SHORT REPORT

Bedshift: perturbation of genomic interval sets

Aaron Gu[1,5], Hyun Jae Cho[1,5], and Nathan C. Sheffield[1,2,3,4,✉]

[1]Center for Public Health Genomics, University of Virginia
[2]Department of Public Health Sciences, University of Virginia
[3]Department of Biomedical Engineering, University of Virginia
[4]Department of Biochemistry and Molecular Genetics, University of Virginia
[5]Department of Computer Science, University of Virginia School of Engineering
✉ Correspondence: nsheffield@virginia.edu

Results of functional genomics experiments such as ChIP-Seq or ATAC-Seq produce data summarized as a region set. Many tools have been developed to analyze region sets, including computing similarity metrics to compare them. However, there is no way to objectively evaluate the effectiveness of region set similarity metrics. In this paper we present *bedshift*, a command-line tool and Python API to generate new BED files by making random perturbations to an original BED file. Perturbed files have known similarity to the original file and are therefore useful to benchmark similarity metrics. To demonstrate, we used bedshift to create an evaluation dataset of 3,600 perturbed files generated by shifting, adding, and dropping regions from a reference BED file. Then, we compared four similarity metrics: Jaccard score, coverage score, Euclidean distance, and cosine similarity. The results show that the Jaccard score is most sensitive to detecting adding and dropping regions, while the coverage score is more sensitive to shifted regions.
Availability: BSD2-licensed source code and documentation can be found at `https://bedshift.databio.org`.

Introduction

In the past few years, projects such as ENCODE (Encyclopedia of DNA Elements) and IHEC (International Human Epigenome Consortium) have established large catalogs of genomic features, including regulatory regions, transcription factor binding sites, and SNPs (Dozmorov, 2017). These data can be summarized into region sets, often stored in BED file format containing genomic regions represented by a chromosome number, start position, stop position, and optional metadata. Increasingly, computational tools are being developed to produce and consume BED files (Zhou *et al.*, 2020). Region sets are of particular interest in epigenomics, where hundreds of thousands of cell-type specific elements have been shown to play an important part in gene regulation (Song *et al.*, 2011; Thurman *et al.*, 2012).

Among the many applications of region sets, interest has grown in methods to compare region sets with one another (Kanduri *et al.*, 2018). New genomic regions produced from experiments can be associated with established genomic regions using co-occurrence, under the assumption that region sets with many overlapping regions may reflect biological relationships (Dozmorov, 2017). There are many tools that evaluate the similarity of two region sets. One general tool that provides the user with multiple results is the GSuite Hyperbrowser, which includes the most similar region sets, unique region sets, and how the co-occurrence counts change along the genome (Simovski *et al.*, 2018). Some tools use a statistical test to measure the significance of the co-occurrence. For example, GenomeRunner (Dozmorov *et al.*, 2016), LOLA (Sheffield and Bock, 2016; Nagraj *et al.*, 2018), GIGGLE (Layer *et al.*, 2018), and IGD (Feng and Sheffield, 2020) take BED files as input and compute region overlap counts, followed by a Fisher's exact test to produce a similarity score or ranking of most similar files. Other tools, such as regioneR and ChIP-Seeker, use permutation or sampling of regions or random background region set to calculates the probability of observing more extreme overlap between it and the provided data (Yu *et al.*, 2015; Gel *et al.*, 2015). Another approach is to examine distribution along the genome such as the approach taken by GenometriCorr (Favorov *et al.*, 2012). All of these tools allow users to find novel relationships between their data and reference data sets (Kanduri *et al.*, 2018).

As more tools are being developed in this field, it is important to understand how one improves upon another, and when it is better to use one tool over another. One approach is to simply run several different analysis tools at once, which is facilitated by Coloc-stats (Simovski *et al.*, 2018). This approach shows that the region set similarities vary by tool. Considering multiple metrics can give a more complete picture for a particular question, but it does not objectively compare the differences.

Here, we address this issue by simulating perturbations of region sets, allowing us to construct ground truth results between two region sets. We introduce bedshift, a command line interface and Python package that provides users the ability to create new BED files based on random modifications to an original BED file. A user can specify what percentage of regions they want to shift, drop, add, cut, and merge. Then, the user measures similarity between the original and perturbed BED files to see if the similarity scores match the user-specified change rate. The goal of bedshift is to provide a uniform and reliable way to evaluate region set similarity metrics. To demonstrate how to use bedshift, we created an evaluation set of 3,600 files with controlled levels of divergence to an original file. We then compared different similarity metrics to see how scores vary as the level of perturbation is increased.

Results

Overview of bedshift

Bedshift is available as a command line interface as well as a Python package. Bedshift perturbs regions in a region set with 5 possible operations: shift, add, cut, merge, and drop. The operations can be specified one at a time or all in one command, in which case bedshift will run them in the order of shift, add, cut, merge, and then drop. The number of regions perturbed is set as a proportion of the total number of regions in the region set.

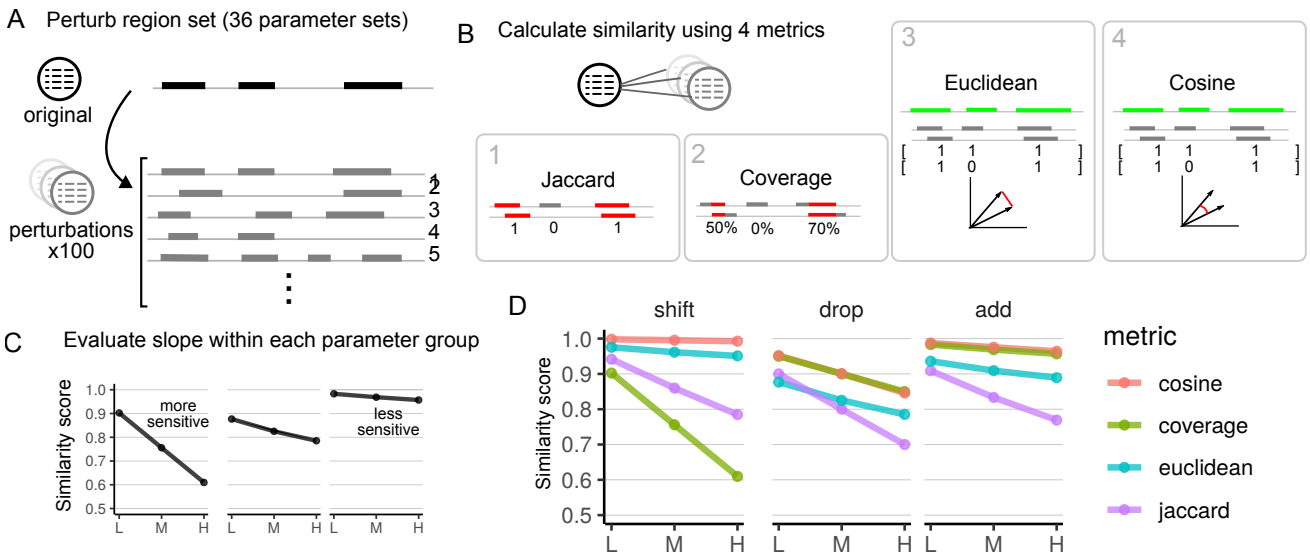

Figure 1: ***Overview of simulation study and comparison of four similarity metrics.*** *A) One BED file was used to create 3600 perturbed files, 100 repetitions for each of 36 different combinations of add, drop, and shift perturbations. B) We calculated four similarity metrics between the original file and the perturbed files. C) Within each parameter group, as the perturbation increases, the similarity score decreases. The slope of decrease is used to measure the sensitivity of the metric to the perturbation. D) Results for shift, drop, and add-only perurbations. L-Low; M-Moderate; H-High perturbation.*

For example, if given a BED file with 1000 regions, the operation `bedshift -b example.bed -a 0.2 -s 0.4` would add 200 new regions (20%) and shift 400 of the regions (40%).

The shift operation will shift the start and end position of a region by a random value based on a normal distribution specified by the user. The add operation will create randomly generated regions on any chromosome with a length based on a normal distribution specified by the user. The drop operation will randomly delete regions from the region set. The cut operation will split a region into two new regions, with the split position in the region being randomly determined. Finally, the merge operation will merge two adjacent regions, potentially creating very large regions.

Simulation study and evaluation approach

To test bedshift and demonstrate how it can be used to objectively evaluate similarity measures of region sets, we created an evaluation set of perturbed BED files. We randomly selected one file from publicly available data from ENCODE (ENCODE Consortium, 2012). Then, we generated 3,600 BED files made from 36 different bedshift parameter sets (Supplemental Materials Table S1), such as adding and shifting, only adding, or dropping and shifting, repeating each combination 100 times (Figure 1A). These parameters were chosen arbitrarily to be not too small as to not have significant change, and not too large as to be completely different.

After simulating perturbed region sets to known parameters, we sought to evaluate the performance of different measures of region set similarity. For each pairwise comparison of original to perturbed BED file, we computed four similarity metrics: Jaccard score, coverage score, Euclidean distance, and cosine similarity (Figure 1B). The Jaccard score and coverage metrics were chosen based on their common usage in other similarity scoring methods (Kanduri *et al.*, 2018). The Euclidean distance and cosine similarity metrics are more exploratory. The results for each metric and parameter set is shown in Table S1. We also computed the Fisher's Exact test statistic used in LOLA and GIGGLE (Supplemental Materials Figure S1), which we analyzed separately due to its odds ratio not being on a 0 to 1 scale like the other similarity scores, as well as its insensitivity to dropping and adding (see Methods). To evaluate the metrics, we compared each measure for its ability to reflect bedshift perturbations by calculating the slope change of the metric against the scale of the perturbation (Figure 1C). We began by evaluating each perturbation type separately, and then evaluated the results of pairwise combinations where we perturbed each region file in two ways.

Evaluating individual perturbations

Shift. The perturbation with the least consistence across measures was *shift*. The most sensitive metric was coverage (Figure 1D). This is expected, since coverage measures the percentage of individual regions that overlap with other regions, and is therefore affected by any shift; in contrast the other measures are based only on overlap counts, which will only change if the shift is substantial enough to eliminate overlap. We found that the Jaccard score had intermediate sensitivity, and Euclidean distance and cosine similarity were least affected by shifts. This is likely due to the universe we used in constructing the vectors, which contained regions with 537 bp median length compared to 320 bp median length from the original dataset. Consequently, more regions in the original BED file would overlap with universe regions, even after shifting. It follows that the vector representation of the region, used in Euclidean distance and cosine similarity, would remain the same.

Drop. Compared to shifting, the metrics were more similarly sensitive to the *drop* perturbation (Figure 1D). The Jaccard score had the lowest similarity scores overall, as well as the steepest decrease across the increasing levels of perturbation, indicating the highest sensitivity. Other metrics were almost equally as sensitive.

Add. For the *add* perturbation, the Jaccard score again had the lowest similarity scores and steepest decline (Figure 1D). Interestingly, we found that all metrics were less sensitive to adding regions than dropping regions, and this is particularly true for the vector-based Euclidean distance and cosine similarity. We expected sensitivity to adding to resemble sensitivity to dropping,

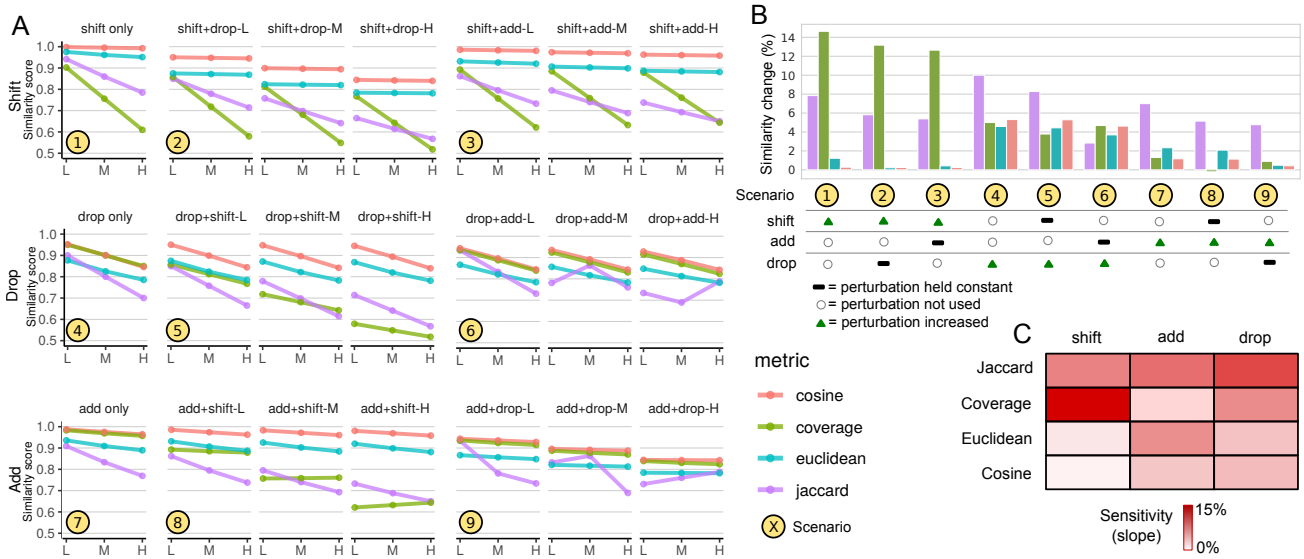

Figure 2: ***Bedshift and similarity score results****. A) Similarity scores for shift, drop, and add perturbations, along with their pairwise combinations. B) Summary of similarity score change for different perturbation levels measured. For a given scenario, each perturbation is classified as either "increasing", "not used", or "held constant". The slopes are averaged across all three levels held constant. C) Summary of the sensitivity of each metric to the three different perturbations.*

because the perturbations are complementary. We explore this further in the next section.

Sensitivity of dropping vs adding. We were initially surprised that all metrics were more sensitive to dropping than adding. Employing a hypothetical overlap counting example, we can see why that happens for the Jaccard score: Given a set of 4 regions, if a non-overlapping region is added, the score would decrease by 25% to $\frac{3}{4}$. On the other hand, if a region is dropped, then the score decreases by 33% to $\frac{2}{3}$. This provides a theoretical explanation of the observation that the Jaccard score is more sensitive to drop than add. Remarkably, our perturbation results were able to capture this nuance. Similarly, the coverage score is less sensitive to add than drop, which can be explained similarly: adding a region would decrease the score by 12.5% to $(\frac{3}{4} + \frac{3}{3})/2$, and dropping a region would decrease the score by 16.7% to $(\frac{2}{3} + \frac{2}{2})/2$. Thus, dropping regions still decreases the coverage score more, causing a larger slope.

Evaluating combinatorial perturbations

To extend our results, we next examined the sensitivity, as measured by slope, when combinations of perturbations are used. We used bedshift to create perturbed files with each pairwise combination of parameters, at each level (Table S1). This resulted in 36 parameter sets: 9 represent the 3 individual perturbations at 3 levels each discussed previously, and then 27 sets represent each pairwise combination of 3 perturbations at 3 levels. We grouped these results into 9 scenarios, 3 of which represent individual perturbations and 6 of which represent each pairwise comparison of perturbation (Figure 2A). For example, in *Scenario 2*, we plotted the slope of the lines as the *add* perturbation is increased, with the *drop* perturbation held constant. For each pairwise scenario, we used 3 different levels for the perturbation held constant (high, moderate, and low), resulting in 3 plots per pairwise scenario. To summarize these results, we computed the slope of the lines, which can be thought of as a sensitivity score measuring how much a metric changes with a fixed increase in a particular perturbation (Figure 2B), and then summarized the slopes by computing the average across all analyses (Figure 2C).

Pairwise perturbations reduce sensitivity to shifts. In most cases, the pairwise perturbations showed similar patterns to the individual perturbations. For example, coverage has a by far the highest slope, over 12%, for all *shift* scenarios (Scenarios 1-3). The combinations in scenarios 2 and 3 show that adding in other perturbations lowers the sensitivity. In contrast, the vector similarity metrics show very little sensitivity to shift, whether or not other perturbations are included.

Jaccard score is quickly saturated with add and drop. The Jaccard score is the most sensitive to the *drop* perturbation, as shown in Scenario 4. Combining *drop* with *shift* (Scenario 5) reduced the sensitivity, which we attribute to diluted signal. When combined with *add*, we observed a very significant decrease in sensitivity; in fact, the combination of drop and add leads to a *positive* slope in score as *add* is increased when *drop* is held at a constant high level (Scenarios 6 and 9). This can be explained by the interaction between *add* and *drop* – once enough regions have been *dropped*, then adding has an increased probability to add back a region that was previously dropped, causing a canceling effect. This causes the similarity score to actually increase as the perturbation increases, against intuition. Thus, Scenario 6 and 9 demonstrate that caution must be used as the level of perturbation increases; in this case, the moderate and high levels of dropping have saturated the sensitivity of the Jaccard score, causing increased adding to lead to increased or inconsistent similarity.

Coverage score is saturated by shift and add. In Scenario 8, the coverage score similarity turns negative (*i.e.* the slope becomes positive). This means that the files are becoming more similar as regions are added in the presence of shifted regions. This affect is similar to the Jaccard score inconsistency in presence of both drop and add; when the shift perturbation is strong enough, then adding regions begins to have a higher probability of adding back regions that have been shifted away (Scenario 8).

Conclusion

In this paper we present bedshift, a new tool to help researchers evaluate the effectiveness of region set similarity metrics. Similarity scoring metrics and tools are becoming increasingly common, and it is important to know how each tool performs on different datasets. Bedshift is a way to generate new BED files with perturbations such as shifted regions, added regions, dropped regions, and more.

Our results provide an initial analysis to compare different similarity scoring metrics. We uncovered interesting information about the relative performance of these metrics. The key conclusion is that similarity scores have unique sensitivities to types of perturbation. One key caution is that a metric that is more sensitive will more quickly reach a saturation point; at this stage, the metric becomes unreliable. In our analysis, we did not identify a global "best" metric, but each metric is likely to be more appropriate depending on use case. For example, Euclidean distance and cosine similarity were generally the least sensitive overall and would therefore be more useful for measuring similarities between distant region sets. Overall, the Jaccard score seemed to be the most sensitive, and is therefore most useful for more similar region sets. It also showed consistency in the slope decrease across multiple levels of perturbation. The coverage score would be most appropriate for detecting slight shifts, which may or may not be desirable. Discovering the performance of these different similarity metrics shows the advantage of bedshift, as it has allowed us to discover pros and cons of different similarity metrics.

Our analysis also leads to several directions for future work. First, it is possible that the universe-based measures would behave differently depending on the universe used to construct the vectors. More work needs to be done to explore optimal ways for constructing universes, which would benefit vector-based similarity metrics. Second, it will be interesting to explore the behavior of new similarity metrics, such as *seqpare*, which is based on mutual best matches (Feng *et al.*, 2020). Finally, similarity scoring methods could be combined for increased confidence in results. Bedshift will be a helpful tool going forward as we develop and evaluate new ways to measure similarity of region sets.

Acknowledgments

We would like to thank Erfaneh Gharavi and Guangtao Zheng for helpful comments. The authors declare no competing interests. This work was supported by NIGMS award GM128636 and the 4-VA program (NCS).

Methods

Jaccard score

The first metric was the Jaccard score based on overlapping regions between two BED files, computed by the formula

$$\frac{|A \cap B|}{|A \cup B|} \tag{1}$$

where A and B are the two BED files. Any region with at least 1 base pair overlap was counted in the total number of overlaps. Overlaps were computed using Augmented Interval List (AIList) (Feng *et al.*, 2019).

Coverage score

The BedTools coverage function was used, which takes in two BED files and uses the first one as the reference region set to determine coverage for each region in the second BED file (Quinlan and Hall, 2010). A normalization technique was applied to assign coverage scores to every region in both files. First, BedTools coverage was run with the perturbed file as the first argument and the original file as the second. Then the files were passed as arguments to the coverage tool in the opposite order, in order to account for coverage in regions across both files. This produced a coverage score between 0 and 1 for each region in both the original and the perturbed file. To get the final similarity score, the mean was taken of coverage values for every region in both files:

$$(\frac{\Sigma Coverage(A,B)_i}{|A|} + \frac{\Sigma Coverage(B,A)_i}{|B|})/2 \tag{2}$$

where A and B are the two BED files and $Coverage$ is the BedTools coverage score.

Euclidean distance

In vector-based similarity methods, a standard vocabulary was needed to represent each region as a position in the vector. To do this, a "vocabulary", or a universe, was constructed by merging regions from 300 BED files in the ENCODE dataset. Regions were merged if they were closer than 1000 base pairs. The resulting universe had 384,639 regions, with a median region length of 430 base pairs. This universe could then be represented by a vector of length 384,639. When casting new files into the universe, if a region in the new file overlapped with a universe region, then that index in the vector was set to 1. Therefore, each BED file was represented by a vector of 0's and 1's. The Euclidean distance is defined as

$$\sqrt{\Sigma(\mathbf{A}_i - \mathbf{B}_i)^2} \tag{3}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the two vectors representing the BED files. A normalized Euclidean distance was calculated by dividing by the maximum distance of two vectors (a vector of all 0's and a vector of all 1's), which was 518.02. That value was subtracted from 1, because a smaller normalized distance indicates a higher similarity.

Cosine similarity

The same vectorization technique and vectors used for the Euclidean distance metric were also used for the cosine similarity analysis. The cosine similarity is defined as

$$\frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{4}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the two vectors representing the BED files. In vector space, the closer two vectors are, the closer their cosine is to 0. Thus, the resulting cosine score was subtracted from 1 to get the final similarity score.

Slope of decrease

The slope change was measured by taking the difference between the highest and lowest score in the perturbation level (for example, the score difference between add 0.1 and add 0.3 using the Jaccard score was 0.15) and then dividing by 2 for the degrees of change

between the three levels. For groups of 9 perturbation parameters, such as increasing shift from 0.1 to 0.3 while holding add constant at 0.1, 0.2, and 0.3, the three slopes from the levels of add were averaged.

Fisher's Exact test

The contingency table for the Fisher's Exact test was constructed by using the overlap counts of the original and the perturbed BED files. When running the Fisher's Exact test on the contingency table, we found that the odds ratio and p-value could not be calculated when using add and drop perturbations. We suspect that adding and dropping regions causes one BED file to contain a subset of the regions in the other BED file. This would then cause one cell of the contingency table (regions in file X that are not in file Y) to be 0, making the test not be able to calculate the odds ratio and p-value. Therefore, we only use the shift perturbation in our analysis. The odds ratio can be interpreted as the odds of a region in region set Y that will appear in region set X is the odds ratio times as likely to do so as a region that is not in Y.

References

Dozmorov,M.G. (2017) Epigenomic annotation-based interpretation of genomic data: from enrichment analysis to machine learning. *Bioinformatics*, **33**, 3323–3330.

Dozmorov,M.G. *et al.* (2016) GenomeRunner web server: Regulatory similarity and differences define the functional impact of SNP sets. *Bioinformatics*, **32**, 2256–2263.

ENCODE Consortium (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature*, **489**, 57–74.

Favorov,A. *et al.* (2012) Exploring massive, genome scale datasets with the genometricorr package. *PLoS Computational Biology*, **8**.

Feng,J. *et al.* (2019) Augmented interval list: A novel data structure for efficient genomic interval search. *Bioinformatics*.

Feng,J. and Sheffield,N.C. (2020) IGD: High-performance search for large-scale genomic interval datasets. *bioRxiv*.

Feng,S.C. *et al.* (2020) Seqpare: A self-consistent metric of similarity between genomic interval sets. *F1000Research*, **9**, 581.

Gel,B. *et al.* (2015) regioneR: an R/Bioconductor package for the association analysis of genomic regions based on permutation tests. *Bioinformatics*, **32**, 289–291.

Kanduri,C. *et al.* (2018) Colocalization analyses of genomic elements: Approaches, recommendations and challenges. *Bioinformatics*, **35**, 1615–1624.

Layer,R.M. *et al.* (2018) GIGGLE: A search engine for large-scale integrated genome analysis. *Nature Methods*, **15**, 123–126.

Nagraj,V. *et al.* (2018) LOLAweb: A containerized web server for interactive genomic locus overlap enrichment analysis. *Nucleic Acids Research*.

Quinlan,A.R. and Hall,I.M. (2010) BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**, 841–842.

Sheffield,N.C. and Bock,C. (2016) LOLA: Enrichment analysis for genomic region sets and regulatory elements in R and bioconductor. *Bioinformatics*, **32**, 587–589.

Simovski,B. *et al.* (2018) Coloc-stats: A unified web interface to perform colocalization analysis of genomic features. *Nucleic Acids Research*, **46**, W186–W193.

Song,J. *et al.* (2011) Structure of dnmt1-dna complex reveals a role for autoinhibition in maintenance DNA methylation. *Science*, **331**, 1036–1040.

Thurman,R.E. *et al.* (2012) The accessible chromatin landscape of the human genome. *Nature*, **489**, 75–82.

Yu,G. *et al.* (2015) ChIPseeker: an R/Bioconductor package for ChIP peak annotation, comparison and visualization. *Bioinformatics*, **31**, 2382–2383.

Zhou,Y. *et al.* (2020) EpiCOLOC: Integrating large-scale and context-dependent epigenomics features for comprehensive colocalization analysis. *Frontiers in Genetics*, **11**, 53.

Supplemental Materials

parameter set	add	drop	shift	Jaccard mean	Coverage mean	Euclidean mean	Cosine mean
add1	0.1	0.0	0.0	0.909	0.981	0.939	0.988
add2	0.2	0.0	0.0	0.833	0.964	0.914	0.977
add3	0.3	0.0	0.0	0.769	0.951	0.895	0.966
drop1	0.0	0.1	0.0	0.900	0.950	0.883	0.954
drop2	0.0	0.2	0.0	0.800	0.900	0.834	0.905
drop3	0.0	0.3	0.0	0.700	0.850	0.796	0.852
shift1	0.0	0.0	0.2	0.941	0.902	0.979	0.998
shift2	0.0	0.0	0.5	0.860	0.756	0.966	0.996
shift3	0.0	0.0	0.8	0.785	0.610	0.957	0.994
add_drop1	0.1	0.1	0.0	0.942	0.933	0.874	0.946
add_drop2	0.1	0.2	0.0	0.840	0.886	0.831	0.901
add_drop3	0.1	0.3	0.0	0.737	0.838	0.795	0.852
add_drop4	0.2	0.1	0.0	0.783	0.920	0.865	0.939
add_drop5	0.2	0.2	0.0	0.878	0.886	0.827	0.898
add_drop6	0.2	0.3	0.0	0.772	0.828	0.795	0.852
add_drop7	0.3	0.1	0.0	0.736	0.910	0.857	0.932
add_drop8	0.3	0.2	0.0	0.693	0.867	0.824	0.894
add_drop9	0.3	0.3	0.0	0.807	0.828	0.795	0.851
shift_drop1	0.0	0.1	0.2	0.850	0.857	0.882	0.953
shift_drop2	0.0	0.1	0.5	0.779	0.718	0.879	0.950
shift_drop3	0.0	0.1	0.8	0.714	0.579	0.877	0.949
shift_drop4	0.0	0.2	0.2	0.758	0.812	0.833	0.904
shift_drop5	0.0	0.2	0.5	0.765	0.767	0.832	0.902
shift_drop6	0.0	0.2	0.8	0.642	0.548	0.830	0.900
shift_drop7	0.0	0.3	0.2	0.665	0.767	0.795	0.851
shift_drop8	0.0	0.3	0.5	0.615	0.643	0.794	0.849
shift_drop9	0.0	0.3	0.8	0.568	0.518	0.793	0.847
add_shift1	0.1	0.0	0.2	0.862	0.891	0.936	0.987
add_shift2	0.1	0.0	0.5	0.796	0.756	0.931	0.984
add_shift3	0.1	0.0	0.8	0.734	0.621	0.926	0.982
add_shift4	0.2	0.0	0.2	0.862	0.891	0.936	0.975
add_shift5	0.2	0.0	0.5	0.741	0.757	0.909	0.973
add_shift6	0.2	0.0	0.8	0.690	0.633	0.906	0.971
add_shift7	0.3	0.0	0.2	0.739	0.7874	0.894	0.965
add_shift8	0.3	0.0	0.5	0.695	0.760	0.891	0.963
add_shift9	0.3	0.0	0.8	0.652	0.645	0.889	0.961

Table S1: ***Parameter combinations used in the analysis and their results.***

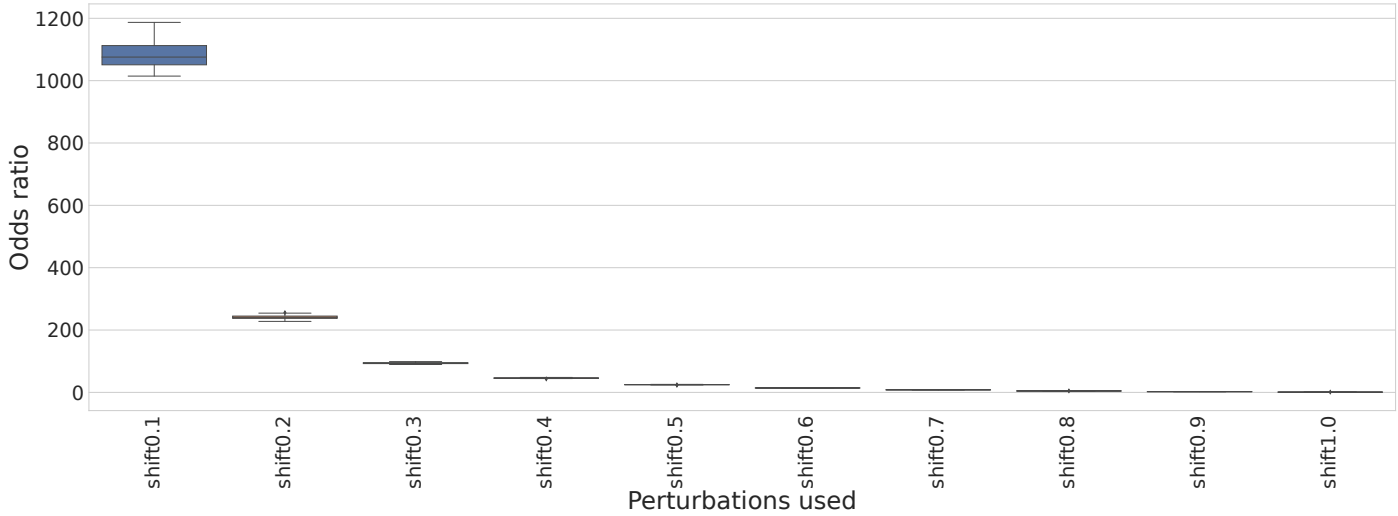

Figure S1: ***Fisher's Exact Test odds ratio.*** *Only the shift parameter was used because the Fisher's Exact Test does not measure adding and dropping regions well (see Methods). As the shift perturbation increases, the odds ratio decreases, meaning that it is less likely that a region in the original BED file will appear in the perturbed BED file.*