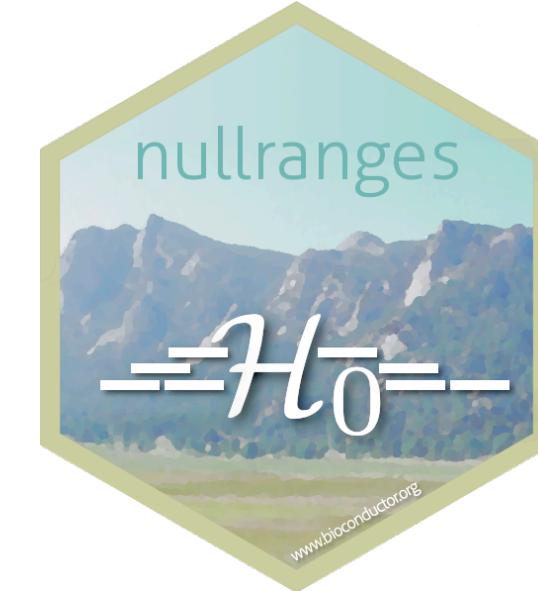




Modular framework:
specialized packages do one thing



Tidy enrichment analysis with *plyranges* and *nullranges*

nullranges: matchRanges() and bootRanges()



Eric Davis, Wancen Mu,
Doug Phanstiel and myself



Mikhail Dozmorov, Stuart Lee, Tim Triche,
others from #nullranges on Bioc Slack

nullranges.github.io/nullranges/
tidybiology.github.io/tidy-ranges-tutorial/

matchRanges: Generating null hypothesis genomic ranges via covariate-matched sampling

Eric S. Davis¹, Wancen Mu², Stuart Lee³, Mikhail G. Dozmorov^{4,5}, Michael I. Love^{2,6*}, Douglas H. Phanstiel^{1,7,8,9,10*}

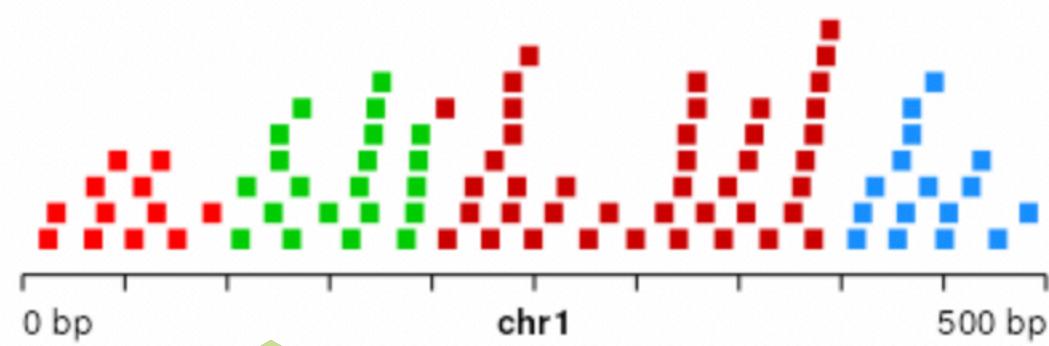
Availability and implementation
<https://nullranges.github.io/nullranges>

Deriving biological insights from genomic data commonly requires comparing attributes of selected genomic loci to a null set of loci. The selection of this null set is non-trivial, as it requires careful consideration of potential covariates, a problem that is exacerbated by the non-uniform distribution of genomic features including genes, enhancers, and transcription factor binding sites. Propensity score-based covariate matching methods allow selection of null sets from a pool of possible items while controlling for multiple covariates; however, existing packages do not operate on genomic data classes and can be slow for large data sets making them difficult to integrate into genomic workflows. To address this, we developed *matchRanges*, a propensity score-based covariate matching method for the efficient and convenient generation of matched null ranges from a set of background ranges within the Bioconductor framework.

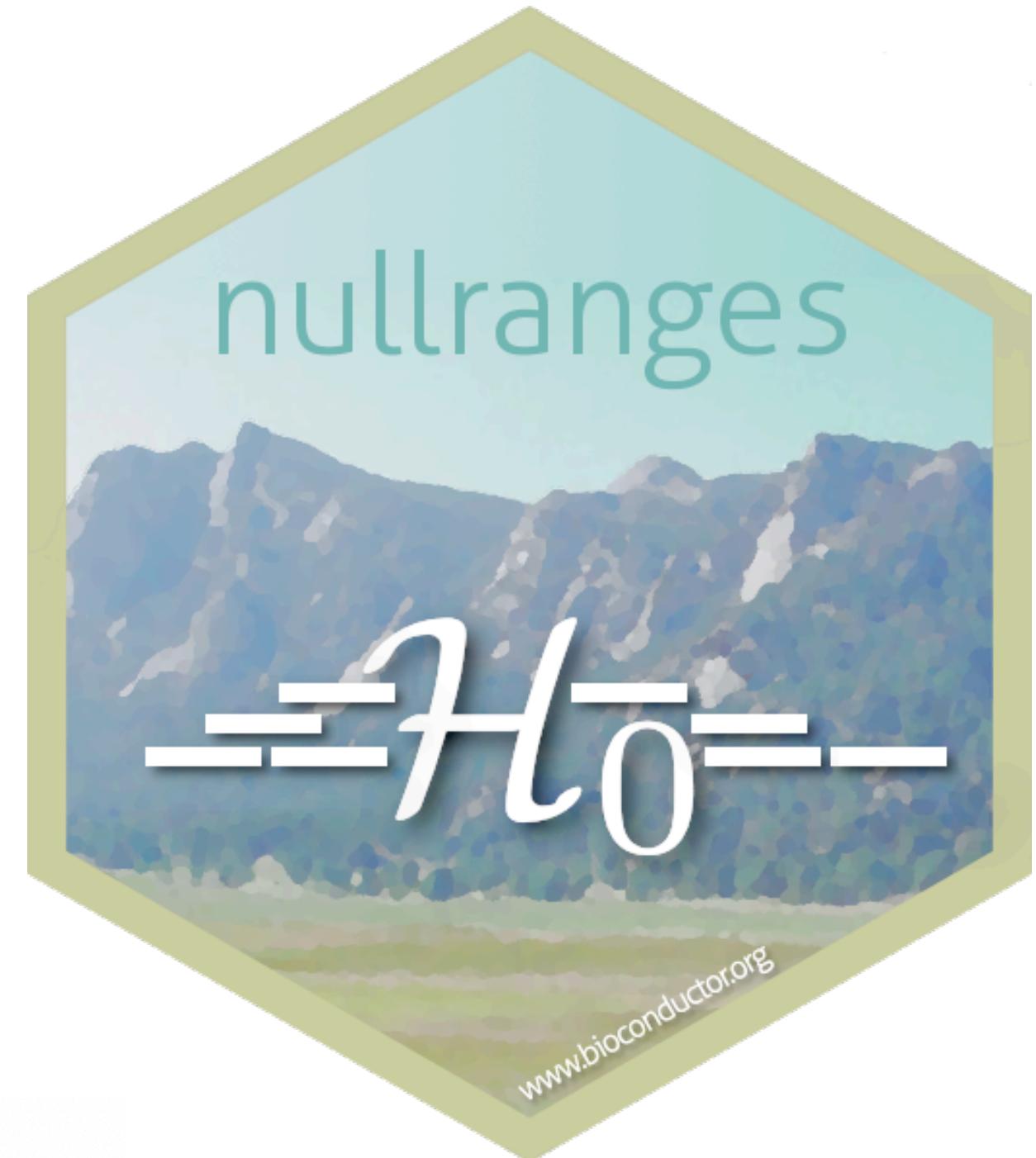
bootRanges: Flexible generation of null sets of genomic ranges for hypothesis testing

Wancen Mu¹, Eric Davis², Stuart Lee⁵, Mikhail Dozmorov⁶, Douglas H. Phanstiel^{2,3}, and Michael I. Love ^{*1,4}

¹Department of Biostatistics,
²Curriculum in Bioinformatics and Computational Biology,
³Thurston Arthritis Research Center, Department of Cell Biology & Physiology, Lineberger Comprehensive Cancer Center, Curriculum in Genetics & Molecular Biology, and
⁴Department of Genetics, University of North Carolina-Chapel Hill, NC 27599
⁵Genentech, South San Francisco, CA, USA
⁶Department of Biostatistics, Department of Pathology, Virginia Commonwealth University, Richmond, VA 23298, USA

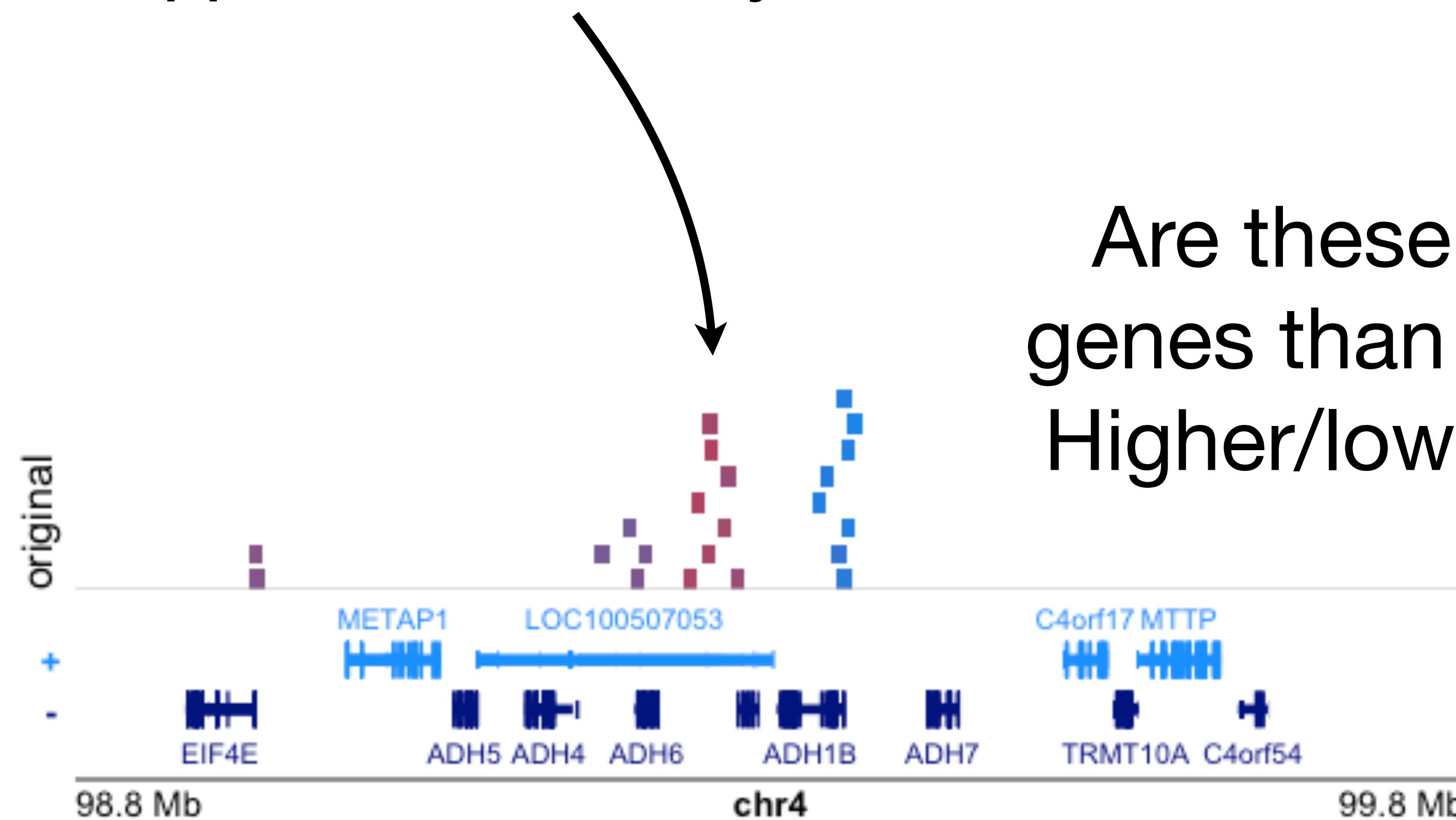


Can visualize null range set with *plotgardener*, by Nicole Kramer, et al.



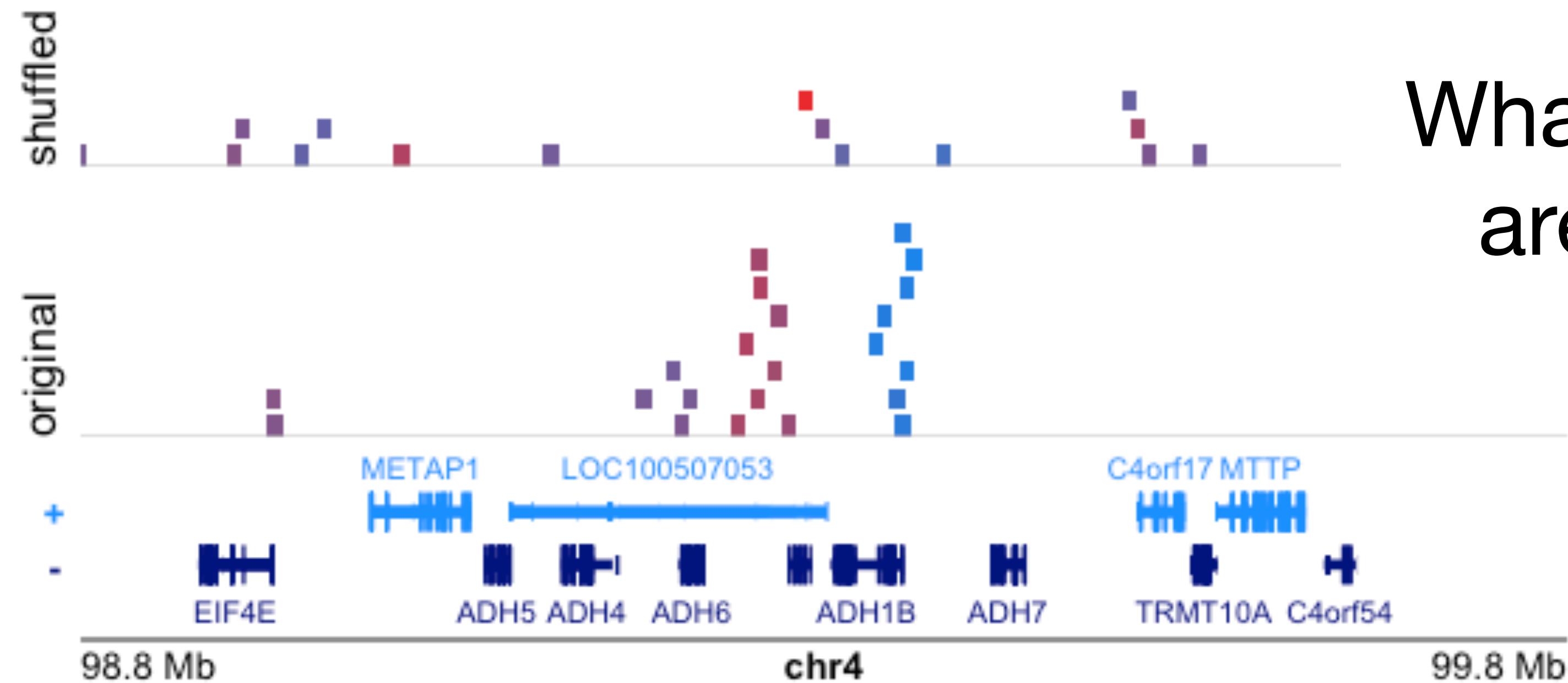
Funded by CZI EOSS,
NIH: R35-GM128645,
R01-HG009937,
T32-GM067553

Consider a set of features (“original”)
e.g. ATAC-seq peaks, color by **score**



Are these closer to
genes than expected?
Higher/lower scores?

Many have noted, shuffling position not a good model



What two things
are missing?

Related work

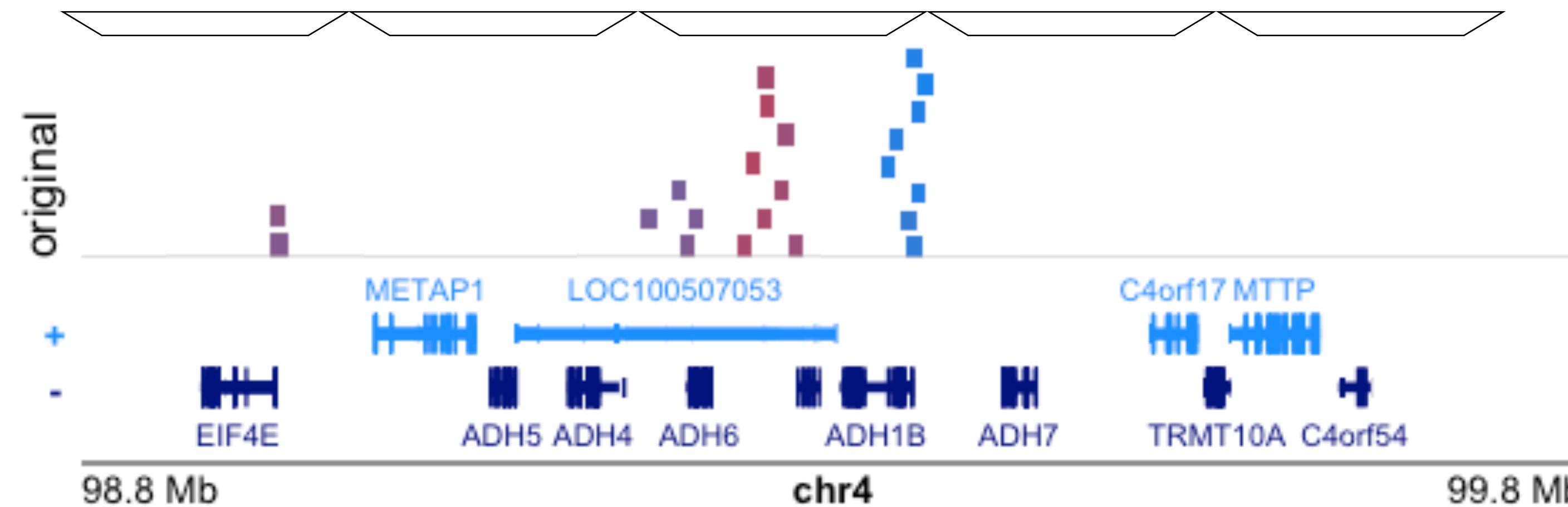
For general considerations of generation of null feature sets or segmentation for enrichment or colocalization analysis, consider the papers of De, Pedersen, and Kechris (2014), Haiminen, Mannila, and Terzi (2007), Huen and Russell (2010), Ferkingstad, Holden, and Sandve (2015), Dozmorov (2017), Kanduri et al. (2019) (with links in references below). Other Bioconductor packages that offer randomization techniques for enrichment analysis include [LOLA](#) (Sheffield and Bock 2016) and [regioneR](#) (Gel et al. 2016). Methods implemented outside of Bioconductor include [GAT](#) (Heger et al. 2013), [GSC](#) (Bickel et al. 2010), [GREAT](#) (McLean et al. 2010), [GenometriCorr](#) (Favorov et al. 2012), [ChIP-Enrich](#) (Welch et al. 2014), and [OLOGRAM](#) (Ferré et al. 2019).

SUBSAMPLING METHODS FOR GENOMIC INFERENCE¹

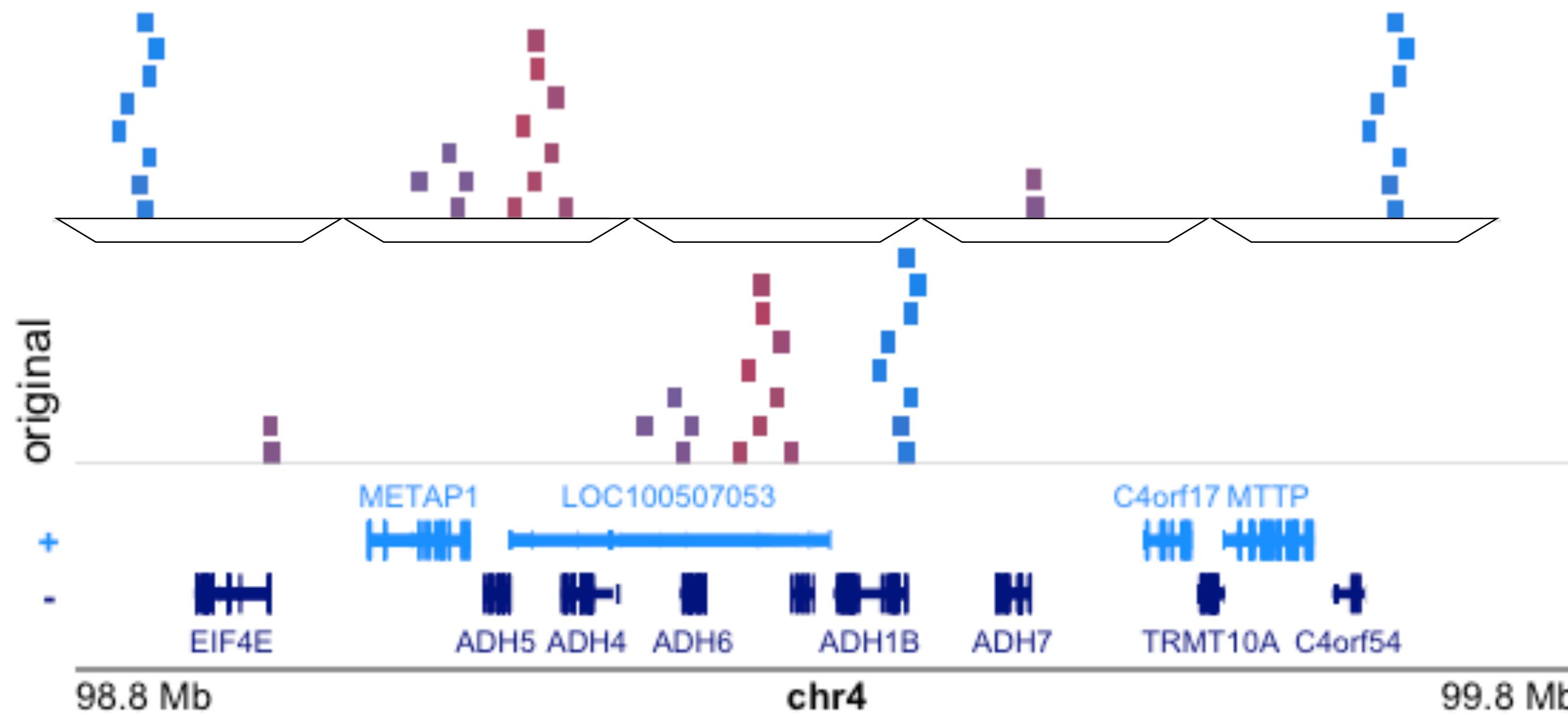
BY PETER J. BICKEL*, NATHAN BOLEY*, JAMES B. BROWN*,
HAIYAN HUANG* AND NANCY R. ZHANG*

*University of California at Berkeley, University of California at Berkeley,
University of California at Berkeley, University of California at Berkeley,
and Stanford University*

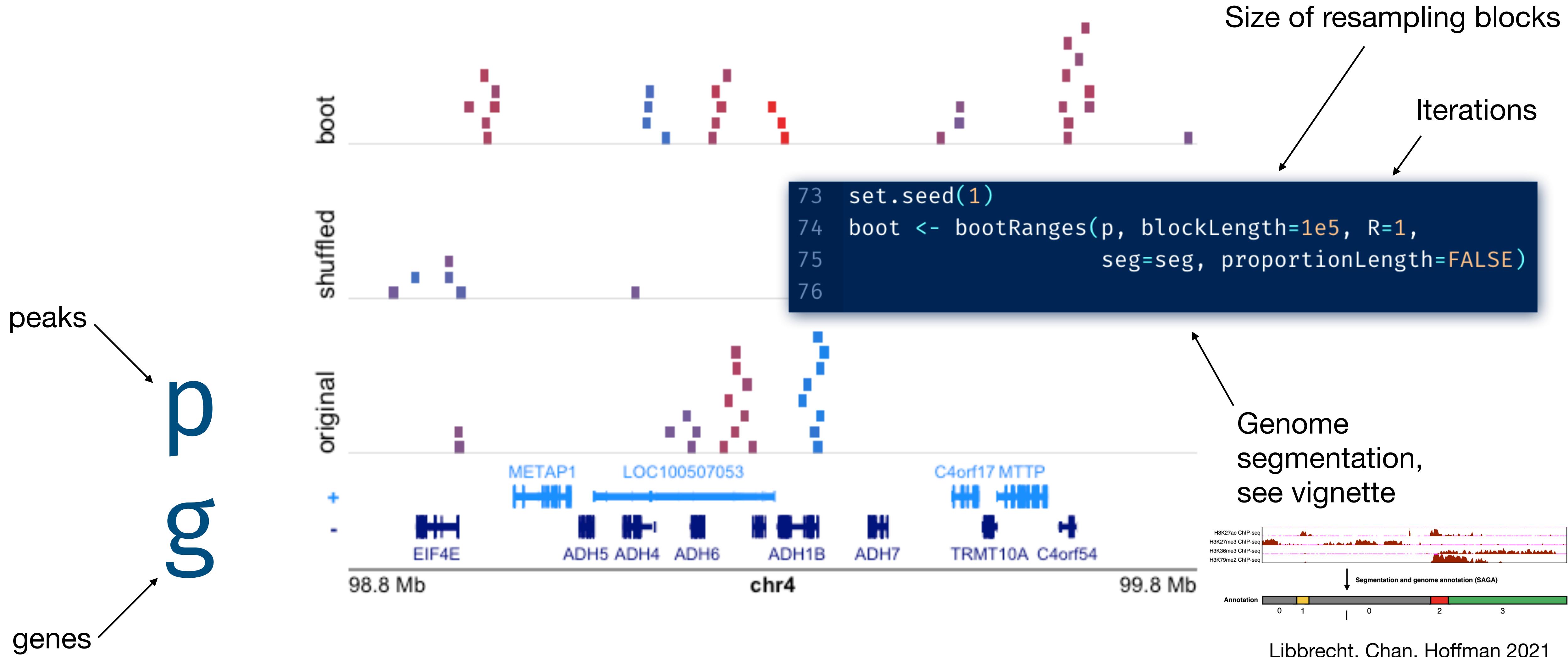
Idea: sample blocks with replacement



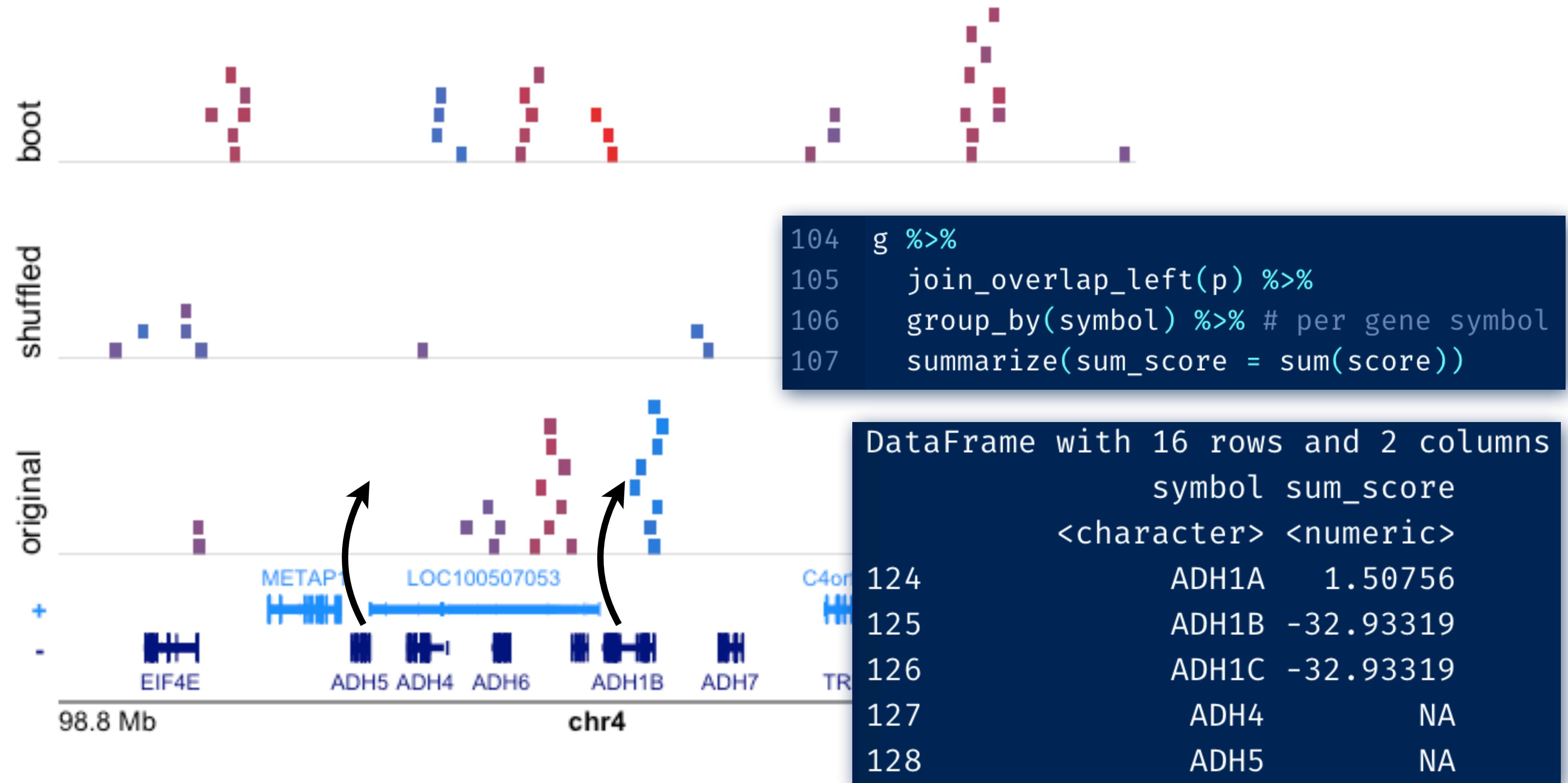
What is preserved?



An actual block bootstrap of a larger region



Overlaps, count or other statistics



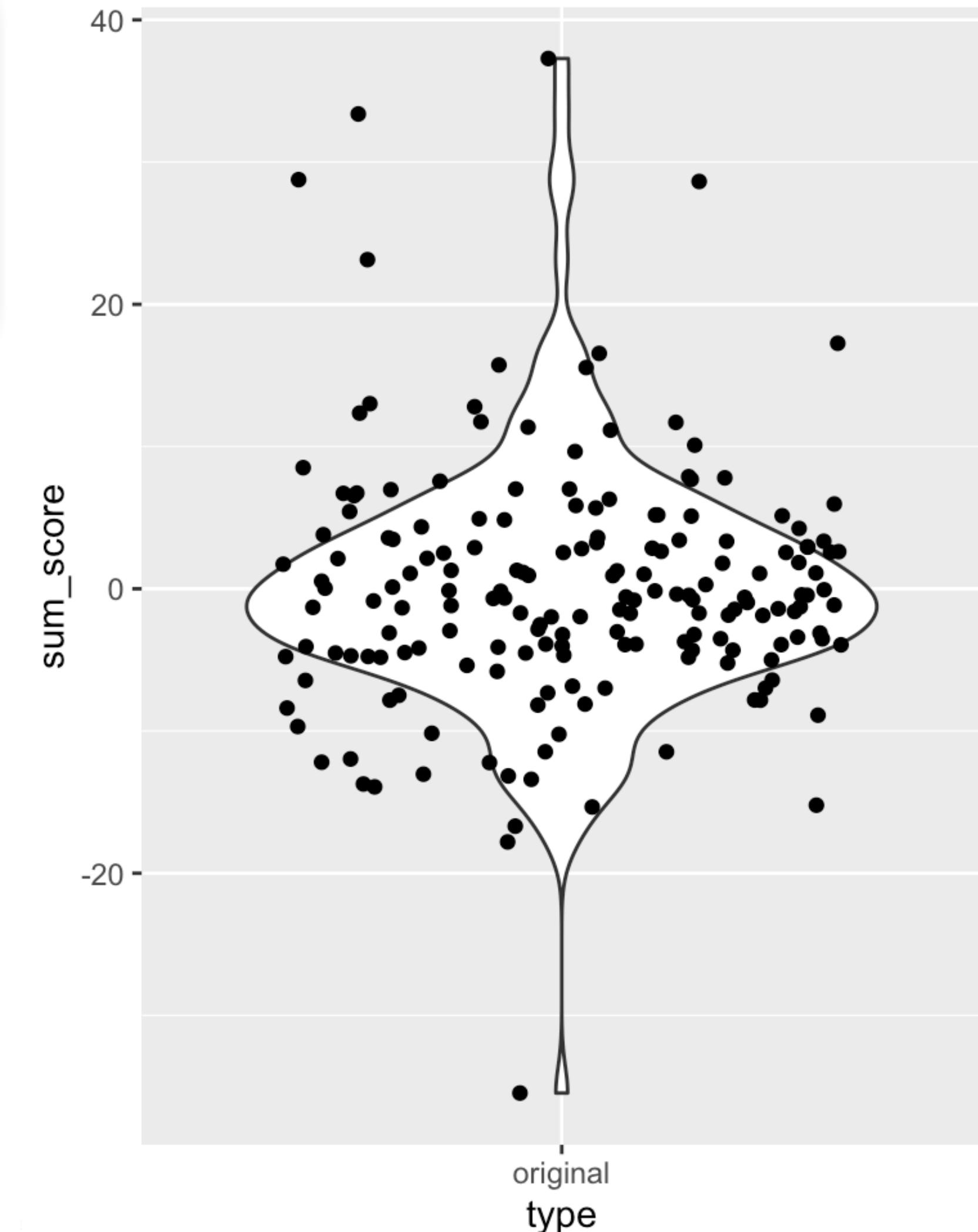
Because we simulate the data, we can sample more draws from distribution

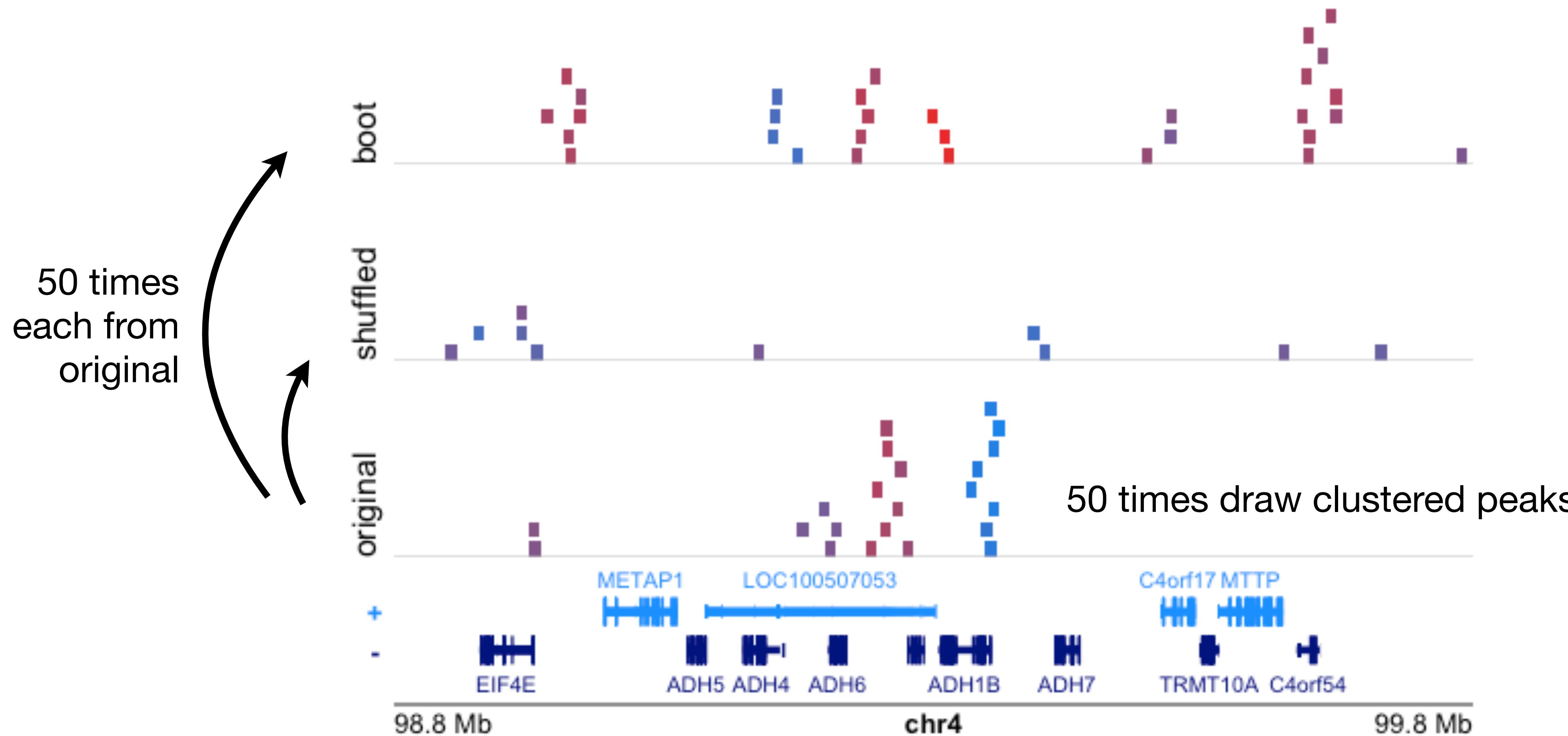
```
126 niter <- 50  
127 sim_list <- replicate(niter, {  
128   makeClusterRanges(chrom, rng_big, n=300, lambda=5, seqlens)  
129 })  
130 sim_long <- bind_ranges(sim_list, .id="iter")
```

```
132 g %>%  
133   join_overlap_inner(sim_long) %>%  
134   mutate(type = "original") %>%  
135   group_by(symbol, iter, type) %>%  
136   summarize(sum_score = sum(score))
```

Adds an iteration column

“inner” removes no overlaps



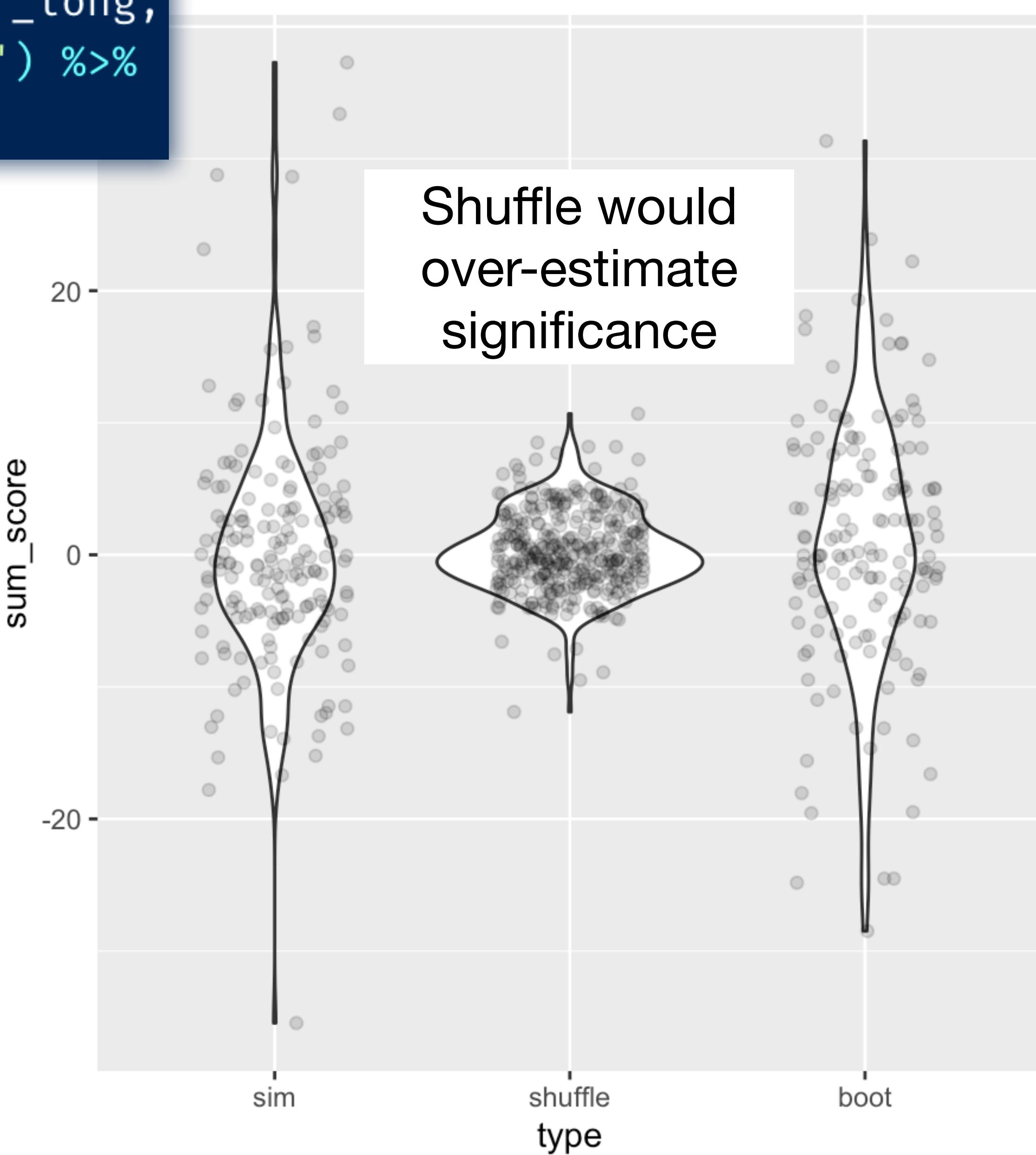


```
153 all <- bind_ranges(sim=sim_long, shuffle=shuf_long,  
154                      boot=boot_long, .id="type") %>%  
155   mutate(type = factor(type, levels=lvls))
```

```
163 g %>%  
164   join_overlap_inner(all) %>%  
165   group_by(symbol, iter, type) %>%  
166   summarize(sum_score = sum(score)) %>%  
167   as_tibble() %>%  
168   ggplot(aes(type, sum_score)) +  
169   geom_violin() +  
170   geom_jitter(width=.25, alpha=.15)
```

What is missing?

```
tidy::complete(  
  symbol, iter, type  
  fill=list(sum_score = 0)  
)
```



Review mechanics of bootstrapping

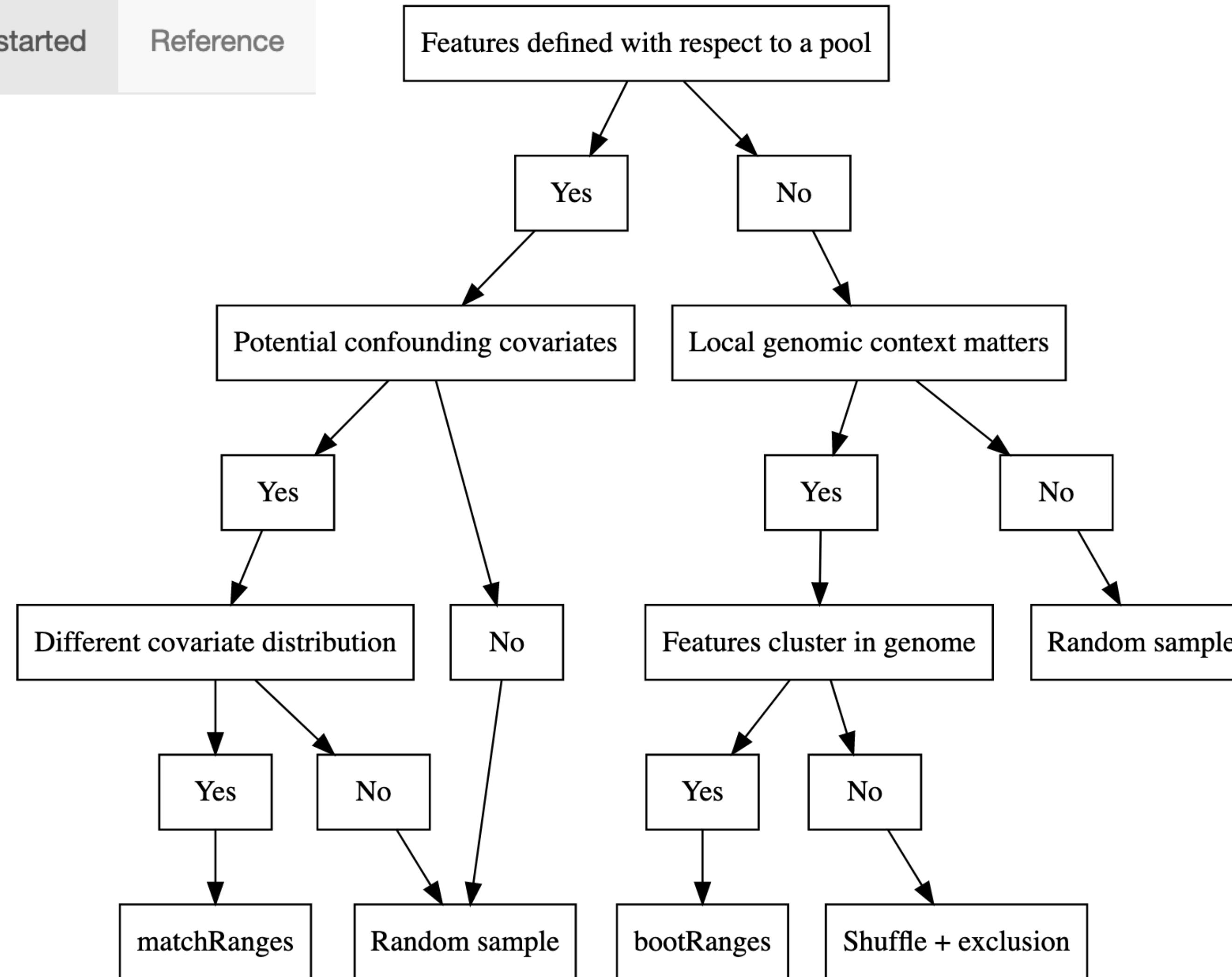
- Avoid copy/paste code with `bind_ranges()` specifying `.id`
 - Alternatively set `observed` to `iter=0`
- A single `join_overlap`
- Followed by `group_by(iter)` and `summarize()` and `complete()`

Which null generating method to choose?

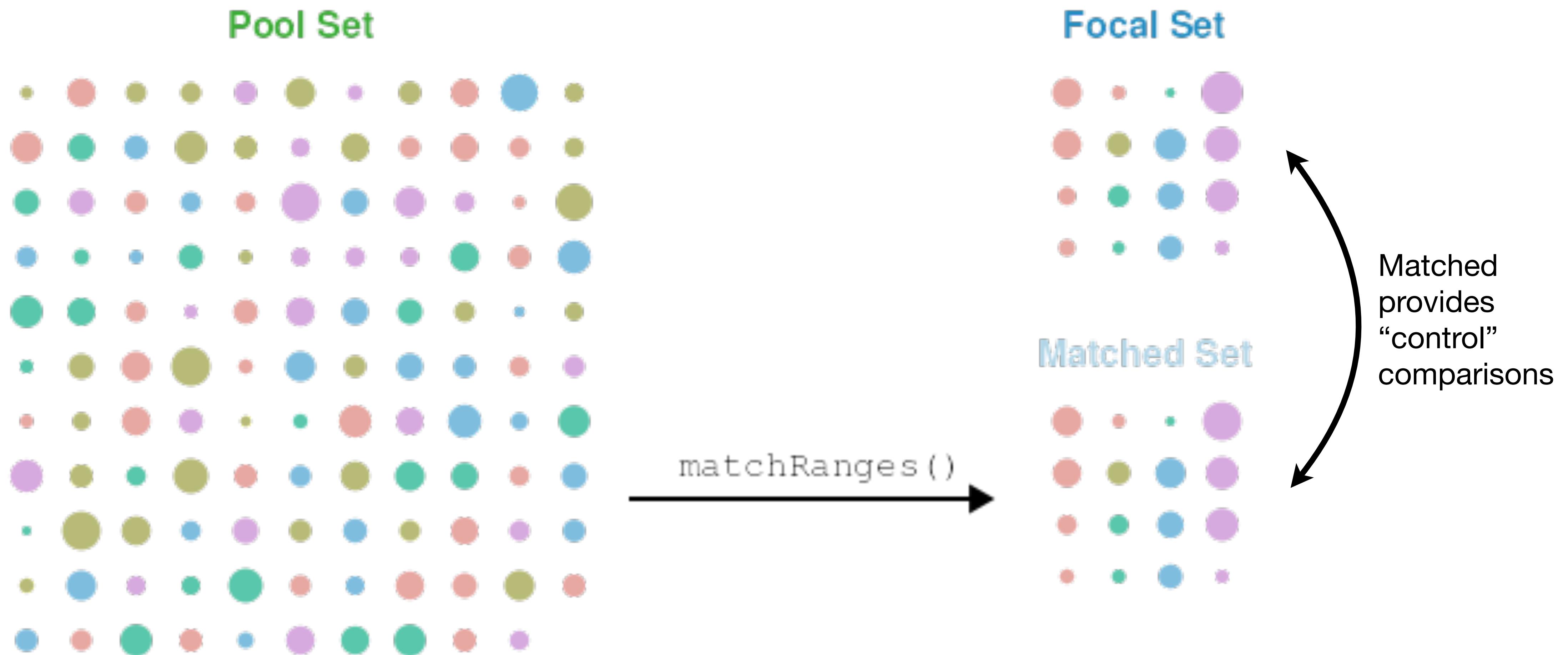
nullranges 1.5.5

Get started

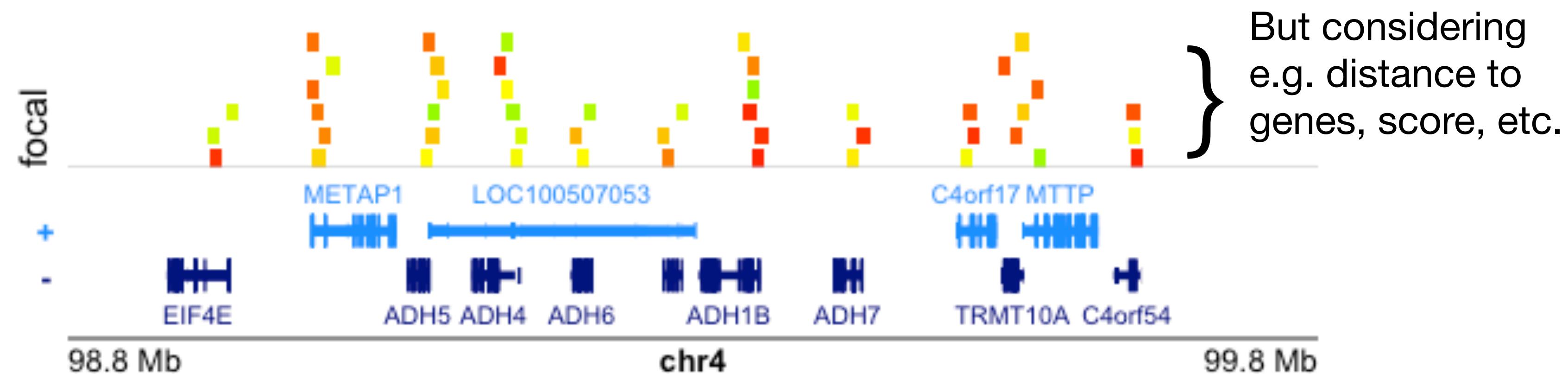
Reference



Quick introduction to matching

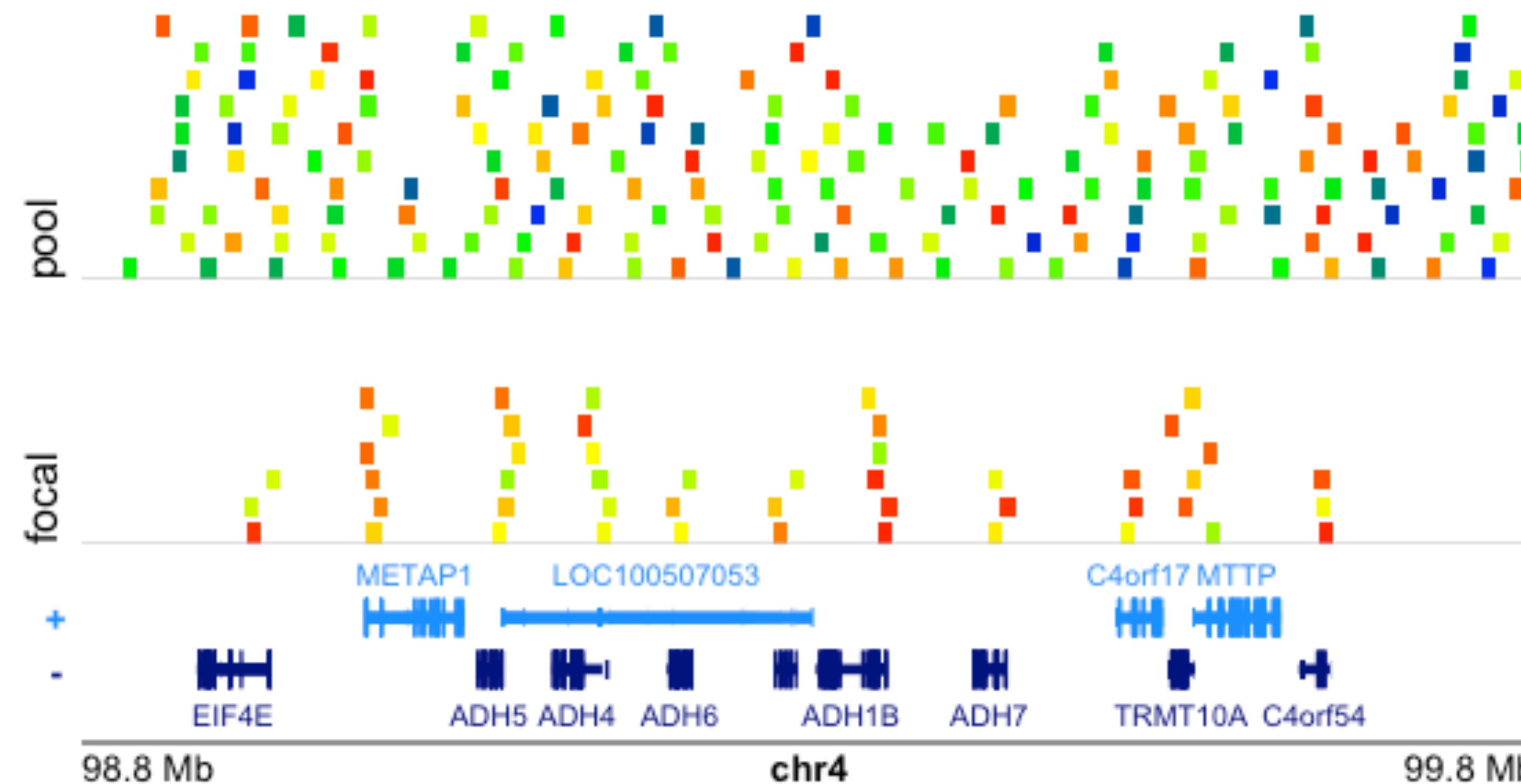


Want: compare focal set to some background ranges, in terms of internal properties, or overlap

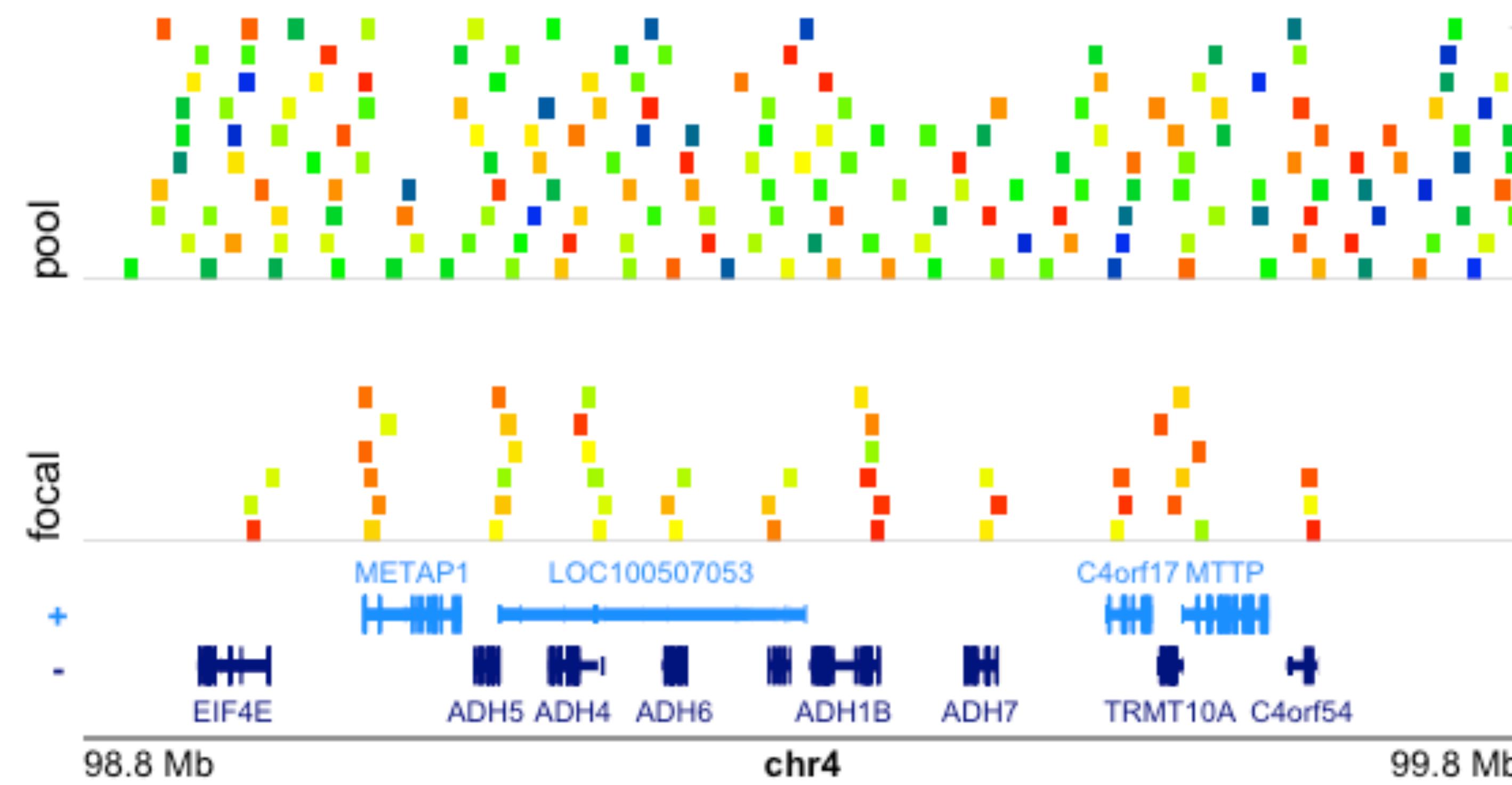


Suppose a much larger pool to select from

```
205 # add another feature: distance to nearest TSS  
206 tss <- g %>% anchor_5p() %>% mutate(width=1)  
207  
208 both <- bind_ranges(focal = focal, pool = pool, .id="type") %>%  
209     add_nearest_distance(tss) %>%  
210     mutate(log10dist = log10(distance + 1000))
```

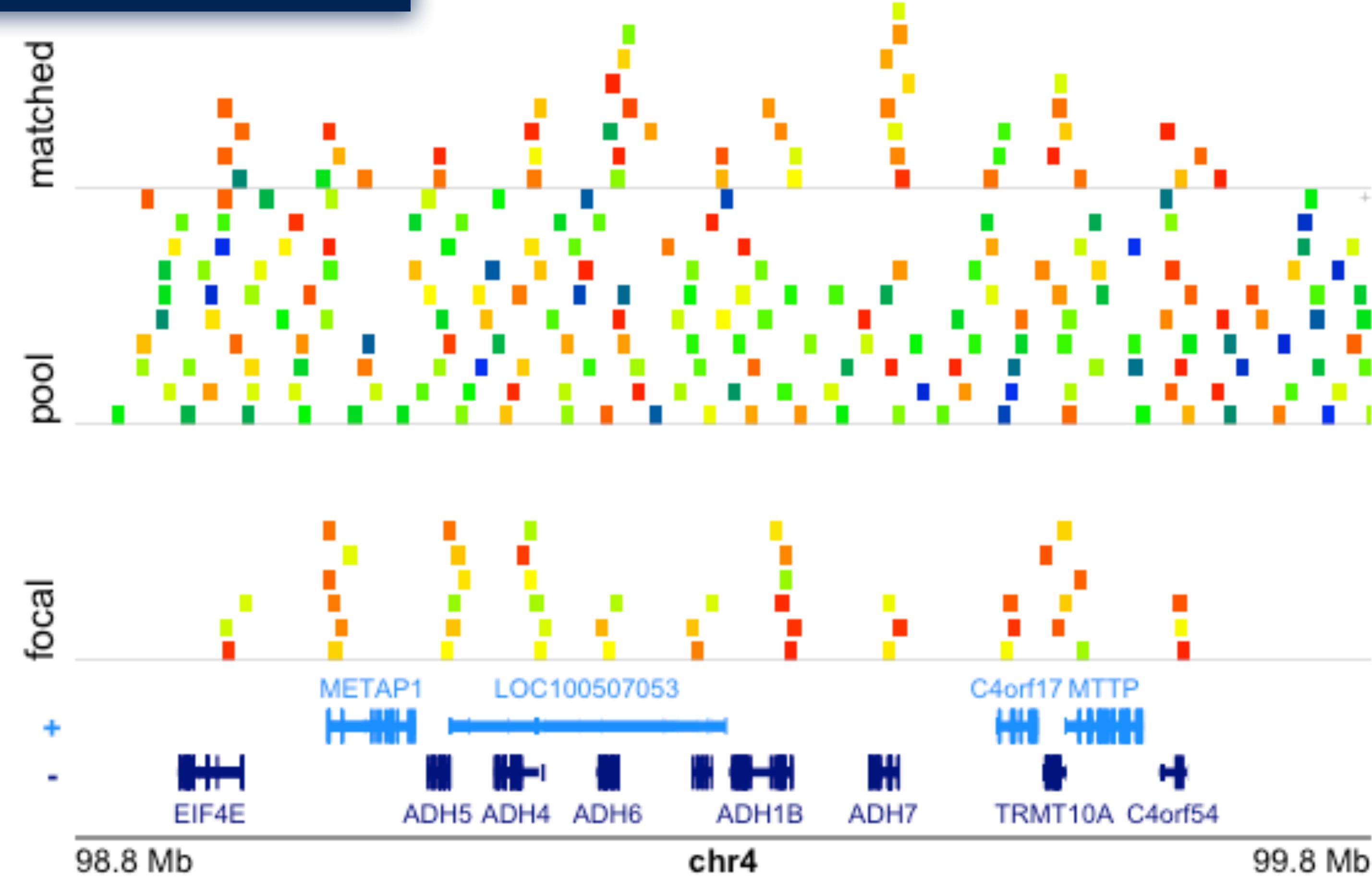


```
215 ▾ m <- both %>% {  
216   matchRanges(filter(., type=="focal"),  
217     filter(., type=="pool"),  
218     covar=~score + log10dist,  
219     method="nearest", replace=TRUE)  
220 ▴ }
```

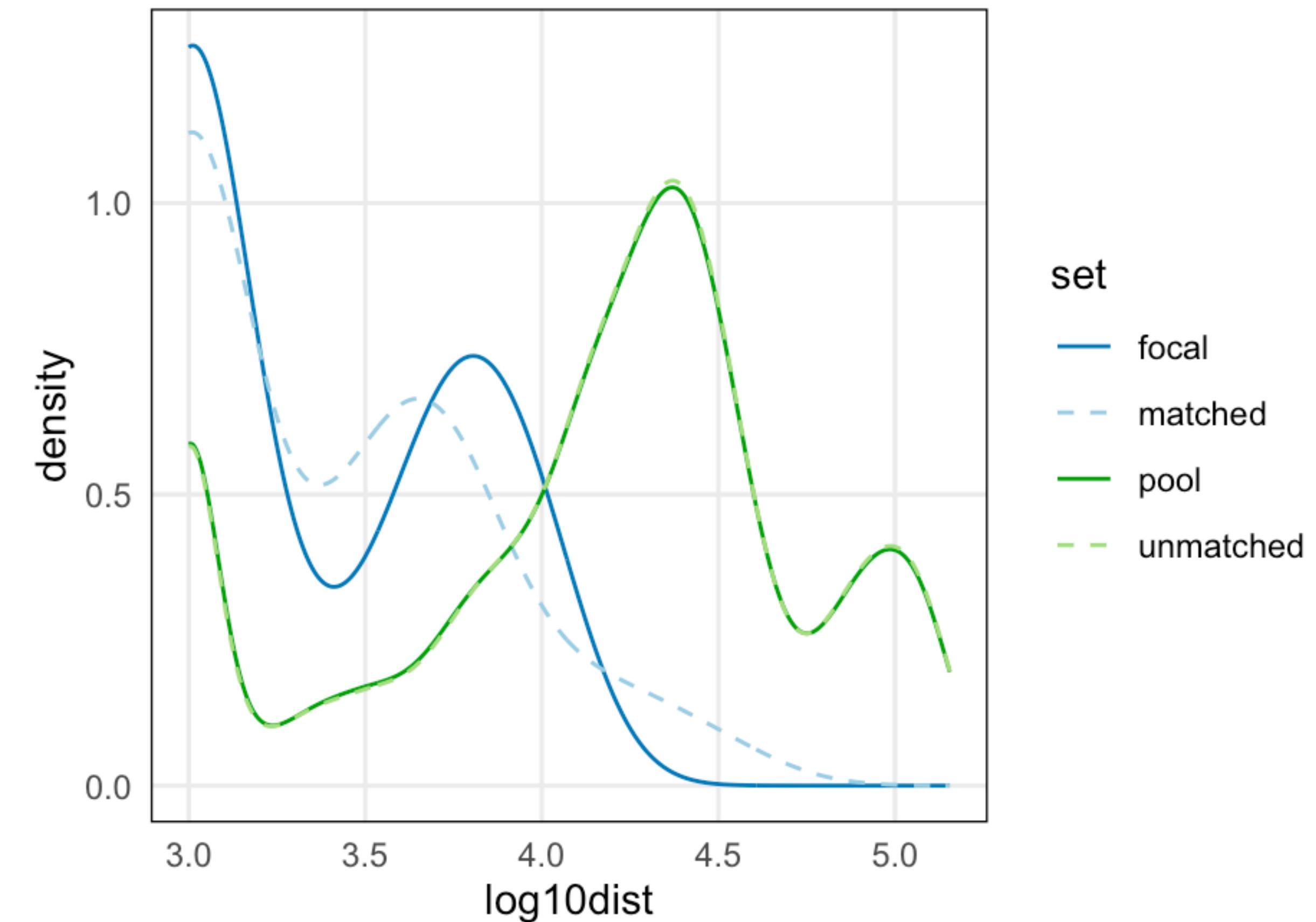
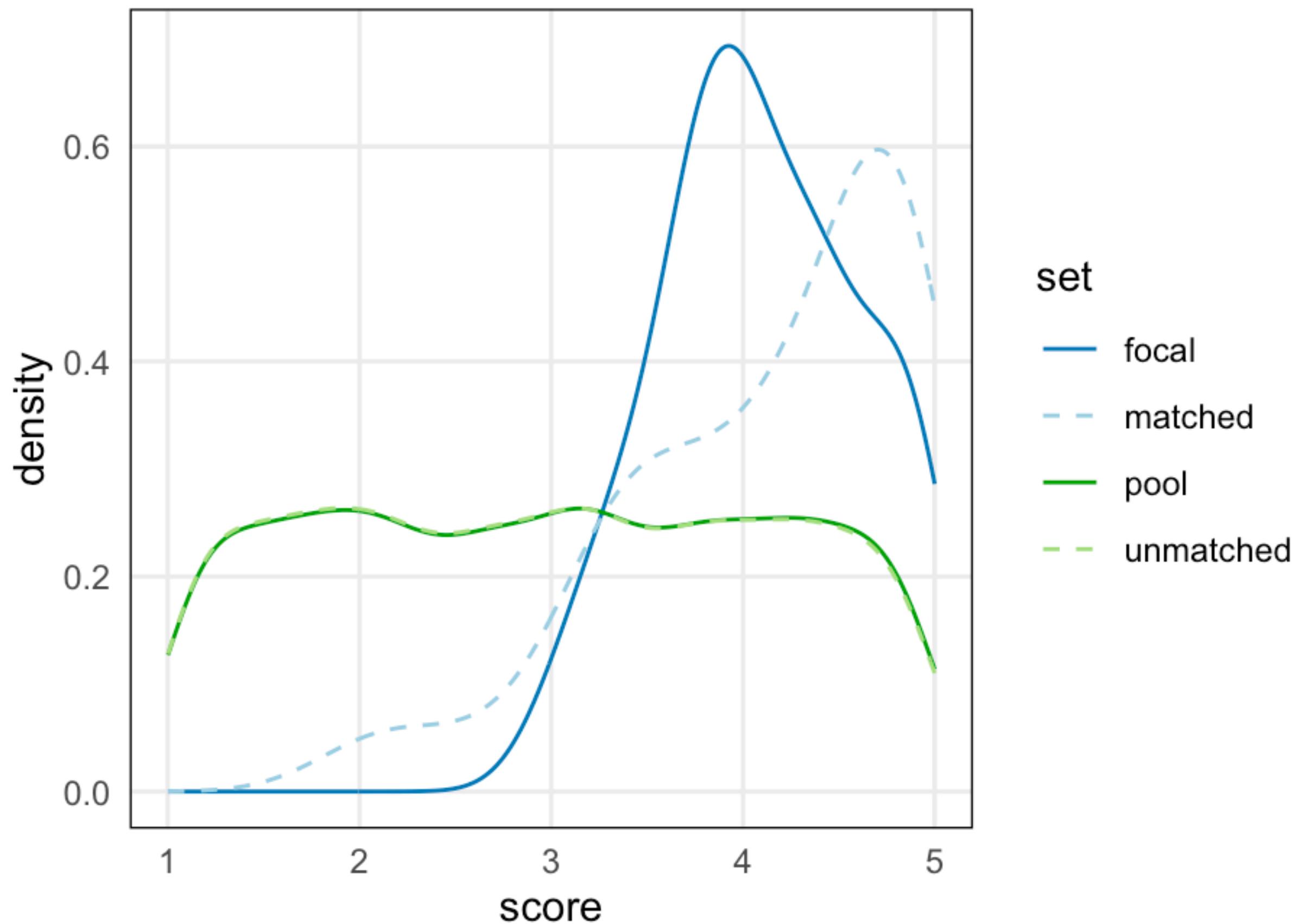


```
215 m <- both %>% {  
216   matchRanges(filter(., type=="focal"),  
217     filter(., type=="pool"),  
218     covar=~score + log10dist,  
219     method="nearest", replace=TRUE)  
220 }
```

```
method = c("nearest", "rejection", "stratified")
```



Bigger the pool, better covariate balance



```
plotCovariate(m, covar="score") +  
plotCovariate(m, covar="log10dist")
```

nullranges: matchRanges() and bootRanges()



Eric Davis, Wancen Mu,
Doug Phanstiel and myself



Mikhail Dozmorov, Stuart Lee, Tim Triche,
others from #nullranges on Bioc Slack

nullranges.github.io/nullranges/
tidybiology.github.io/tidy-ranges-tutorial/

matchRanges: Generating null hypothesis genomic ranges via covariate-matched sampling

Eric S. Davis¹, Wancen Mu², Stuart Lee³, Mikhail G. Dozmorov^{4,5}, Michael I. Love^{2,6*}, Douglas H. Phanstiel^{1,7,8,9,10*}

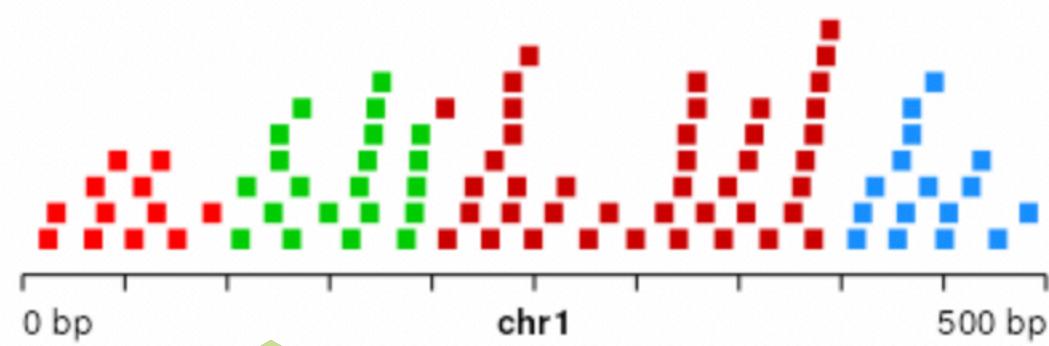
Availability and implementation
<https://nullranges.github.io/nullranges>

Deriving biological insights from genomic data commonly requires comparing attributes of selected genomic loci to a null set of loci. The selection of this null set is non-trivial, as it requires careful consideration of potential covariates, a problem that is exacerbated by the non-uniform distribution of genomic features including genes, enhancers, and transcription factor binding sites. Propensity score-based covariate matching methods allow selection of null sets from a pool of possible items while controlling for multiple covariates; however, existing packages do not operate on genomic data classes and can be slow for large data sets making them difficult to integrate into genomic workflows. To address this, we developed *matchRanges*, a propensity score-based covariate matching method for the efficient and convenient generation of matched null ranges from a set of background ranges within the Bioconductor framework.

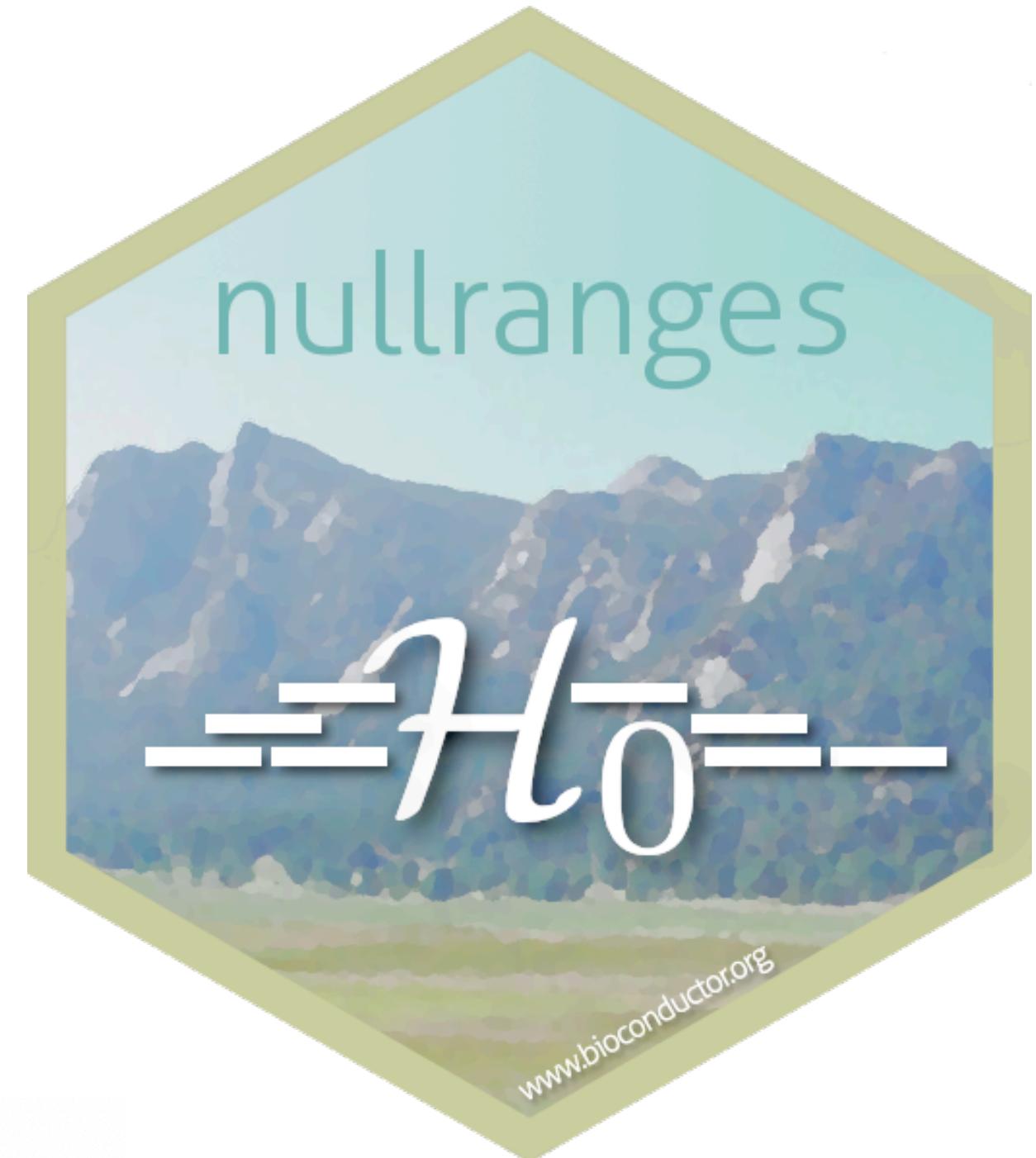
bootRanges: Flexible generation of null sets of genomic ranges for hypothesis testing

Wancen Mu¹, Eric Davis², Stuart Lee⁵, Mikhail Dozmorov⁶, Douglas H. Phanstiel^{2,3}, and Michael I. Love ^{*1,4}

¹Department of Biostatistics,
²Curriculum in Bioinformatics and Computational Biology,
³Thurston Arthritis Research Center, Department of Cell Biology & Physiology, Lineberger Comprehensive Cancer Center, Curriculum in Genetics & Molecular Biology, and
⁴Department of Genetics, University of North Carolina-Chapel Hill, NC 27599
⁵Genentech, South San Francisco, CA, USA
⁶Department of Biostatistics, Department of Pathology, Virginia Commonwealth University, Richmond, VA 23298, USA



Can visualize null range set with *plotgardener*, by Nicole Kramer, et al.



Funded by CZI EOSS,
NIH: R35-GM128645,
R01-HG009937,
T32-GM067553