Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: 2 April 2015
Applications Note

OXFORD

Genome analysis

Flexible generation of genomic features sets for null hypothesis testing with bootRanges

Wancen Mu [1], Eric Davis [2], Stuart Lee [3], Mikhail Dozmorov [4], Douglas H. Phanstiel [2], 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.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

bootRanges provides fast functions for generation of bootstrapped feature sets representing the null hypothesis in enrichment analysis. It can also be used in more complex analyses, such as computing correlations between cis-regulatory elements (CREs) and genes. We show that conventional shuffling or permutation schemes may result in overly narrow null distributions, while creating new feature sets with block bootstrap captures more variance in the null distribution. In addition, *bootRanges* provides for analyses with flexible effect size cut-offs, e.g. *p*-value in GWAS, or log fold change in differential expression analysis. The *bootRanges* functions are available in the R/Bioconductor package *nullranges* at `https://bioconductor.org/packages/nullranges`.

1 Introduction

In genomics analyses, there is a common belief in association analysis that significant enrichment(depletion) or correlation between genome features sets indicates biological relationship (De *et al.*, 2014). Therefore, there are always interest of finding association among genomic, transcriptomic and epigenomic features to verify differential expressed SNPs / genes set, or identify transcription factor binding motifs and potential enhancer-gene, enhancer-promoter regulations. While those analysis having statistical significance usually rely on a null distribution. One of the strategy 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 GAT additionally allowing controlling for GC content (Heger *et al.*, 2013), and regioneR implement a circular randomization to preserve inter-feature distance (Gel *et al.*, 2016). However, random-start feature sets will not generally exhibit natural clumping as well as keeping compositional changes of typical genomics features because there often exists a complex dependency structure between features, even when excluded regions are considered. Therefore misleading conclusion can be derived without considering this.

The block bootstrap (Politis *et al.*, 1999) provides an alternative to random starts, where one instead generates random feature sets by sampling large blocks of features within the segments from the original set with replacement to preserve the features' cluster and isochoric property. However, the computational expensive has always been an issue to address, GSC (Bickel *et al.*, 2010) kept swapping a pair of blocks to trying generate a block-wise bootstrap with limited number of statistics. However, we proposed an efficient vectorized code for operation on *GRanges* (Lawrence *et al.*, 2013) objects to generate *bootRanges*. Its provides fully flexible frameworks includes 1) different block bootstrap strategy 2) summarize customizable interested statistics at either block or genome level given tidy downstream pipelines with *plyranges* (Lee *et al.*, 2019). This paper concludes recommended segmentation and block length and shows their impact on the hypothesis test conclusion from shuffling. We additionally sought to perform penalized splines across a range of effect sizes on the null sets, through which to derive confidence interval at the same time on every effect size. Therefore, an optimized effect size threshold could be derived, eg. rather than limiting DEG logFC at a arbitrary threshold.

2 Features

The schematic diagram of a block bootstrap, and the workflow of *bootRanges* in combination with *plyranges* is provided in Figure 1. *bootRanges* offers a simple "unsegmented" as well as a "segmented"

block bootstrap: since the distribution of features in the genome is not uniform, we follow the logic of Bickel *et al.* (2010) and consider to perform block bootstrapping within segments of the genome, which are more homogeneous in terms of base composition and inter-feature distance. We consider various genome segmentation procedures(e.g. based on gene density) or annotations(e.g. Giemsa bands, Segway segments (Hoffman *et al.*, 2012), or ChromHMM segments(Ernst and Kellis, 2012)), defining a number of large (e.g. on the order of megabase pairs), relatively homogeneous segments within which to block bootstrap features. The `GRanges` objects are taken as the inputs for feature $\boldsymbol{x}$ and $\boldsymbol{y}$ with optional metadata columns used for computing a test statistic. Given segmentation method and block length, a *bootRanges* object of $\boldsymbol{y}$ is generated, which is a `GRanges` object with all the ranges concatenated, and iteration, blocks and block length indicated by metadata columns. After deriving the bootstrap distribution of test statistics, an empirical p-value, the number of times that show an equal or greater statistics than the observed statistic, or any types of intervals can be reported for testing the null hypothesis that there is no true biological relevant association between features. The *bootRanges* implementing algorithms are explained schematically in Supplementary Section 1.7.

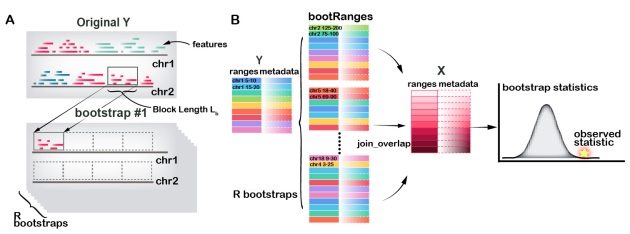

Fig. 1. Overview of airpart framework. (a) The schematic diagram of bootRanges with blockLength L_b across chromosome. (b) Feature $\boldsymbol{x}$ is overlapped with feature $\boldsymbol{y}$ and bootRanges using plyranges join_overlap function to derive the the observed statistic and bootstrap distribution of interested statistics.

3 Application

First, *bootRanges* was evaluated on if there was significant overlap between caQTLs in human liver tissue and SNPs associated with total cholesterol (Currin *et al.*, 2021). Figure 2A shows various segmentation method and block length effect on overlap rate distribution. Here overlap rate was defined as the proportion of SNPs across genome overlapping with peaks within 10kb. The fact that the estimated statistics variance increase at large L_b indicated data were inhomogeneous and segmentation could alleviate the scenario. The exceptional decreasing trend using ChromHMM annotations for Roadmap Epigenomics indicated too many short segments failed to random swap the block genome when L_b close to L_s. Regarding the choice of genome segmentation and of block length selection, we considered a number of diagnostic statistics including those recommended by Bickel *et al.* (2010): the variance of the null distribution of test statistics (Figure 2A) and a scaled version of the change in the width of the test statistic as L_b changes; as well as examination of the inter-feature distance distribution (see Supplementary Methods). After evaluation, $L_b \in [300000, 600000]$ was shown to be a good range for carfully defined null distribution(Figure S1A-C). More details were given in Supplementary Section 2.

The scientific conclusion of this example was there exists a strong association between liver caQTLs and total cholesterol SNPs because of the empirical p value = 0. z score, independent of number of bootstraps, was used to measure the distance between the expected value and the observed one according to the standard deviations. since the the $z = 4.10$ if look at circular binary search(CBS) (Seshan and Olshen, 2021) segmentation method with $L_s = 2e6$ and $L_b = 5e5$(Figure 2B). As seen in applications of Bickel *et al.* (2010), the effect of segmentation did not greatly alter

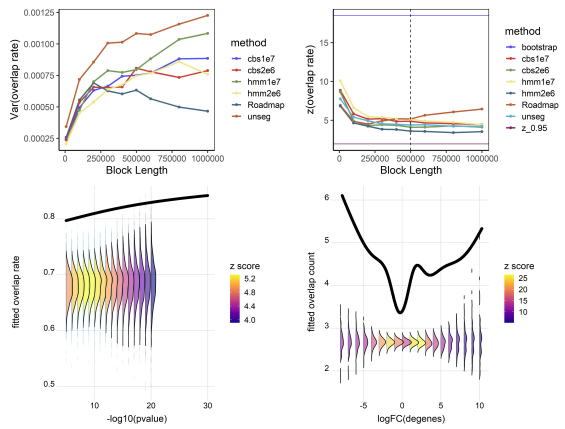

Fig. 2. Results of liver caQTL-GWAS(A-C) and macrophage enrichment analysis(D). A) Variance of overlap rate over various segmentation and L_b choice. B) z score over block length assuming block bootstrap's overlap rate follow Gaussian distribution. Upper and lower horizontal line represents shuffle and upper tailed $z_{0.95}$, separately. C-D) Splines was observsed data's fitted overlap rate over -log10(pvalue) from 5 to 20 or overlap count over log(FC) from -8 to 10. Null sets' fitted value distribution on every integer was plotted by conditional density plot. Color represents the amount of standard deviation(SD) that observed fitted statistics away from null sets' statistics distribution.

conclusions, e.g. rejection of the null hypothesis, in this case, although the z score varies greatly among the different segmentations and block lengths. Genome shuffling (Supplementary Section 1.4) that people usually used when performing such analysis had much higher $z = 18.5$. We believe it may result low specificity in some cases and block bootstrap was, as much as possible, close to actual distribution of genomics elements.

In this study, we showed optimized selection of data driven p-value and logFC through applying *bootRanges* on pairs of features: aforementioned example, and chromatin accessibility and gene expression in a macrophage immune response datasets data (Alasoo *et al.*, 2018). A generalized linear model (GLM) with penalized regression splines from *gam* function in the *mgcv* R package were fitted and *predict_gam* function in the *tidymv* R package were predicted on observed and each null feature sets.

$$log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + f(-log_{10}p), log(\mu) = \beta_0 + f(log_{FC})$$

for rate and count-based statistic, separately. All generated 95% percentile intervals at the same time across a range of effect sizes were displayed by conditional density plot (Figure 2 C,D). We found z score was highest when -log10(pvalue)=8 (Figure S1D) which is quite close to Bonferroni correction and DEG logFC at -2 or 2(Figure S1E).

We additionally applied *bootRanges* to Chromium Single Cell Multiome ATAC + Gene Expression, to assess the correlation of the two modalities for all pairs of genes and promoter peaks, across the 14 cell types (pseudo-bulk). For the whole gene set, the mean correlation of RNA-seq and ATAC log read counts was 0.33 that was significantly far from the subsampling correlation distribution (Figure S1F) with expectation 0.007 and empirical value indicated there was significant high correlation between genes expression and open chromatin. Additionally, average gene-promoter correlation per gene can be derived. XXX of genes have a significant higher correlation.

According to the time efficiency, we compared with GSC using the ENCODE kidney and bladder H3K27ac ChIP-seq peaks. The average time to block bootstrap the whole genome using *bootRanges* is around 0.3s and 0.37s if adding the *plyranges* pipeline. While GSC approximate cost 7.56s. Therefore, *bootRanges* is 20 times faster. All of the R code and data used in this paper are available at the following repository: `https://github.com/Wancen/bootRangespaper`.

Funding

This work was funded by a CZI EOSS award to M.I.L., and a grant to M.I.L. from NIH [NHGRI R01 HG009937].

References

Alasoo, K. *et al.* (2018). Shared genetic effects on chromatin and gene expression indicate a role for enhancer priming in immune response. *Nature genetics*, **50**(3), 424–431.

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

Currin, K. W. *et al.* (2021). Genetic effects on liver chromatin accessibility identify disease regulatory variants. *The American Journal of Human Genetics*, **108**(7), 1169–1189.

De, S. *et al.* (2014). The dilemma of choosing the ideal permutation strategy while estimating statistical significance of genome-wide enrichment. *Briefings in Bioinformatics*, **15**(6), 919–928.

Ernst, J. and Kellis, M. (2012). Chromhmm: automating chromatin-state discovery and characterization. *Nature methods*, **9**(3), 215–216.

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.

Hoffman, M. M. *et al.* (2012). Unsupervised pattern discovery in human chromatin structure through genomic segmentation. *Nature Methods*, **9**, 473–476.

Lawrence, M. *et al.* (2013). Software for computing and annotating genomic ranges. *PLoS computational biology*, **9**(8), e1003118.

Lee, S. *et al.* (2019). Plyranges: A grammar of genomic data transformation. *Genome biology*, **20**(1), 1–10.

Politis, D. N. *et al.* (1999). *Subsampling*. Springer Science & Business Media.

Seshan, V. E. and Olshen, A. (2021). *DNAcopy: DNA copy number data analysis*. R package version 1.66.0.