Denoising Genome-wide Histone ChIP-seq with Convolutional Neural Networks

Pang Wei Koh 1,2,†, Emma Pierson 1,† and Anshul Kundaje 1,2,*

1 Department of Computer Science, Stanford University, Stanford, 94305, USA
2 Department of Genetics, Stanford University, Stanford, 94305, USA.

† These authors contributed equally to this work.
‡ To whom correspondence should be addressed.

1 Introduction

Distinct combinations of histone modifications are associated with different classes of functional genomic elements such as promoters, enhancers, and genes (Consortium et al., 2015). Chromatin immunoprecipitation sequencing (ChIP-seq) experiments targeting these histone modifications have been used to profile genome-wide chromatin state in diverse populations of cell types and tissues (Consortium et al., 2015), allowing us to better understand the mechanisms of development (Bernstein et al., 2006) and disease (Gjoneska et al., 2015).

However, the quality of histone ChIP-seq data is affected by a number of experimental parameters including antibody specificity and efficiency, library complexity, and sequencing depth (Jung et al., 2014). Achieving optimal experimental parameters and comparable data quality across experiments is often difficult, costly, or even impossible, resulting in low sensitivity and specificity of measurements especially in low input samples such as rare populations of primary cells and tissues (Brind’Amour et al., 2015; Cao et al., 2015; Acedo et al., 2007). For example, (Brind’Amour et al., 2015) found that single mouse embryos do not provide enough cells to profile using conventional ChIP-seq techniques. Similarly, (Acedo et al., 2007) notes that tumor biopsies, fractionated mixed cell populations, and differentiating embryonic stem cells provide very small numbers of cells to use as input populations. Further, the high sequencing depths (>50-100M reads) required for saturated detection of enriched regions in mammalian genomes for several broad histone marks (Jung et al., 2014) are often not met due to cost and material constraints. Suboptimal and variable data quality significantly complicate and confound integrative analyses across large collections of data.

To overcome these limitations, we introduce here a convolutional denoising algorithm, called Coda, that uses convolutional neural networks (CNNs) (Jain and Seung, 2009; Krizhevsky et al., 2012) to learn a generalizable mapping between ‘suboptimal’ and high-quality ChIP-seq data (Fig. 1). Coda substantially attenuates three primary sources of noise – due to low sequencing depth, low cell input, and low ChIP enrichment – enhancing signal in low-quality samples across individuals, cell types, and species. Our approach is conceptually related to the existing literature on structured signal recovery, in particular supervised denoising in images (Jain and Seung, 2009; Xie et al., 2012; Mousavi et al., 2015) and speech (Maas and Le, 2012). It complements other efforts to impute missing genomic data, such as ChromImpute (Ernst and Kellis, 2015), which predict profiles for a missing target mark in a target cell type (e.g., H3K4me3 in embryonic stem cells) by leveraging other available marks in the target cell type (e.g., H3K27ac in embryonic stem cells) and target mark datasets in other reference cell types (e.g., H3K4me3 in 100s of other celltypes). In contrast, our models take in low-quality signal of multiple target marks in a target cell type and denoise them all (e.g., using low-quality H3K27ac and H3K4me3 signal from a given cell population to produce higher-quality H3K27ac and H3K4me3 signal in that same cell population).

Neural networks have been successfully used to reduce noise in image data (Jain and Seung, 2009) and speech data (Maas and Le, 2012; Amodei et al., 2016), and there are several reasons to believe that neural networks could similarly denoise histone ChIP-seq data. First, histone marks have regular structure: peaks in each mark, for example, might tend to have
2 Methods

Model

Coda takes in a pair of matching ChIP-seq datasets of the same histone modifications in the same cell-type – one high-quality and the other noisy – and uses convolutional neural networks (CNNs) to learn a mapping from the noisy to the high-quality ChIP-seq data. The noisy dataset used in training can be derived computationally (e.g., by subsampling the high-quality data) or experimentally (e.g., by conducting the same ChIP-seq experiment with fewer input cells). Once this mapping has been learned, the same mapping can then be applied to new, noisy data in any other cellular context with the same underlying noise structure.

For each type of noise (e.g., due to low cell numbers, sequencing depth, or enrichment) and each target histone mark, we train two separate CNNs to accomplish two tasks: a regression task to predict histone ChIP signal (i.e., the fold enrichment of ChIP reads over input DNA control) and a binary classification task to predict the presence or absence of a significant histone mark peak (Fig. 2).

In total, if a given experiment has $M$ marks, then we train $2M$ models separately (one regression and one classification model for each mark). Each individual model optionally makes use of the noisy ChIP-seq data from all available marks but outputs only one target histone mark. This allows us to learn separate features for each mark and task while still leveraging information from multiple input histone marks; we find empirically that this improves performance.

For computational efficiency, we first bin the genome into 25bp bins, averaging the signal in each bin. Let $L$ be the number of bins in the genome (i.e., the length of the genome divided by 25). Each individual model takes in an $M \times L$ input matrix $X$ and returns a $1 \times L$ output vector $Y$ representing the predicted high-quality signal (in the regression setting) or peak calls (in the classification setting). It does this by feeding the noisy data through a first convolutional layer, a rectified linear unit (ReLU) layer, a second convolutional layer, and then a final ReLU or sigmoid layer (for regression or classification, respectively). For the first convolutional layer, we use 6 convolutional filters, each 51 bins in length; for the second convolutional layer, we use 6 convolutional filters, each 1001 bins in length. At the end of each convolutional layer, the network passes the output through a ReLU layer to compute the prediction at that location. To compute the prediction at the $i$-th bin, the network needs to perform $6 \times 51\times L$ operations at the first convolutional layer and $6 \times 1001\times L$ operations at the second convolutional layer. To compute the prediction at the $(i+1)$-th bin, the network needs to perform $6 \times 51$ more operations at the first convolutional layer and $6 \times 1001$ operations at the second convolutional layer, saving $6 \times 1001 \times 50$ operations. Other models, especially non-linear models such as random forests, would require a significantly higher computational cost.

The convolutional nature of our models (and the lack of max-pooling layers commonly seen in neural network architectures for computer vision) enables us to do efficient genome-wide prediction, as 98% of the computation required for predicting signal at the $i$-th bin is shared with the computation required for predicting the $(i+1)$-th bin. In particular, to compute the prediction at the $i$-th bin, the network needs to perform $6 \times 1001 \times 51$ operations at the first convolutional layer and $6 \times 1001$ operations at the second convolutional layer. To compute the prediction at the $(i+1)$-th bin, the network needs to perform $6 \times 51$ more operations at the first convolutional layer and $6 \times 1001$ operations at the second convolutional layer, saving $6 \times 1001 \times 50$ operations.
Denoising Genome-wide Histone ChIP-seq with Convolutional Neural Networks

3 Results

Removing noise from low sequencing depth data

A minimum of 40-50M reads is recommended for optimal sensitivity for histone ChIP-seq experiments in human samples targeting most canonical histone marks (Jung et al., 2014). As adhering to this standard can often be infeasible due to cost and other limitations, a substantial proportion of publicly available datasets do not meet these standards. Motivated by these constraints, we tested whether our model could recover high-read depth signal from low-read depth experiments.

Training and testing on the same cell type across different individuals

We evaluated Coda on lymphoblastoid cell lines (LCLs) derived from six individuals of diverse ancestry (European (CEU), Yoruba (YRI), Japanese, Han Chinese, San) (Kasowski et al., 2013). We used the CEU-derived cell line (GM12878) to train our model to reconstruct the high-depth signal (100M+ reads per mark; exact numbers in Data Availability and Processing) from a simulated noisy signal derived by subsampling 1M reads per mark. On the other five cell lines, Coda significantly improved Pearson correlation between the full and noisy signal (Fig. 3A, left) and the accuracy of peak calling (Fig. 3A, right). Using just 1M reads per mark, the predicted output of our model was equivalent in quality to signal derived from 15M+ reads (H3K27ac) and 25M+ reads (H3K36me3) (Fig. 3B). Fig. 4 shows how Coda can accurately reconstruct histone modification levels at the promoter of the PA53 gene, a master transcription factor required for differentiation into the B-lymphoid lineage (Nutt et al., 1999).

We confirmed Coda was not simply memorizing the profile of the training cell line (GM12878) and copying it to the test cell lines by examining differential regions, called by DESeq (Anders and Huber, 2010), between GM12878 and the other cell lines (Kasowski et al., 2013). Coda improved correlation and peak-calling even in those regions (Table 1). Similarly, it also improved correlation on the regions of the genome with enriched signal, i.e., called as statistically significant peaks (Table 2).

Table 1: Denoising differential regions (diff. reg.) between test cell line GM18526 and training cell line GM12878. Performance reported is improvement of the denoised model over baseline (original, subsampled reads) on the test cell line. In parentheses we report the baseline results followed by the denoised results. Peak-calling results on H3K27me3 are omitted due to the lack of peak calls in differential regions; all results on H3K36me3 are omitted due to low number of differential regions.

<table>
<thead>
<tr>
<th>Region</th>
<th>MISE (diff. reg.)</th>
<th>Pearson R (diff. reg.)</th>
<th>AUPRC (diff. reg.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H3K4me1</td>
<td>$-85%$ (4.01, 0.67)</td>
<td>$+59%$ (0.37, 0.59)</td>
<td>$+63%$ (0.93, 0.57)</td>
</tr>
<tr>
<td>H3K4me3</td>
<td>$-75%$ (2.88, 0.70)</td>
<td>$+14%$ (0.63, 0.72)</td>
<td>$+11%$ (0.76, 0.87)</td>
</tr>
<tr>
<td>H3K27ac</td>
<td>$-86%$ (1.43, 0.48)</td>
<td>$+39%$ (0.55, 0.77)</td>
<td>$+6%$ (0.90, 0.96)</td>
</tr>
<tr>
<td>H3K27me3</td>
<td>$-80%$ (0.78, 0.15)</td>
<td>$+106%$ (0.14, 0.30)</td>
<td></td>
</tr>
</tbody>
</table>

Training and testing on different cell types across different individuals

We next assessed if Coda could be trained on one cell type in one individual and used to denoise low-sequencing-depth data from a different cell type in a different individual. As above, the model was trained to output high-depth data (30M reads) from low-depth data (1M reads). We used histone ChIP-seq data spanning T-cells (E037), monocytes (E029), mesenchymal stem cells (MSCs, E026), and fibroblasts (E056) from the Roadmap Epigenomics Consortium (Consortium et al., 2015). Coda substantially improved the quality of the low-depth signal on the test cell type for all pairs of cell types (Table 3), illustrating its ability to denoise low-depth data on a cell type even if high-depth training data for that cell type is not available.

The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC 4.0 International license.

bioRxiv preprint doi: https://doi.org/10.1101/052118; this version posted January 27, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC 4.0 International license.
we report the baseline results followed by the denoised results. Over baseline (original, subsampled reads) on the test cell line. In parentheses cell line GM12878. Performance reported is improvement of the denoised model

Table 2. Denoising peak regions between test cell line GM18526 and training regions, with similar (very slightly better) MSE and correlation across the achieved 3x lower MSE on peak regions and 2x lower MSE on differential useful in genome-wide prediction.

multiple marks limits the complexity of models that would be practically
to be predicted, binned into 25bp bins). As noted above, the desire
denoising and peak calling, respectively. In both cases, we used an input
We compared Coda to a linear and logistic regression baseline for signal
Coda outperforms linear baselines

Fig. 3. Coda removes noise from low-sequencing-depth experiments on lymphoblastoid cell lines derived from different individuals. (A) Compared to the signal from subsampled reads (blue), the denoised signal (green) shows greater correlation with the full signal (left) and more accurate peak-calling (right) across all cell lines. The model was trained on GM12878 and tested on different cell lines; within each column in the plot, each point is a single test cell line. (B) With 1M reads per mark, the denoised H3K27ac data is equivalent in quality to a dataset with 15M+ reads per mark, and the H3K36me3 data is equivalent to a dataset with 21M+ reads per mark. Similar results hold for other marks. These results are from training on GM12878 and testing on GM18526.

Table 3. Cross cell-type experiments. Rows are train cell type, while columns are

Coda outperforms linear baselines

We compared Coda to a linear and logistic regression baseline for signal
denoising and peak calling, respectively. In both cases, we used an input
region of the same size as Coda (i.e., 25,025bp centered on the location
to be predicted, binned into 25bp bins). As noted above, the desire for computational efficiency in making genome-wide predictions across multiple marks limits the complexity of models that would be practically useful in genome-wide prediction.

When evaluated in the same cell type, different individual setting, Coda achieved 3x lower MSE on peak regions and 2x lower MSE on differential regions, with similar (very slightly better) MSE and correlation across the whole genome. This implies that Coda is better able to learn to match the exact values of the signal tracks on “difficult” regions (i.e., where there is the greatest deviation from the training signal), even though the linear model matches the rough shape. These regions are important to predict well because they can give insight into the differences between individuals and cell types.

We note that many forms of smoothing can be represented via linear regression. For example, a standard Gaussian filter can be interpreted as taking a linear combination of surrounding points with fixed coefficients. The comparison against a linear regression baseline therefore sets an upper
bound for the performance of simple smoothing measures on this task (assuming no overfitting, which we do not observe in our case).

**Comparisons to denoising and imputation**

Next, we studied Coda's performance in two additional settings: pure denoising (using the noisy target mark as the only input mark) and imputation from noise (using all noisy histone marks but the target mark as the input marks). This is in contrast to the standard setting described above, where we use all noisy histone marks, including the noisy version of the target mark, to recover a high-quality version of the target mark.

In the denoising case, Pearson correlation dropped by 0.03 points and AUPRC dropped by 0.05, on average, compared to when all marks were used as input. Thus, additional marks provided some information, but the denoised signal was still substantially better than the original subsampled signal.

In the imputation case, performance dropped somewhat on the narrow marks (H3K4me1, H3K4me3, H3K27ac, −0.12 correlation, −0.13 AUPRC) and dropped more on the broad marks (H3K27me3, H3K36me3, −0.29 correlation, −0.30 AUPRC). The gap in correlation was even larger within peak regions. Thus, having a noisy version of the target mark substantially boosts recovery of the high-quality signal.

**Removing noise from low cell input**

Conventional ChIP-seq protocols require a large number of cells to reach the necessary sequencing depth and library complexity (Brind’Amour et al., 2015; Cao et al., 2015), precluding profiling when input material is limited. Several ChIP-seq protocols were recently developed to address this problem. We studied ULI-NChIP-seq (Brind’Amour et al., 2015) and MOWChIP-seq (Cao et al., 2015), which use low cell input (10^2-10^3 cells) to generate signal that is highly correlated, when averaged over bins of size 2-4kb, with experiments with high cell input. However, at a finer scale of 25bp, the low-input signals from both protocols are poorly correlated with the high-input signals (Table 4).

We thus used Coda to recover high-resolution, high-cell-input signal from low-cell-input signal specific to each protocol. For ULI-NChIP-seq, we used a single mouse embryonic stem cell dataset (Brind’Amour et al., 2015). For MOWChIP-seq, we trained on data from the human LCL, GM12878 and tested on hematopoietic stem and progenitor cells (HSPCs) from mouse fetal liver (Cao et al., 2015). Coda successfully denoised the low-input signal from both protocols (Table 4). Fig. 5 illustrates our model denoising MOWChIP-seq signal across the Runx1 gene, a key regulator of HSPCs (North et al., 2002); the results of peak calling were too noisy, even on the original 10,000-cell data, to allow for any qualitative judgment of improvement.

**Removal of noise from low-enrichment ChIP-seq**

Histone ChIP-seq experiments use antibodies to enrich for genomic regions associated with the target histone mark. When an antibody with low specificity or sensitivity for the target is used, the resulting ChIP-seq data will be poorly enriched for the target mark. This is a major source of noise (Landt et al., 2012). We simulated results from low-enrichment experiments by corrupting GM12878 and GM18526 LCL data (Kasowski et al., 2013). For each histone mark profiled in those cell lines, we kept only 10% of the actual reads and replaced the other 90% with reads taken from the control ChIP-seq experiment, which was done without the use of any antibody; this simulates an antibody with very low specificity.

This corruption process significantly degraded the genome-wide Pearson correlation and the accuracy of peak calling (Table 5). This shows that recovering the true signal from the corrupted data cannot be achieved by simply linearly scaling the signal (e.g., multiplying the empirical fold enrichment by 10 since only 10% of the actual reads were kept), as if that were the case, the correlation would be unchanged. In contrast, when trained on GM12878 and tested on GM18526, Coda accurately recovered high-quality, uncorrupted signal from the corrupted data (Table 5). Fig. 6 shows a comparison of Coda’s output versus the corrupted and uncorrupted data at the promoter of the EBF1 gene, another key transcription factor of the B-lymphoid lineage. (Nechankir et al., 2013).

To further validate Coda’s output, we examined aggregate histone ChIP-seq signal around known biological regions of interest. In particular, we used the fact that H3K4me1 and H3K27ac, known enhancer marks, are enriched at DNase 1 hypersensitivity sites (DHSs), whereas H3K36me3

---

**Table 4. Low-cell-input experiments.** We report improvement of the denoised model output over baseline (original low-input experiments), as compared to high-input experiments. In parentheses we report the baseline results followed by the denoised output over baseline (original low-input experiments), as compared to high-input experiments.

<table>
<thead>
<tr>
<th>Mark</th>
<th>MSE Improvement</th>
<th>Pearson R Improvement</th>
<th>AUPRC Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>H3K4me3</td>
<td>−61% (&lt;1.19, 0.54)</td>
<td>+208% (0.13, 0.41)</td>
<td>+61% (0.24, 0.38)</td>
</tr>
<tr>
<td>H3K9me3</td>
<td>−46% (0.51, 0.27)</td>
<td>+28% (0.41, 0.53)</td>
<td>+32% (0.28, 0.36)</td>
</tr>
<tr>
<td>H3K27me3</td>
<td>−41% (0.68, 0.40)</td>
<td>+57% (0.34, 0.54)</td>
<td>+52% (0.34, 0.45)</td>
</tr>
<tr>
<td>H3K4me3</td>
<td>−42% (1.18, 0.68)</td>
<td>+122% (0.14, 0.31)</td>
<td>+34% (0.19, 0.25)</td>
</tr>
<tr>
<td>H3K27me3</td>
<td>−21% (1.44, 1.14)</td>
<td>+159% (0.09, 0.24)</td>
<td>+66% (0.15, 0.24)</td>
</tr>
</tbody>
</table>

---

**Fig. 5. Genome browser tracks for low-cell-input experiments.** We compare noisy signal obtained from 100 cells (top) with Coda’s output (middle) and the target, high-quality signal obtained from 10,000 cells (bottom) at the Runx1 gene in mouse hematopoietic stem and progenitor cells. The model was trained on MOWChIP-seq data generated from human LCL (GM12878) and captured two strong peaks at the promoters of the two isoforms, removing much of the intervening noise. The signal tracks are in arcsinh units, with a scale of 0-40 for both histone marks.
4 Conclusion

We describe a convolutional denoising algorithm, Coda, that uses paired noisy and high-quality samples to substantially improve the quality of new, noisy ChIP-seq data. Our approach transfers information from generative noise processes (e.g., mixing in control reads to simulate low-enrichment, or performing low-input experiments) to a flexible discriminative model that can be used to denoise new data. We believe that a similar approach can be used in other biological assays, e.g., ATAC-seq and DNase-seq (Buenrostro et al., 2013; Crawford et al., 2006), where it is near impossible to analytically characterize all types of technical noise or the overall data distribution but possible to generate noisy versions of high-quality samples through experimental or computational perturbation. This can significantly reduce cost while maintaining or even improving quality, especially in high-throughput settings or when dealing with limited amounts of input material (e.g., in clinical studies).

An important caveat to our work is that the performance of Coda depends strongly on the similarity of the noise distributions and underlying material (e.g., in clinical studies).
models to data that is very different from what it was trained on is unlikely to work. We also assume that the noise parameters in the test data are known in advance, e.g., the sequencing depth, the number of input cells, or the level of ChIP enrichment. In some cases (e.g., the low-sequencing-depth and low-cell-input settings) this is true, but in others (e.g., the low-enrichment setting) it is not always possible. An important direction for future work is to make Coda more robust, for example, training a single model over various settings of the noise parameters and various cell types could improve the generalizability of the models.

To further improve performance, more complex neural network architectures could also be explored: for example, using recurrent neural networks (Sutskever et al., 2014) to explicitly model long-range spatial correlations in the genome; multi-tasking across output marks instead of training separate models for each mark; or using deeper networks.

Another avenue for future work is exploring using more than just the noisy histone ChIP-seq data at test time. In this work, we use only the noisy data at test time, training our models to transform it into high-quality data. In reality, at test time we might have access to other data; for example, we might also have the DNA sequence of the test sample or access to high-quality ChIP-seq data on a closely related cell type. Other work has used DNA sequence to predict transcription factor binding (Alipanahi et al., 2015; Zhou and Troyanskaya, 2015), chromatin accessibility (Kelley et al., 2015), and methylation status (Angermueller et al., 2016a). A natural next step would be to combine the ideas from these methods with ours, e.g., by having a separate convolutional module in our neural network that incorporates sequence information and joins with the ChIP-seq module at an intermediate layer. Others have also used high-quality ChIP-seq data from closely related cell types for imputation (Ernst and Kellis, 2015), combining this with our denoising approach could help to avoid a potential pitfall of imputation approaches, namely the loss of cell-type-specific signal, while improving the accuracy of our denoised output.

Below, we provide a link to a script that trains a model for low-sequencing-depth noise using the LCL data described above. Since the type of noise can vary from context to context, we also provide the code for the general Coda framework to allow for developers of new protocols (e.g., new low-cell-count techniques) or core facilities that have high throughput to train Coda with data specific to their context.

Data Availability and Processing

Datasets

We used the following publicly-available GEO datasets in this work:

1. GSE50893 for ChIP-seq data on LCLs (Kasowski et al., 2013)
2. GSE63323 for ULI-NCHIP-seq data (Brind’Amour et al., 2015)
3. GSE65516 for MOWCHIP-seq data (Cao et al., 2015)
4. GSM376620 for DNase hypersensitive peaks (Bernstein et al., 2012)

For the low-sequencing-depth experiments, the full depth for GM12878 (training set) was 171M (million reads) for H3K4me1, 168M for H3K4me3, 328M for H3K27ac, 265M for H3K27me3, and 123M for H3K36me3. The full depth for GM18526 (test set) was 120M for H3K4me1, 136M for H3K4me3, 125M for H3K27ac, 138M for H3K27me3, and 223M for H3K36me3.

For the cross-cell-type experiments, we used the consolidated Roadmap Epigenomics data (Consortium et al., 2015), which is publicly available from http://gig2.wustl.edu/roadmap/data/byFileType/alignments/. Each mark is downsampled to a maximum of 30M reads to maximize consistency across marks; we used this as the full depth data, and downsampled to 1M reads for the noisy data. A detailed description of this dataset is available in (Roa, 2015).

Dataset preparation

Fold change signal profiles and peak calling.

For each experiment, we used align2rawsignal (Kundaje, 2013) to generate signal tracks and MACS2 (Feng et al., 2012) to call peaks, as implemented in the AQUAS package (Lee and Kundaje, 2016). For the signal track, we used fold change relative to the expected uniform distribution of reads after an inverse hyperbolic sine transformation (Hoffman et al., 2012). We used the gappedPeaks output from MACS2 as the peak calls. For computational efficiency, we binned the genome into 250bp segments, averaging the signal in each segment.

We evaluated our peak calling on a bin-by-bin basis, i.e., our model output one number for each bin representing the probability that that bin was a true peak, and we treated each bin as a separate example for the purposes of computing AUROC, our metric for peak calling performance. To get ground truth data for our peak calling tasks, we labeled each bin as “peak” or “non-peak” based on whether that bin was part of a peak called by MACS2 on the high-quality data.

Computing AUROC requires predictions to be ranked in order of confidence. For our model, we used the output probabilities for each bin to calculate the ranking. MACS2 outputs both a peak p-value track, assigning a p-value to each genomic coordinate, and a set of binary peak calls. To measure baseline performance on the noisy data, we ranked each bin by the maximum peak p-value assigned by MACS2 to a genomic coordinate in that bin, unless that bin did not intersect with any of the binary peak calls, in which case it was assigned a p-value of $\infty$ (i.e., ranked last). We did this to ensure that the high-quality peak track had an AUROC of 1; empirically, this also improved performance of the noisy MACS2 baseline.

Histone marks used

We used different sets of input and output histone marks for different experiments depending on which marks each dataset provided. For the same cell type, different individual experiments (using lymphoblastoid cell lines), we trained and tested on H3K4me1, H3K4me3, H3K27ac, H3K27me3, and H3K36me3; we used the same data for the low-ChIP-enrichment experiments. For the different cell type, different individual experiments (using the uniformly-processed Roadmap Epigenomics Consortium datasets (Consortium et al., 2015)), we trained and tested on H3K4me1, H3K4me3, H3K9me3, H3K27ac, H3K27me3, and H3K36me3. For all of the above experiments, we also used data from the control experiments (no antibody) as input. Lastly, for the low-cell-input experiments, we used H3K4me3, H3K9me3, and H3K27me3 from the ULI-NCHIP-seq dataset and H3K4me3 and H3K27ac from the MOWCHIP-seq dataset.

Low-cell-input datasets

The ULI-NCHIP-seq (Brind’Amour et al., 2015) and MOWCHIP-seq (Cao et al., 2015) papers provided several datasets corresponding to different numbers of input cells used. For each protocol, we used the datasets with the lowest number of input cells as the noisy input data (ULI-NCHIP-seq: 10⁵ cells for H3K9me3 and H3K27me3, 5x10⁴ cells for H3K4me3; MOWCHIP-seq: 10² cells) and the datasets with the highest number of input cells as the gold-standard, high-quality data (ULI-NCHIP-seq: 10⁶ cells for H3K9me3, 10⁵ cells for H3K4me3 and H3K27me3, MOWCHIP-seq: 10⁴ cells). The ULI-NCHIP-seq data had matching low- and high-input experiments only for a single cell type, so we divided it into chr5-19 for training, chr3-4 for validation, and chr1-2 for testing.

Code, data, and browser track availability

Our code is available on Github at https://github.com/kundajelab/coda, including a script that downloads pre-processed
Acknowledgements

We thank Jun-Wook Lee for his assistance with the AQUAS pipeline and Kyle Loh, Lincoln Kaplow, and Nasa Sinnott-Armstrong for their helpful feedback and suggestions.

Funding

EP acknowledges support from a Hertz Fellowship and an NDSEG Fellowship. This work was also supported by NIH grants DP2-GM-123485 and 1R01ES025009-01.

References


Mechanizky, R., Abbas, D., Scherer, S. et al. (2013) Transcription factor EBF1 is essential for the maintenance of B cell identity and prevents of alternative fates in committed cells. Nature Immunology, 14, 867–875. URL http://www.nature.com/doifinder/10.1038/ni.2641


