Supplementary Methods, Tables, and Figures

Wancen Mu[1], Eric Davis[2], Stuart Lee[3], Mikhail Dozmorov[4], Douglas H. Phanstiel[2], and Michael I. Love *[1,2]

[1]Department of Biostatistics, and
[2]Department of Genetics, University of North Carolina-Chapel Hill, NC 27599
[3]Department of Econometrics and Business Statistics, Monash University, Clayton, Australia
[4]Department of Biostatistics, Department of Pathology, Virginia Commonwealth University, Richmond, VA 23298, USA

May 20, 2022

1 Supplementary Methods

1.1 Previous methods

In order to generate background data sets for null hypothesis testing of association analysis, there are two ways to formulate. One method for generating background genomic features sets is to sample from a larger experimental pool or database, like LOLA (Sheffield and Bock, 2016) using Fisher's exact test, and Poly-Enrich (Lee *et al.*, 2020a) using likelihood ratio test based on negative binomial likelihood. Another method is to permute or shuffle the genomic ranges given random start sites to existing features, possibly considering an exclusion list of regions where features should not be located, like bedtools (Quinlan and Hall, 2010), ChIP-Enrich (Welch *et al.*, 2014), GenometriCorr (Favorov *et al.*, 2012), GAT additionally allowing controlling for GC content (Heger *et al.*, 2013), and one of strategies of regioneR implement a circular randomization to preserve inter-feature distance (Gel *et al.*, 2016).

1.2 Segmentation formulation

From annotation databases generated from Ensembl, we can obtain a sequence of gene density count per million base $\{X_1, ..., X_n\}$ positioned linearly. We assume there exist integers $\boldsymbol{\tau} = \boldsymbol{\tau^{(n)}} = (\tau_0, ..., \tau_U)$, where $0 = \tau_0 < \tau_1 < ... < \tau_U = n$, such that the collections of variables, $\boldsymbol{X}_{\tau_i}, ..., \boldsymbol{X}_{\tau_i+1}$, are separately stationary for each $i = 0, ..., U-1$. We let $n_i = \tau_i - \tau_{i-1}$ be the length of the ith region. Hence, we introduce the mapping

$$\pi : \{1, ..., n\} \rightarrow \{(i, j) : 1 \leq i \leq U, 1 \leq j \leq n_i\}$$

which relates the original sequence $\{X_1, ..., X_n\}$ to the segmentation sequence $\{X_{ij} : 1 \leq i \leq U, 1 \leq j \leq n_i\}$

1.3 ChromHMM annotations for Roadmap Epigenomics

[wancen: Should we talk about what is HMM and CBS?] Data were downloaded from `https://egg2.wustl.edu/roadmap/web_portal/` which include 15 small states and summeraize them into 3 big categories: low density("E9","E13","E14","E15"), middle density("E10","E11","E12"), high density("E1-E8"). [wancen: XX pieces left after merging states and mean width is]

*michaelisaiahlove@gmail.com

1.4 Shuffling

Genome shuffling was performed by random sample SNPs in acceptance region on Genome with probability proportional to SNPs count on each chromosome. The acceptance region exclude all ENCODE excludable regions plus telomere,centromere from UCSC and rCGH derived from `excluderanges` for hg38 (Dozmorov *et al.*, 2022).

1.5 Segmentation and block length chosen

On the issue of block length selection, we considered it in two ways. One is trying to find L_b that has the minimum value of a pseudometric $d^*(v) = |\sqrt{\frac{L_{v-1}}{L_v}} IQR(\mathcal{L}_{L_v}) - IQR(\mathcal{L}_{L_{v-1}})|$ where $\mathcal{L}_{L_v}$ is the statistic distribution at length L_v, $v = 1, 2, \cdots, V$, V is the number of candidate block length and $IQR(\mathcal{L})$ is the interquartile range of statistics distribution followed Bickel *et al.* (2010). Second way was evaluating conversing spatial distribution that generated null sets have similar properties with original sets, eg. inter-feature distance. We used the Earth's Mover distance(EMD) to quantify the similarity between the distributions of a inter-feature distance in the original and null dataset, resulting in values between zero (identical dis-tributions) to one (totally disjoint distributions). The Earth Mover's Distance (EMD) between two distributions is proportional to the minimum amount of work required to change one distribution into the other.

$$EMD(y, y') = \frac{min_{F=(f_{ij} \in F(y,y'))} WORK(F, y, y')}{min(w_{\sum}, u_{\sum})}$$

where y and y' is the histogram of original and null inter-feature distance with bin size = 0.3. Since the EMD always decreased as L_b increased because more neighbouring features reserved, the right L_b should be chosen so that longer length doesn't improve much as well as it is much smaller than the L_s.

1.6 Vector linear statistics and Gaussian approximation

Suppose we are interested in region overlap which is defined as one feature $\boldsymbol{x}$ $T_1, ..., T_\alpha$ with lengths $\tau_1, ..., \tau_\alpha$, and the feature $\boldsymbol{y}$ $S_1, ..., S_\beta$ with lengths $\rho_1, ..., \rho_\beta$, then the region overlap of feature $\boldsymbol{x}$ with feature $\boldsymbol{y}$ is defined as $Q \equiv \frac{1}{\alpha} \sum_{t=1}^{\alpha} V_t$ where $V_t = 1 - \prod_{k \in (T_t + \theta)} (1 - J_k)$ and $J_k = 1$ if position k belongs to feature B and 0 otherwise. θ is the shift among feature $\boldsymbol{x}$ and $\boldsymbol{y'}$, such as 1kb when linking promoters to gene. This statistic V_t is stationary except for end effects due to feature instances crossing segment boundaries. Here α can be derived as $\sum_{k=1}^{n} I_{k-1}(1 - I_k)$ where $I_k = 1$ if position k belongs to feature A and 0 otherwise. After we do the block bootstrap according to the segmentation region, we generate R new sets of null features $\boldsymbol{y'}$ and $R \times \frac{n}{L_b}$ blocks. Therefore, we can do both genome-wise or block-wise analysis based on the question being addressed. Either using $Q^{1'}, Q^{2'}, ..., Q^{R'}$ or $Q^{1'}, Q^{2'}, ..., Q^{R \times \frac{n}{L_b}'}$ to construct a normal distribution $\mathcal{Q}'$ according to the Central Limit Theorem. Notate, if block-wise analysis is preferred, the SD of bootstrap distribution should be scaled by $\sqrt{L_b}$.

Null hypothesis of no associations with 0.05 type I error:

$$Q \leq \mathcal{Q}'_{0.975}$$

So that the z score in Figure 2 is calculated by

$$z = \frac{\widehat{Q} - \widehat{Q^*}}{se_R(\widehat{Q})}$$

where $\widehat{Q(.)^*} = \frac{\sum_{r=1}^{R} \widehat{Q^*(r)}}{R}$ is the sample mean of the R replications and $se_R(\widehat{Q}) = \sqrt{\frac{\sum_{r=1}^{R} [\widehat{Q^*(r)} - \widehat{Q(.)^*}]^2}{R-1}}$.

1.7 Swaping algorithm

Algorithm 1: Block bootstrap GRanges across chromosome

```
Data: Feature GRanges, Block length($L_b$), bootstrap times(R), type('permute' or 'bootstrap')
Result: Bootstrapped distribution of test statistics
while $r \leq R$ do
    rearranges block: Generate consecutive tiling blocks with width = $L_b$ ; // where 'bait' blocks will be moved
    if permutation then
        random block: Sample blocks without replacement from rearranges block
    else if bootstrap then
        $n_b = \sum_{j=1}^{24} ceiling(L_c/L_b)$
        random block: Generate $n_b$ blocks with replacement and probability weights proportional to $L_c$ ; // these blocks are the 'bait' for capturing features in $\boldsymbol{y}$
    end
    Find overlap of random blocks and feature $\boldsymbol{y}$ ; // use the bait to sample features in $\boldsymbol{y}$
    Swap the ranges in those bait blocks given the shift between random block and rearranges block
end
Bind bootranges object which has $r$ and $L_b$ as metadata columns ← plyranges
```

Algorithm 2: Segmented block bootstrap with proportional block length

```
Data: Feature GRanges, Block length($L_b$), bootstrap times(R), segmentation GRanges
Result: Bootstrap distribution of test statistics
while $r \leq R$ do
    for each segmentation state $i$ do
        /* suppose $\alpha_i$ ranges in $\boldsymbol{y}$ of state $i$ */
        $L_s^i = \sum_{j=1}^{\alpha_i} L_j$ ; // $L_j$ is width of ranges
        $L_b^i = L_b * L_s^i/L_c$ ; // block length of state $i$
        $n_b^i = \sum_{j=1}^{\alpha_i} ceiling(L_j/L_b^i)$ ; // total # of blocks
        random block: Generate $n_b^i$ blocks start site with replacement and probability weights proportional to $L_j$ ; // these blocks are the 'bait' for capturing features in $\boldsymbol{y}$
        rearranges block: Generate $n_b^i$ tiling blocks start site ; // where 'bait' blocks will be moved
        Return: random and rearranges block start and chromosome name
    Construct random blocks Granges and find overlap of random blocks and feature $\boldsymbol{y}$ ; // use the bait to sample features in $\boldsymbol{y}$
    Swap the ranges in those bait blocks given the shift between random block and rearranges block
Bind bootranges object which has $r$ and $L_b$ as metadata columns ← plyranges
```

```
Algorithm 3: Segmented block bootstrap with fixed block length across chromosome
  Data: Feature GRanges, Block length($L_b$), bootstrap times(R), segmentation GRanges
  Result: Bootstrap distribution of test statistics
  while $r \leq R$ do
      $n_j = ceiling(L_j/L_b)$ ; // number of blocks within each ranges
      $n_b = \sum_{j=1}^{\alpha} n_j$; // suppose $\alpha$ ranges in feature $\boldsymbol{y}$
      random block: Generate $n_b$ blocks start site with replacement and probability weights
        proportional to $L_j$ ; // these blocks are the 'bait' for capturing features in $\boldsymbol{y}$
      rearranges block: Generate $n_b$ tiling blocks start site by order with width = $L_b$ ; // where
        'bait' blocks will move to
      for each segmentation state $i$ do
          Identify random blocks chosen that are in state $i$
          Identify rearranged blocks chosen that are in state $i$
          Return: random block start, random block chromosome name, rearranges block start,
            rearranges block chromosome name
      Construct random blocks Granges, concatenate rearranges start and chromosome name
        vector
      Find overlap of random blocks and feature $\boldsymbol{y}$ ; // use the bait to sample features in $\boldsymbol{y}$
      Swap the ranges in those bait blocks given the shift between random block and rearranges
        block
  Bind bootranges object which has $r$ and $L_b$ as metadata columns $\leftarrow$ plyranges
```

2 Supplementary Results

2.1 liver caQTL-GWAS colocalizations

1872 SNP data was download from the NHGRI-EBI GWAS catalog (Buniello *et al.*, 2018) on September 22, 2021, extracted only single variant associated with total cholesterol, and consensus peaks information of 20 samples were downloaded from GSE164870. Then caQTL genomic coordinates were converted from hg19 to hg38 using `liftOver` to construct 221,606 peaks GRanges. Through the below code, variance of statistics and the z score in Figure 2a,b was derived.

2.1.1 L_b selection

On the issue of block length selection, we considered it in two ways. One is trying to find L_b that has the minimum value of a pseudometric $d^*(v) = |\sqrt{\frac{L_{v-1}}{L_v}} IQR(\mathcal{L}_{L_v}) - IQR(\mathcal{L}_{L_{v-1}})|$ followed Bickel *et al.* (2010), where $\mathcal{L}_{L_v}$ is the statistic distribution at length L_v, $v = 1, 2, \cdots, V$ and $IQR(\mathcal{L})$ is the interquartile range. Figure S1A showed d^* were in common had smaller values when $L_b \in [300000, 800000]$. Another way was evaluating conversing spatial distribution by assuming generated null sets have similar properties with original sets, eg. inter-feature distance. The Earth Mover's Distance (EMD) was used to access two distribution similarity because it is proportional to the minimum amount of work required to change one distribution into the other. Since the EMD always decreased as L_b increased as more neighbouring features were reserved, the right L_b should be chosen that a longer length won't improve much as well as requesting L_b is much smaller than the L_s. So [200000,600000] was shown to be a good range by visualization and according to the Elbow Method of EMD(Figure S1B-C).

```
len <- do.call(c,lapply(boots, length))
Overlaps <-boots %>% join_overlap_left(y,maxgap=10e3) %>%
    group_by(iter) %>%
    summarize(rate = 1-sum(is.na(y.id))/len) %>%
    as.data.frame() %>%
    select(rateOverlaps) %>%
    summarise_all(funs(mean,var,IQR)) %>%
    mutate(rate.z = abs(obs.mean-rate.mean)/sqrt(rate.var))
```

2.2 macrophage cell lines

Macrophage processed 24 RNA-seq samples and 145 ATAC-seq samples were loaded from `fluentGenomics`(Lee *et al.*, 2020b). They are measured after interferon gamma(IFNg) stimulation. Since the transcriptomic response to IFNg stimulation may be mediated through an increasing transcription factors bindings on nearby regions and ATAC-seq can captioned those regions' accessibility, we expect there is an enrichment of differentially accessible (DA) ATAC-seq peaks in the vicinity of differentially expressed (DE) genes.

When performing block bootstrap with 5e5 block length on 100 times, we got $t_{99} = -108.1$ and p-value ¡.05. We could reject the null hypothesis and concluded that there was significant enrichment of DA peaks near DE genes by finding overlaps.

For fitting the generalized penalized splines, `gam` function in the `mgcv` package was used to fit the model, which used a penalized likelihood maximization, and generalized cross-validation is used to choose the optimal value for the smoothing parameter, λ. Then `tidymv` package was used to predict and extract the fitted value.

```
boot_stats <- x %>% join_overlap_inner(y') %>%
  group_by(id.x, iter) %>%
  summarize(count = n(), log2FC = max(log2FC)) %>%
  as.data.frame() %>%
  complete(id.x, iter, fill=list(count = 0)) %>%
  select(iter, count, logFC) %>%
  nest(-iter) %>%
  mutate(fit= map(data, ~gam(count ~ s(log2FC), data = .)),
         pred   = map(fit, ~predict_gam(model = .)),
         fitted = map(pred,~find_fit(data=.)))
```

2.3 Chromium Single Cell Multiome ATAC + Gene Expression assay

Data were downloaded according to Ricard Argelaguet and Marioni (2020) instruction which includes genes and peaks in 10,032 cells. Cell type annotations have been done a priori by the 10x Genomics R&D team. Then information on chromosome 1 to 22 were selected to construct gene and peak GRanges. Since the main goal is not to accurately find gene-promoter pairs but the realization, the following preprocess may not be the most suitable way. First, we aggregated cells within same cell types, to form 'pseudobulks' with 14 samples according to the metadata because psuedobulking provided smoother correlation statistics without loss of the information of interest. Next, remove all the features with 0 standard deviation. Then, log Compute counts per million (CPM) from `edgeR` was used to accounting for different library size.

Since genes' expression is most *cis*-regulated by chromatin accessibility, there is a belief that two modalities would have significant high correlation. And extremely high expressed or low expressed genes would also have high or low accessibility in corresponding cell types. For the whole gene set, the mean correlation of genes and ATAC read counts was 0.33, while the subsampling correlation distribution in Figure S1 F had mean 0.007 across 1000 times block bootstrap. 5644 significant candidate gene-promoter pairs were identified, among which 5591 genes had only one pair, 25 genes had 2 pairs. So those significant pairs could provide important insights into perturb their promoters on chr2:74000098-74003475 and chr6:14116971-14139988 for future tumor treatment.

The block below shows example code for running analysis

```
# split sparse count matrix into NumericList
x <- x_Granges %>%
  mutate(counts_X = NumericList(asplit(x.scaled, 1)))%>% sort()
y <- y_Granges %>%
  mutate(counts_y = NumericList(asplit(y.scaled, 1))) %>% sort()
# First standardize read counts for fast correlation computation
x$counts1 <- NumericList(lapply(x$counts_X,function(z)(z-mean(z))/sd(z)))
y$counts2 <- NumericList(lapply(y$counts_y,function(z)(z-mean(z))/sd(z)))

bootranges <- bootRanges(y,blockLength = 5e5, R=100)

```

```
# for standardized x and y:
correlation = function(x,y) 1/(length(x)-1) * sum(x*y)
## extract bootstrap summary statistics
boot_stats<-x %>% join_overlap_inner(boots, maxgap=1000) %>%
  mutate(rho = correlation(counts_x, counts_y)) %>%
  group_by(iter) %>%
  summarise(meanCor = mean(rho))
```

3 Supplementary Figures

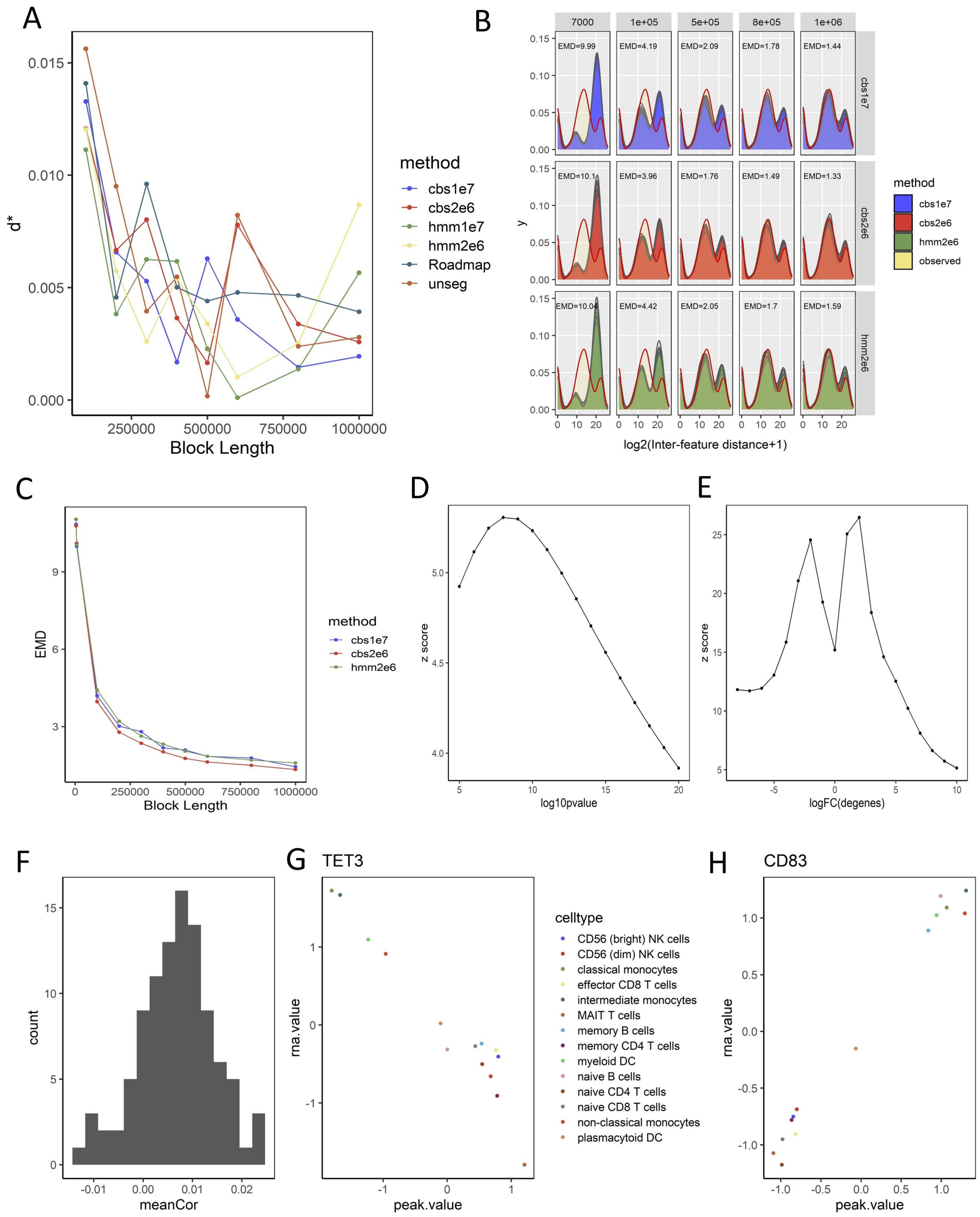

Figure S1: Comparison of block subsampling distributions. A) A pseudometric d^* proposed by Bickel *et al.* (2010) over L_b, B) The null sets' log2(inter-feature distance+1) density plots generated by various block bootstrapped settings over observed feature set's distance. The more null sets' density plots overlapped with observsed features, the better conversing spatial distribution of original set captured. Medan EMD was shown as text in each panel. C) Median EMD over L_b where EMD quantifies similarity between two distributions. D)z score over -log10(pvalue) in caQTL-GWAS analysis. z score indicated the amount of standard deviations that observed fitted overlap rate away from 1000 times block bootstrap's fitted overlap rate. E) z score over DE genes' logFC in macrophage enrichment analysis. z score indicated the amount of standard deviations that observed fitted overlap count away from 1000 times block bootstrap's fitted overlap count. F)The mean correlation distribution of genes expression with 1000 times block bootstrapped ATAC's read counts. G) Gene TET3 read counts over peak chr2:74000098-74003475 read counts, colored by cell types. TET3 has the most negative correlation $\rho = -0.963$. H) Gene CD83 read counts over peak chr6:14116971-14139988 read counts, colored by cell types. The correlation of this gene-promoter pair is 0.992.

References

Bickel, P. J. *et al.* (2010). Subsampling methods for genomic inference. *The Annals of Applied Statistics*, pages 1660–1697.

Buniello, A. *et al.* (2018). The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic Acids Research*, **47**(D1), D1005–D1012.

Dozmorov, M. G. *et al.* (2022). *excluderanges*. https://github.com/mdozmorov/excluderanges/excluderanges - R package version 0.99.6.

Favorov, A. *et al.* (2012). Exploring massive, genome scale datasets with the genometricorr package. *PLoS computational biology*, **8**(5), e1002529.

Gel, B. *et al.* (2016). regioner: an r/bioconductor package for the association analysis of genomic regions based on permutation tests. *Bioinformatics*, **32**(2), 289–291.

Heger, A. *et al.* (2013). Gat: a simulation framework for testing the association of genomic intervals. *Bioinformatics*, **29**(16), 2046–2048.

Lee, C. T. *et al.* (2020a). Poly-enrich: count-based methods for gene set enrichment testing with genomic regions. *NAR genomics and bioinformatics*, **2**(1), lqaa006.

Lee, S. *et al.* (2020b). Fluent genomics with plyranges and tximeta. *F1000Research*, **9**.

Quinlan, A. R. and Hall, I. M. (2010). Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**(6), 841–842.

Ricard Argelaguet, Danila Bredikhin, O. S. and Marioni, J. (2020). Mofa analysis of the chromium single cell multiome atac + gene expression assay. https://raw.githack.com/bioFAM/MOFA2_tutorials/master/R_tutorials/10x_scRNA_scATAC.html.

Sheffield, N. C. and Bock, C. (2016). Lola: enrichment analysis for genomic region sets and regulatory elements in r and bioconductor. *Bioinformatics*, **32**(4), 587–589.

Welch, R. P. *et al.* (2014). Chip-enrich: gene set enrichment testing for chip-seq data. *Nucleic acids research*, **42**(13), e105–e105.