UTLW '11*, pp. 37–50, JMLR.org, Bellevue, WA.

Benjamini Y, Hochberg Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc B Stat Methodol* **57:** 289–300. doi:10.1111/j.2517-6161.1995.tb02031.x

Bird AG, McLachlan SM, Britton S. 1981. Cyclosporin A promotes spontaneous outgrowth in vitro of Epstein-Barr virus-induced B-cell lines. *Nature* **289:** 300–301. doi:10.1038/289300a0

Bolger AM, Lohse M, Usadel B. 2014. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30:** 2114–2120. doi:10.1093/bioinformatics/btu170

Buenrostro JD, Giresi PG, Zaba LC, Chang HY, Greenleaf WJ. 2013. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nat Methods* **10:** 1213–1218. doi:10.1038/nmeth.2688

Buenrostro JD, Wu B, Chang HY, Greenleaf WJ. 2015. ATAC-seq: a method for assaying chromatin accessibility genome-wide. *Curr Protoc Mol Biol* **109:** 21.29.1–21.29.9. doi:10.1002/0471142727.mb2129s109

Chen T, Kornblith S, Norouzi M, Hinton G. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th international conference on machine learning, vol. 119, Proceedings of machine learning research*, pp. 1597–1607, ML Research Press.

Comşa Ş, Cîmpean AM, Raica M. 2015. The story of MCF-7 breast cancer cell line: 40 years of experience in research. *Anticancer Res* **35:** 3147–3154.

Cross JC, Simmons DG, Watson ED. 2003. Chorioallantoic morphogenesis and formation of the placental villous tree. *Ann NY Acad Sci* **995:** 84–93. doi:10.1111/j.1749-6632.2003.tb03212.x

Cunningham F, Achuthan P, Akanni W, Allen J, Amode M, Armean I, Bennett R, Bhai J, Billis K, Boddu S, et al. 2019. Ensembl 2019. *Nucleic Acids Res* **47:** D745–D751. doi:10.1093/nar/gky1113

Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, Whitwham A, Keane T, McCarthy SA, Davies RM, et al. 2021. Twelve years of SAMtools and BCFtools. *GigaScience* **10:** giab008. doi:10.1093/gigascience/giab008

The ENCODE Project Consortium. 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489:** 57–74. doi:10.1038/nature11247

Ernst J, Kellis M. 2017. Chromatin-state discovery and genome annotation with ChromHMM. *Nat Protoc* **12:** 2478–2492. doi:10.1038/nprot.2017.124

Gaspar JM. 2018. Improved peak-calling with MACS2. bioRxiv doi:10.1101/496521

Goren E, Liu P, Wang C, Wang C. 2018. Binquasi: a peak detection method for ChIP-sequencing data with biological replicates. *Bioinformatics* **34:** 2909–2917. doi:10.1093/bioinformatics/bty227

Grandi FC, Modi H, Kampman L, Corces MR. 2022. Chromatin accessibility profiling by ATAC-seq. *Nat Protoc* **17:** 1518–1552. doi:10.1038/s41596-022-00692-9

Gu Z, Hübschmann D. 2023. rGREAT: an R/Bioconductor package for functional enrichment on genomic regions. *Bioinformatics* **39:** btac745. doi:10.1093/bioinformatics/btac745

Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, et al. 2020. Array programming with NumPy. *Nature* **585:** 357–362. doi:10.1038/s41586-020-2649-2

He K, Zhang X, Ren S, Sun J. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans Pattern Anal Mach Intell* **37:** 1904–1916. doi:10.1109/TPAMI.2015.2389824

He K, Zhang X, Ren S, Sun J. 2016. Deep residual learning for image recognition. In *Proc CVPR IEEE*, pp. 770–778.

Hentges LD, Sergeant MJ, Downes DJ, Hughes JR, Taylor S. 2021. LanceOtron: a deep learning peak caller for ATAC-seq, ChIP-seq, and DNase-seq. *Bioinformatics* **38:** 4255–4263. doi:10.1101/2021.01.25.428108

Hocking TD, Goerner-Potvin P, Morin A, Shao X, Pastinen T, Bourque G. 2017. Optimizing ChIP-seq peak detectors using visual labels and supervised machine learning. *Bioinformatics* **33:** 491–499. doi:10.1093/bioinformatics/btw672

Jiang Y, Clark W, Friedberg I, Radivojac P. 2014. The impact of incomplete knowledge on the evaluation of protein function prediction: a structured-output learning perspective. *Bioinformatics* **30:** i609–i616. doi:10.1093/bioinformatics/btu472

Kramer MA. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE J* **37:** 233–243. doi:10.1002/aic.690370209

Kuželová K, Grebeňová D, Pluskalová M, Marinov I, Hrkal Z. 2004. Early apoptotic features of K562 cell death induced by 5-aminolaevulinic acid-based photodynamic therapy. *J Photochem Photobio B* **73:** 67–78. doi:10.1016/j.jphotobiol.2003.07.007

Langmead B, Salzberg SL. 2012. Fast gapped-read alignment with Bowtie 2. *Nat Methods* **9:** 357–359. doi:10.1038/nmeth.1923

Lee DF, Salguero FJ, Grainger D, Francis RJ, MacLellan-Gibson K, Chambers MA. 2018. Isolation and characterisation of alveolar type II pneumocytes from adult bovine lung. *Sci Rep* **8:** 11927. doi:10.1038/s41598-018-30234-x

Le-Khac PH, Healy G, Smeaton AF. 2020. Contrastive representation learning: a framework and review. *IEEE Access* **8:** 193907–193934. doi:10.1109/ACCESS.2020.3031549

Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv:1303.3997 [q-bio.GN].

Li Q, Brown JB, Huang H, Bickel PJ. 2011. Measuring reproducibility of high-throughput experiments. *Ann Appl Stat* **5:** 1752–1779. doi:10.1214/11-AOAS466

McKinney W. 2010. Data structures for statistical computing in Python. In *Proceedings of the 9th Python in science conference* (ed. van der Walt S, Millman J), pp. 56–61, SciPy, Austin, TX.

McLean CY, Bristor D, Hiller M, Clarke SL, Schaar BT, Lowe CB, Wenger AM, Bejerano G. 2010. GREAT improves functional interpretation of *cis*-regulatory regions. *Nat Biotechnol* **28:** 495–501. doi:10.1038/nbt.1630

Musich R, Cadle-Davidson L, Osier MV. 2021. Comparison of short-read sequence aligners indicates strengths and weaknesses for biologists to consider. *Front Plant Sci* **12:** 657240. doi:10.3389/fpls.2021.657240

Neph S, Kuehn MS, Reynolds AP, Haugen E, Thurman RE, Johnson AK, Rynes E, Maurano MT, Vierstra J, Thomas S, et al. 2012. BEDOPS: high-performance genomic feature operations. *Bioinformatics* **28:** 1919–1920. doi:10.1093/bioinformatics/bts277

Newell R, Pienaar R, Balderson B, Piper M, Essebier A, Bodén M. 2021. ChIP-R: assembling reproducible sets of ChIP-seq and ATAC-seq peaks from multiple replicates. *Genomics* **113:** 1855–1866. doi:10.1016/j.ygeno.2021.04.026

Oh D, Strattan JS, Hur JK, Bento J, Urban AE, Song G, Cherry JM. 2020. CNN-Peaks: ChIP-seq peak detection pipeline using convolutional neural networks that imitate human visual inspection. *Sci Rep* **10:** 7933. doi:10.1038/s41598-020-64655-4

Orchard P, Kyono Y, Hensley J, Kitzman JO, Parker SCJ. 2020. Quantification, dynamic visualization, and validation of bias in ATAC-seq data with ataqv. *Cell Syst* **10:** 298–306.e4. doi:10.1016/j.cels.2020.02.009

Park PJ. 2009. ChIP-seq: advantages and challenges of a maturing technology. *Nat Rev Genet* **10:** 669–680. doi:10.1038/nrg2641

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. 2019. Pytorch: an imperative style, high-performance deep learning library. In *Advances in neural information processing systems* (ed. Wallach H, et al.), Vol. 32, pp. 8026–8037. Curran Associates, Inc., Vancouver, Canada.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. 2011. Scikit-learn: machine learning in Python. *J Mach Learn Res* **12:** 2825–2830. doi:10.5555/1953048.2078195

Quinlan AR, Hall IM. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26:** 841–842. doi:10.1093/bioinformatics/btq033

Ramírez F, Ryan DP, Grüning B, Bhardwaj V, Kilpert F, Richter AS, Heyne S, Dündar F, Manke T. 2016. Deeptools2: a next generation web server for deep-sequencing data analysis. *Nucleic Acids Res* **44:** W160–W165. doi:10.1093/nar/gkw257

Raykar V, Yu S, Zhao L, Jerebko A, Florin C, Valadez G, Bogoni L, Moy L. 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th international conference on machine learning, ICML '09*, Montreal, Canada, pp. 889–896.

R Core Team. 2023. *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna. https://www.R-project.org/.

Rye M, Sætrom P, Drabløs F. 2011. A manually curated ChIP-seq benchmark demonstrates room for improvement in current peak-finder programs. *Nucleic Acids Res* **39:** e25. doi:10.1093/nar/gkq1187

Sabetpour N, Kulkarni A, Xie S, Li Q. 2021. Truth discovery in sequence labels from crowds. In *IEEE data mining, ICDM '21*, Auckland, New Zealand, pp. 539–548.

Singh AK, Mueller-Planitz F. 2021. Nucleosome positioning and spacing: from mechanism to function. *J Mol Biol* **433:** 166847. doi:10.1016/j.jmb.2021.166847

Skene PJ, Henikoff JG, Henikoff S. 2018. Targeted in situ genome-wide profiling with high efficiency for low cell numbers. *Nat Protoc* **13:** 1006–1019. doi:10.1038/nprot.2018.015

Starks R, Biswas A, Jain A, Tuteja G. 2019. Combined analysis of dissimilar promoter accessibility and gene expression profiles identifies tissue-specific genes and actively repressed networks. *Epigenetics Chromatin* **12:** 16. doi:10.1186/s13072-019-0260-2

Starks RR, Kaur H, Tuteja G. 2021. Mapping *cis*-regulatory elements in the midgestation mouse placenta. *Sci Rep* **11:** 22331. doi:10.1038/s41598-021-01664-x

Tange O. 2011. GNU parallel—the command-line power tool. *Usenix Mag* **36:** 42–47.

Tanigawa Y, Dyer ES, Bejerano G. 2022. WhichTF is functionally important in your open chromatin data? *PLoS Comput Biol* **18:** e1010378. doi:10.1371/journal.pcbi.1010378

Tao Y, Takagi K, Nakata K. 2018. RDEC: integrating regularization into deep embedded clustering for imbalanced data sets. In *Asian conference on machine learning, ACML '18*, pp. 49–64, PMLR, Beijing, China.

Tarbell E, Liu T. 2019. HMMRATAC: a hidden Markov modeleR for ATAC-seq. *Nucleic Acids Res* **47:** e91. doi:10.1093/nar/gkz533

Watson ED, Cross JC. 2005. Development of structures and transport functions in the mouse placenta. *Physiology* **20:** 180–193. doi:10.1152/physiol.00001.2005

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, et al. 2019. Welcome to the Tidyverse. *J Open Source Softw* **4:** 1686. doi:10.21105/joss.01686

Xie J, Girshick R, Farhadi A. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd international conference on machine learning, vol. 48, proceedings of machine learning research*, pp. 478–487, ML Research Press, New York.

Yan Y, Rosales R, Fung G, Subramanian R, Dy J. 2014. Learning from multiple annotators with varying expertise. *Mach Learn* **95:** 291–327. doi:10.1007/s10994-013-5412-1

Yan F, Powell DR, Curtis DJ, Wong NC. 2020. From reads to insight: a hitchhiker's guide to ATAC-seq data analysis. *Genome Biol* **21:** 22. doi:10.1186/s13059-020-1929-3

Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, et al. 2008. Model-based Analysis of ChIP-Seq (MACS). *Genome Biol* **9:** R137. doi:10.1186/gb-2008-9-9-r137

Zhang Y, Lin YH, Johnson TD, Rozek LS, Sartor MA. 2014. Pepr: a peak-calling prioritization pipeline to identify consistent or differential peaks from replicated ChIP-seq data. *Bioinformatics* **30:** 2568–2575. doi:10.1093/bioinformatics/btu372

Zheng G, Awadallah A, Dumais S. 2021. Meta label correction for noisy label learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35, 12, pp. 11053–11061, AAAI Press, Palo Alto, CA.

Zheng A, Lamkin M, Qiu Y, Ren K, Goren A, Gymrek M. 2022. A flexible ChIP-sequencing simulation toolkit. *BMC Bioinformatics* **22:** 1518–1552. doi:10.1186/s12859-021-04097-5

Zhong H, Chen C, Jin Z, Hua X. 2020. Deep robust clustering by contrastive learning. arXiv:2008.03030 [cs.CV].

Received January 7, 2023; accepted in revised form April 27, 2023.