RiVIERA-MT: Risk Variants Inference using Epigenomic Reference Annotations to predict Multiple Trait-causing colocalized mutations

Yue Li
liyue@mit.edu

June 19, 2017

1 RiVIERA-MT

```
> library(RcppArmadillo)
> library(RiVIERA)
```

2 Brief introduction

RiVIERA-MT is or a full Bayesian framework to fine-map causal variants using GWAS summary statistics in z-scores in one or multiple related traits and large-scale reference annotations in binary or continuous values [2]. The goal of RiVIERA is to infer for each SNP in disease their posterior probability of disease association and to detect functional enrichments or depletions from the annotations, taking into account the underlying multi-trait epigenomic covariance.

3 Data preparation

To run RiVIERA-MT, we will need three types of data:

1. GWAS summary statistics in a nested list of of Z-scores variants for at least one single trait
2. Linkage disequilibrium (LD) either from the GWAS cohort or reference panel (e.g. 1000 Genome consortium)
3. Functional annotation matrix (binary or continuous) for each SNP

For illustration purpose, we provide a simulated dataset saved in RData file and loaded as follows:

```
> rda <- list.files(system.file("extdata", package="RiVIERA"),
+                               "RData", full.names=T)[1]
> load(rda)
> zscoreList <- list(do.call(rbind, simdata$gwasZval[[1]]))
> ldmatList <- simdata$hapr2
> annList <- list(do.call(rbind, simdata$ann))
> locusCursorList <- list(get_locusCursors(ldmatList[[1]]))
> locusNum <- length(ldmatList[[1]])
```

where the `get_locusCursors` generate the start and end positions of the SNP in the zscore matrix. Please become familiar with format of the input to able to process your own data the way. We also provide a wrapper function that takes a set of text files and process them into the require inputs (TO-DO).

4 Inferring risk variants

To train the model and infer causal variants, simply issue the following command:

```
> set.seed(23)
> max_iter <- 20
> #### riviera anno ####
> res <- riviera(
+   zscoreList,
+   ldmatList,
+   annList,
+   locusCursorList,
+   ann_w_mu=matrix(0, ncol(annList[[1]]),length(zscoreList)),
+   max_iter =max_iter,
+   useAnn = TRUE,
+   sampleConfig_iter=10,
+   step = 0.01,
+   nsteps = 100,
+   printfreq = 1,
+   verbose = T)
```

```
Settings:
step = 0.01; nsteps = 100
printfreq = 1; verbose = 1
Initializing parameters ... Done!
1, lprS: 666.31, AFS: 0 (0%), lprW: -855.825, AFW: 1 (100%)
2, lprS: 689.721, AFS: 0 (0%), lprW: -866.102, AFW: 1 (50%)
3, lprS: 697.713, AFS: 0 (0%), lprW: -882.059, AFW: 2 (67%)
4, lprS: 707.966, AFS: 0 (0%), lprW: -851.121, AFW: 3 (75%)
5, lprS: 726.398, AFS: 0 (0%), lprW: -730.916, AFW: 4 (80%)
6, lprS: 725.503, AFS: 0 (0%), lprW: -686.205, AFW: 5 (83%)
```

```
7, lprS: 727.691, AFS: 0 (0%), lprW: -662.549, AFW: 5 (71%)
8, lprS: 1010.32, AFS: 1 (13%), lprW: -651.275, AFW: 6 (75%)
9, lprS: 1008.85, AFS: 1 (11%), lprW: -622.542, AFW: 7 (78%)
10, lprS: 1033.5, AFS: 2 (20%), lprW: -613.001, AFW: 8 (80%)
11, lprS: 1055.38, AFS: 3 (27%), lprW: -614.108, AFW: 9 (82%)
12, lprS: 1070.69, AFS: 4 (33%), lprW: -622.611, AFW: 10 (83%)
13, lprS: 1066.74, AFS: 5 (38%), lprW: -601.616, AFW: 11 (85%)
14, lprS: 1077.12, AFS: 6 (43%), lprW: -564.78, AFW: 11 (79%)
15, lprS: 1077.28, AFS: 7 (47%), lprW: -542.785, AFW: 12 (80%)
16, lprS: 1087.12, AFS: 8 (50%), lprW: -552.344, AFW: 13 (81%)
17, lprS: 1089.6, AFS: 9 (53%), lprW: -540.003, AFW: 14 (82%)
18, lprS: 1089.15, AFS: 10 (56%), lprW: -534.862, AFW: 15 (83%)
19, lprS: 1094.14, AFS: 11 (58%), lprW: -522.373, AFW: 16 (84%)
20, lprS: 1090.21, AFS: 12 (60%), lprW: -510.625, AFW: 17 (85%)
MCMC inference completed.
Accepted annotation-informed models: 11

> #### riviera anno-free ####
> res_noAnn <- riviera(
+   zscoreList,
+   ldmatList,
+   annList,
+   locusCursorList,
+   ann_w_mu=matrix(0, ncol(annList[[1]]),length(zscoreList)),
+   max_iter =max_iter,
+   useAnn = FALSE,
+   sampleConfig_iter=10,
+   step = 0.01,
+   nsteps = 100,
+   printfreq = 1,
+   verbose = T)

Settings:
step = 0.01; nsteps = 100
printfreq = 1; verbose = 1
Initializing parameters ... Done!
1, lprS: 662.677, AFS: 0 (0%)
2, lprS: 941.9, AFS: 1 (50%)
3, lprS: 989.122, AFS: 2 (67%)
4, lprS: 1006.68, AFS: 3 (75%)
5, lprS: 1028.79, AFS: 4 (80%)
6, lprS: 1042.08, AFS: 5 (83%)
7, lprS: 1049.7, AFS: 6 (86%)
8, lprS: 1056.09, AFS: 7 (88%)
9, lprS: 1060.43, AFS: 8 (89%)
```

```
10, lprS: 1065.77, AFS: 9 (90%)
11, lprS: 1068.27, AFS: 10 (91%)
12, lprS: 1071.7, AFS: 11 (92%)
13, lprS: 1082.96, AFS: 12 (92%)
14, lprS: 1080.73, AFS: 13 (93%)
15, lprS: 1080.46, AFS: 14 (93%)
16, lprS: 1087.4, AFS: 15 (94%)
17, lprS: 1084.82, AFS: 16 (94%)
18, lprS: 1090.18, AFS: 17 (94%)
19, lprS: 1086.15, AFS: 18 (95%)
20, lprS: 1092.27, AFS: 19 (95%)
MCMC inference completed.
```

```
> burnFrac <- 0.2
> burnIdx <- 1:round(burnFrac * length(res$pipList_ensemble))
> pred <- averagePIP(res$pipList_ensemble[-burnIdx])[[1]]
> burnIdx_noAnn <- 1:round(burnFrac * length(res$pipList_ensemble))
> pred_noAnn <- averagePIP(res_noAnn$pipList_ensemble[-burnIdx])[[1]]
> # pred <- zscoreList[[1]]
> rownames(pred) <- rownames(pred_noAnn) <- rownames(zscoreList[[1]])
```

For comparison, we ran both the model with and without annotations. Sampled weights indicate causal annotations. The first K (i.e., K=3) annotations are causal, and each causal annotation contain 1/K causal varaints (no restriction on overlap). Model using annotations identifies accurately the causal annotations (Fig. 1).

Sampled locus-specific variance correlate with number of causal variants harbored in the loci (Fig. 2).

Model using annotations perform the best in identifying causal varaints (Fig. 3).

We now visualize the finemapping results in 3 representative loci in the plots below. In the first scenario, the risk locus harbors one causal variant (red cricle), which drives the genetic signals of other non-causal variants via linkage disequilibrium (LD). Notably, the lead SNP (dark diamond) with the most significant p-value is not the causal variant. In this case scenario, the underlying epigenomic activities (middle track) provide a crucial evidence to the inference of functional variants. Methods that assume single causal variant per locus may work well here by normalizing the posterior for each SNP within the locus [3, 1]. However, these methods become inadequate when there are more than one causal variant within the same locus (Fig. **??**b,c) because they will pull down the true signals of all causal variants in order to maintain a properly normalized posterior probabilities.

For instance, Fig. 6,7 and Fig. 8,9 illustrate two loci containing 3 and as many as 10 causal variants in the same loci, respectively. In these cases, our RiVIERA-MT model is still able to efficiently infer the correct PIP by marginalizing over a large number of sampled causal configurations with high local posteriors, which automatically accounts for the potentially large number of causal variants within the same locus without predefining the number of causal variants per locus.

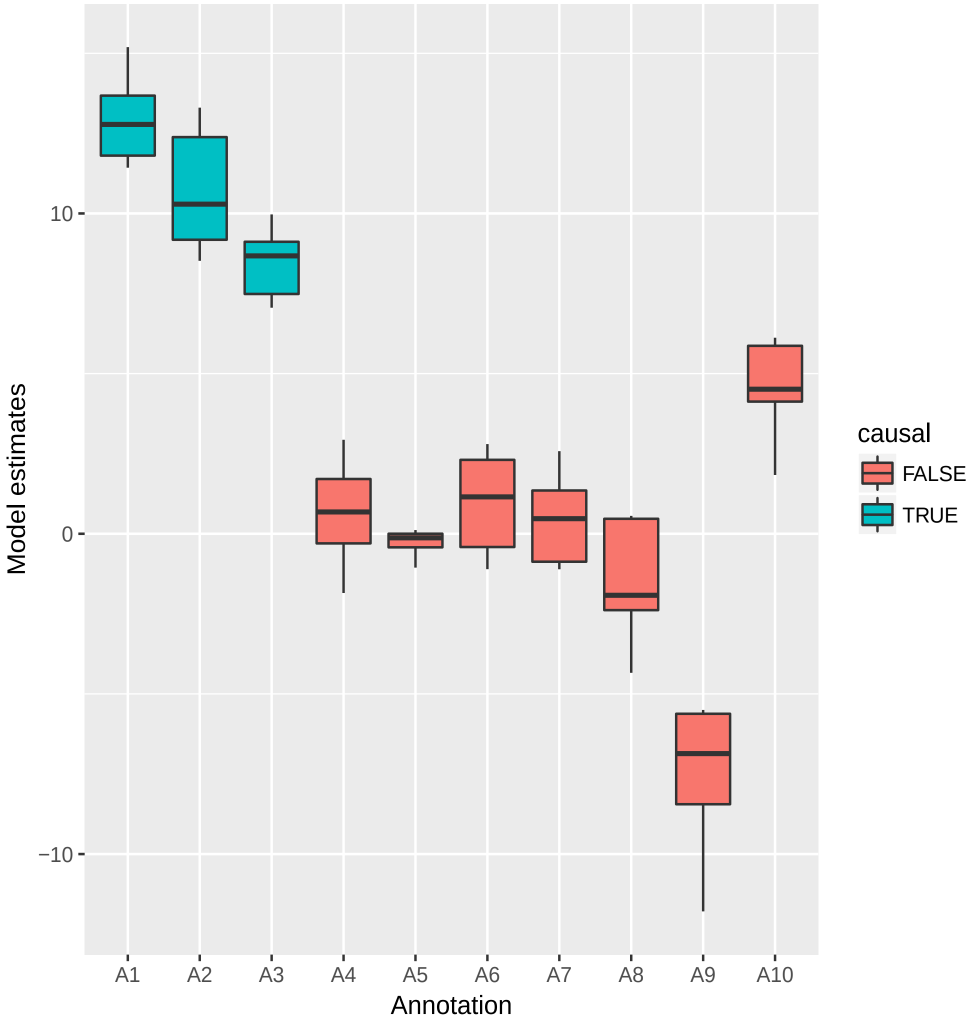

Figure 1: Sample distribution of enrichment weights for each annotations. Causal and non-causal annotations are colored above.

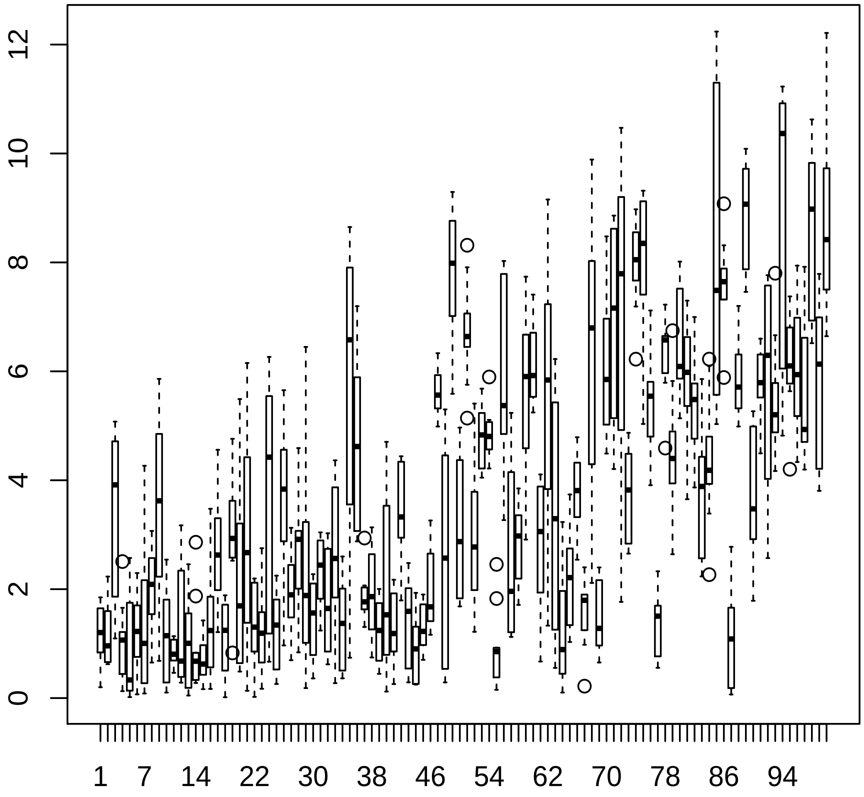

Figure 2: Sample variance explained per locus s^2. Loci are sorted in increasing order of number of causal variants.

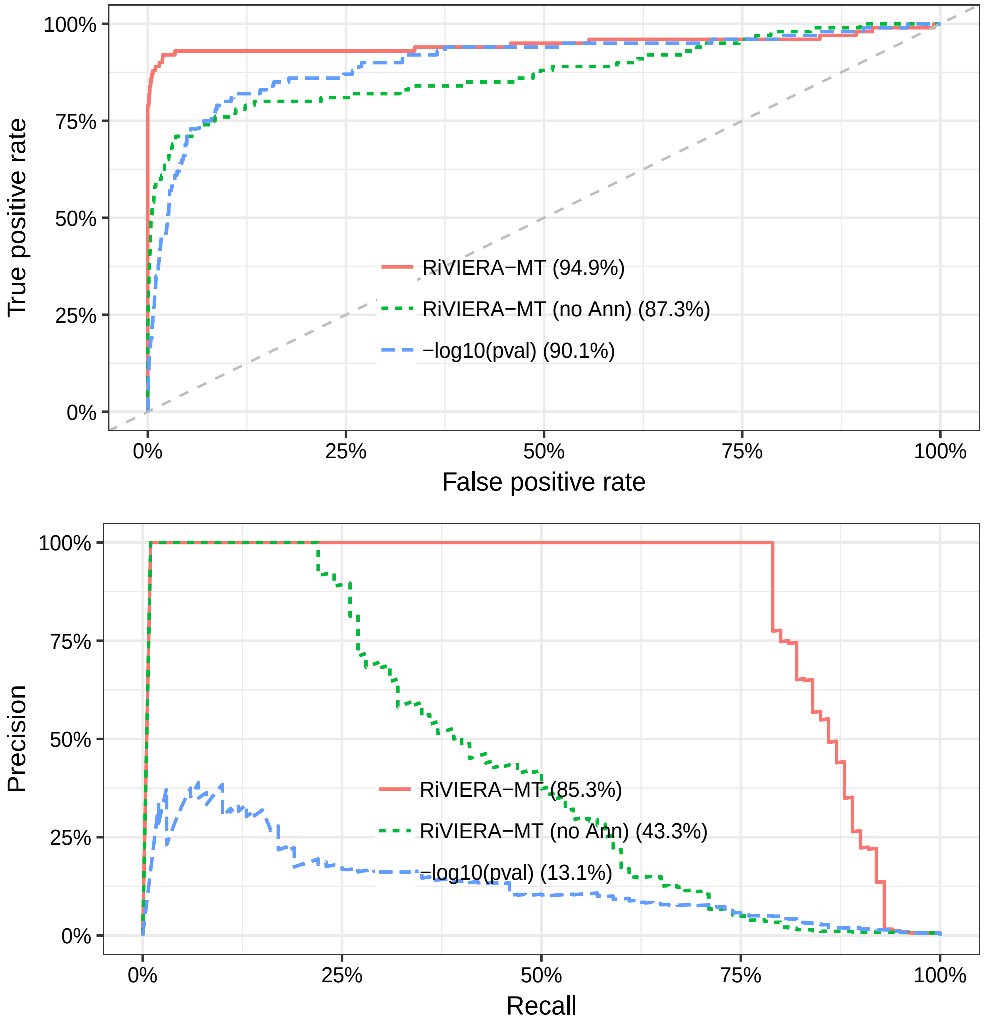

Figure 3: Linkage disequilibrium on Locus 2.

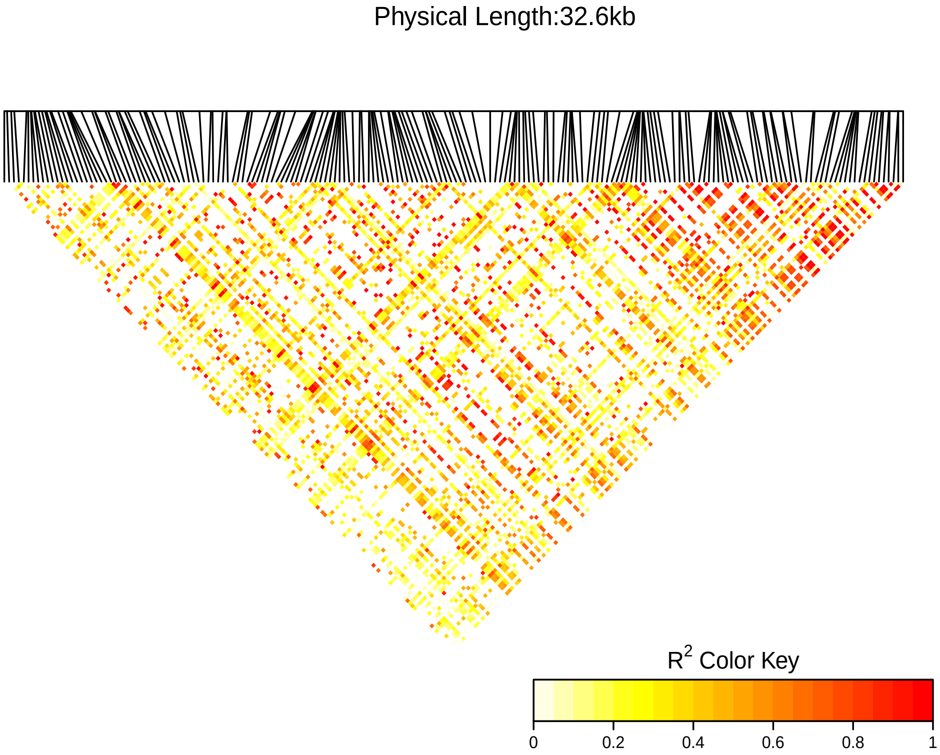

Figure 4: Linkage disequilibrium on Locus 1.

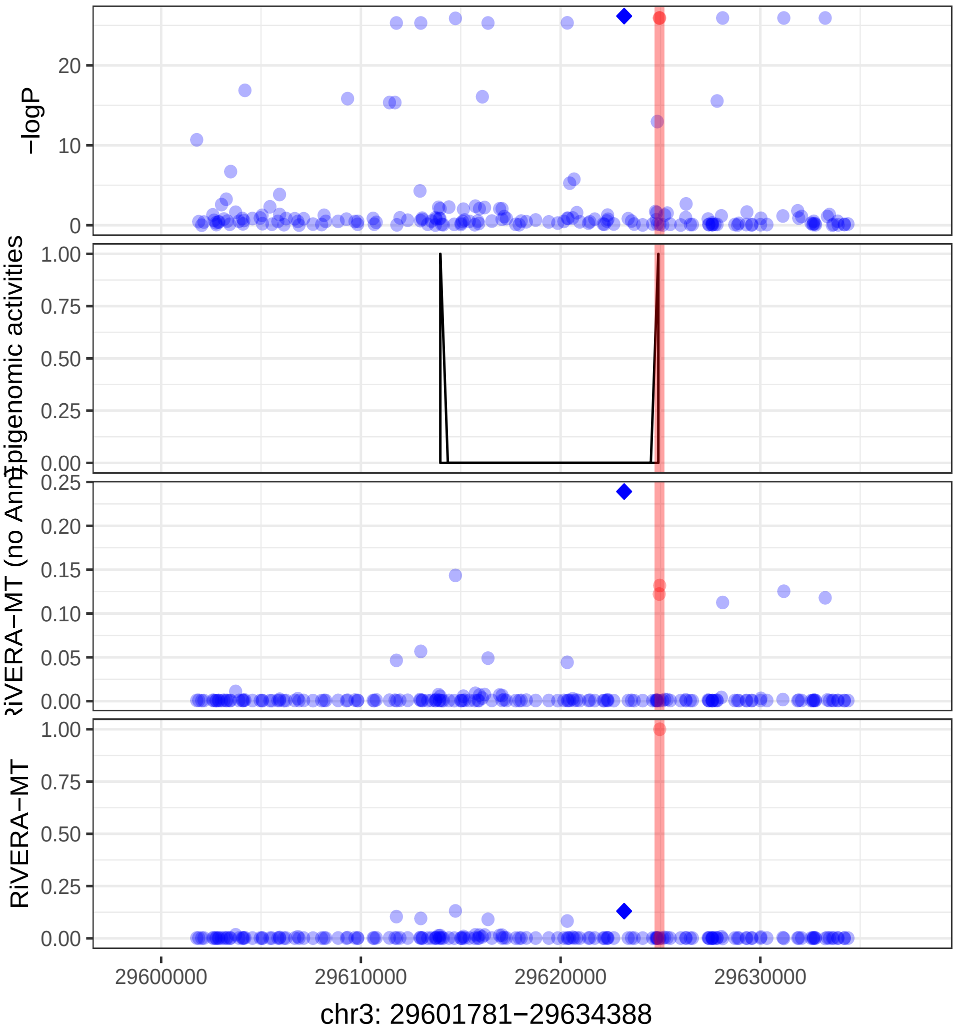

Figure 5: Fine-mapping visualization on Locus 1.

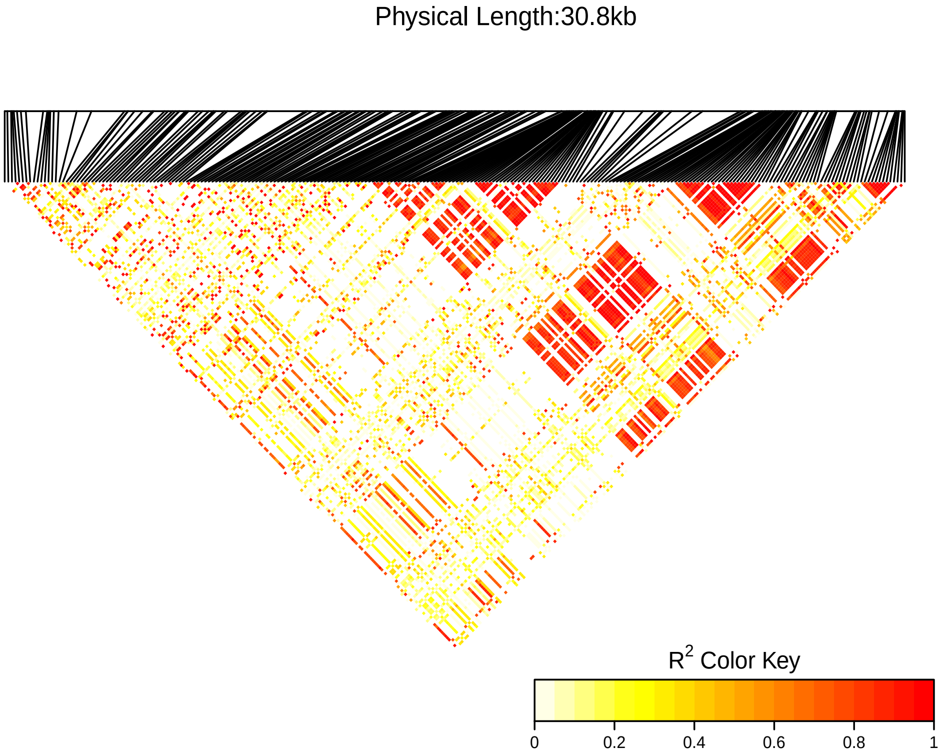

Figure 6: Linkage disequilibrium on Locus 2.

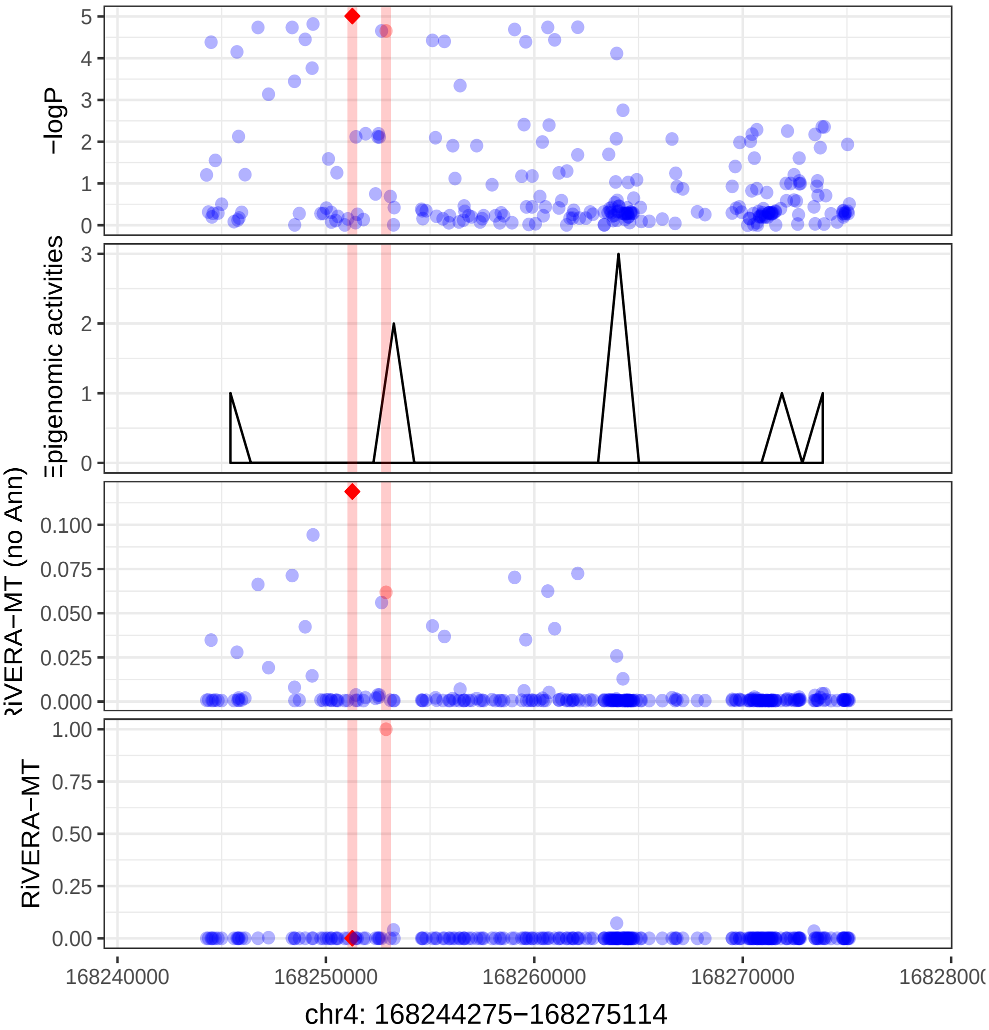

Figure 7: Fine-mapping visualization on Locus 2.

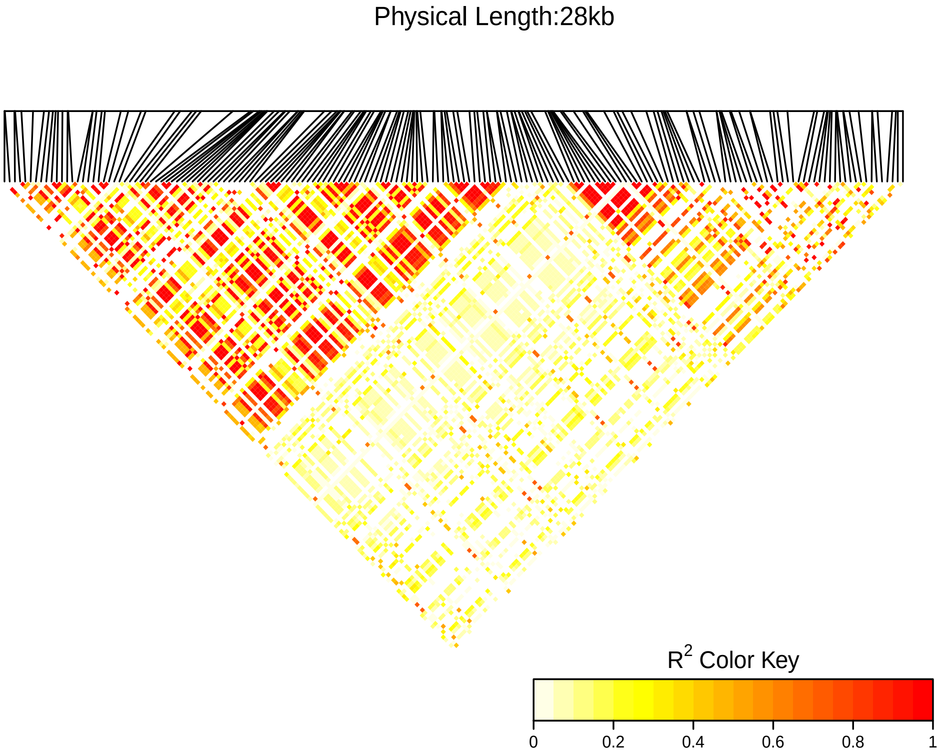

Figure 8: Linkage disequilibrium on Locus 3.

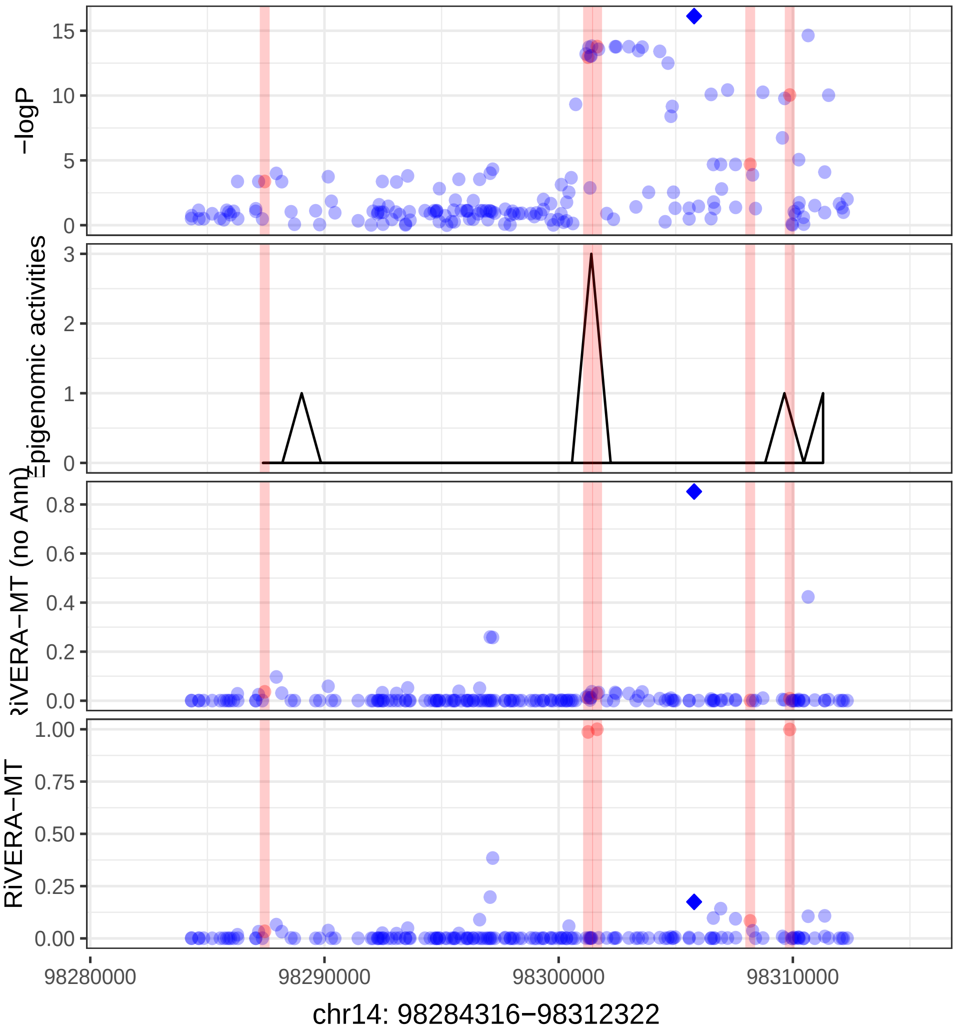

Figure 9: Fine-mapping visualization on Locus 3.

5 Multi-trait modeling

We now demonstrate the usage of inferring multiple related traits. Here we used 3 traits. The first two traits have the same causal annotations, and the third trait has its annotations randomly shuffled so it does not have any causal annotations.

Here we show that our model is able to distinguish causal annotations from non-causal one (Fig. 10) and also sample correctly the trait-by-trait correlation that indicative of the co-enrichment by jointly modeling the 3 traits (Fig. 11). Note: because we run only 20 iterations to reduce run-time installation, the results may not be as accurate as running more iterations.

```
Settings:
step = 0.01; nsteps = 100
printfreq = 1; verbose = 1
Initializing parameters ... Done!
1, lprS: 2233.77, AFS: 0 (0%), lprW: -2481.61, AFW: 1 (100%)
2, lprS: 2598.56, AFS: 0 (0%), lprW: -2477.53, AFW: 2 (100%)
3, lprS: 2680.33, AFS: 0 (0%), lprW: -2306.05, AFW: 3 (100%)
4, lprS: 2723.96, AFS: 0 (0%), lprW: -2162.81, AFW: 4 (100%)
5, lprS: 2753.24, AFS: 0 (0%), lprW: -2094.26, AFW: 5 (100%)
6, lprS: 2767.57, AFS: 0 (0%), lprW: -2031.84, AFW: 6 (100%)
7, lprS: 3047.38, AFS: 1 (14%), lprW: -2007.53, AFW: 7 (100%)
8, lprS: 3079.61, AFS: 2 (25%), lprW: -1965.36, AFW: 8 (100%)
9, lprS: 3102.8, AFS: 3 (33%), lprW: -1929.87, AFW: 9 (100%)
10, lprS: 3113.96, AFS: 4 (40%), lprW: -1883.76, AFW: 10 (100%)
11, lprS: 3119.02, AFS: 4 (36%), lprW: -1881.08, AFW: 11 (100%)
12, lprS: 3128.23, AFS: 4 (33%), lprW: -1894.43, AFW: 12 (100%)
13, lprS: 3138.62, AFS: 5 (38%), lprW: -1853.42, AFW: 13 (100%)
14, lprS: 3151, AFS: 6 (43%), lprW: -1832.97, AFW: 14 (100%)
15, lprS: 3157.62, AFS: 7 (47%), lprW: -1778.71, AFW: 15 (100%)
16, lprS: 3153.15, AFS: 8 (50%), lprW: -1753.12, AFW: 16 (100%)
17, lprS: 3168.83, AFS: 9 (53%), lprW: -1728.23, AFW: 17 (100%)
18, lprS: 3180.03, AFS: 10 (56%), lprW: -1727.18, AFW: 18 (100%)
19, lprS: 3190.53, AFS: 11 (58%), lprW: -1678.35, AFW: 19 (100%)
20, lprS: 3198.07, AFS: 12 (60%), lprW: -1667.87, AFW: 20 (100%)
MCMC inference completed.
Accepted annotation-informed models: 12
```

Sampled weights for the traits:

Average trait correlation sampled from the model (Fig. 11):

6 Session Info

```
> sessionInfo()
```

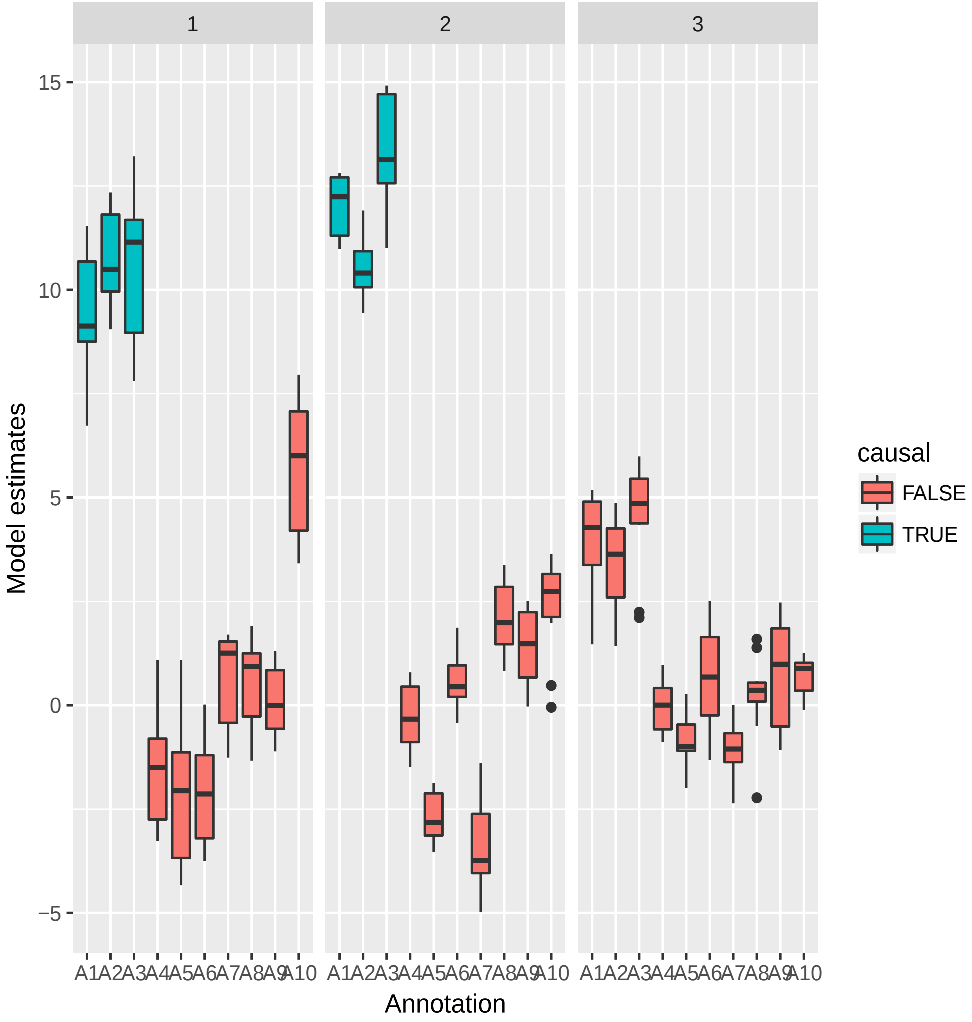

Figure 10: Enrichment weights of multiple traits.

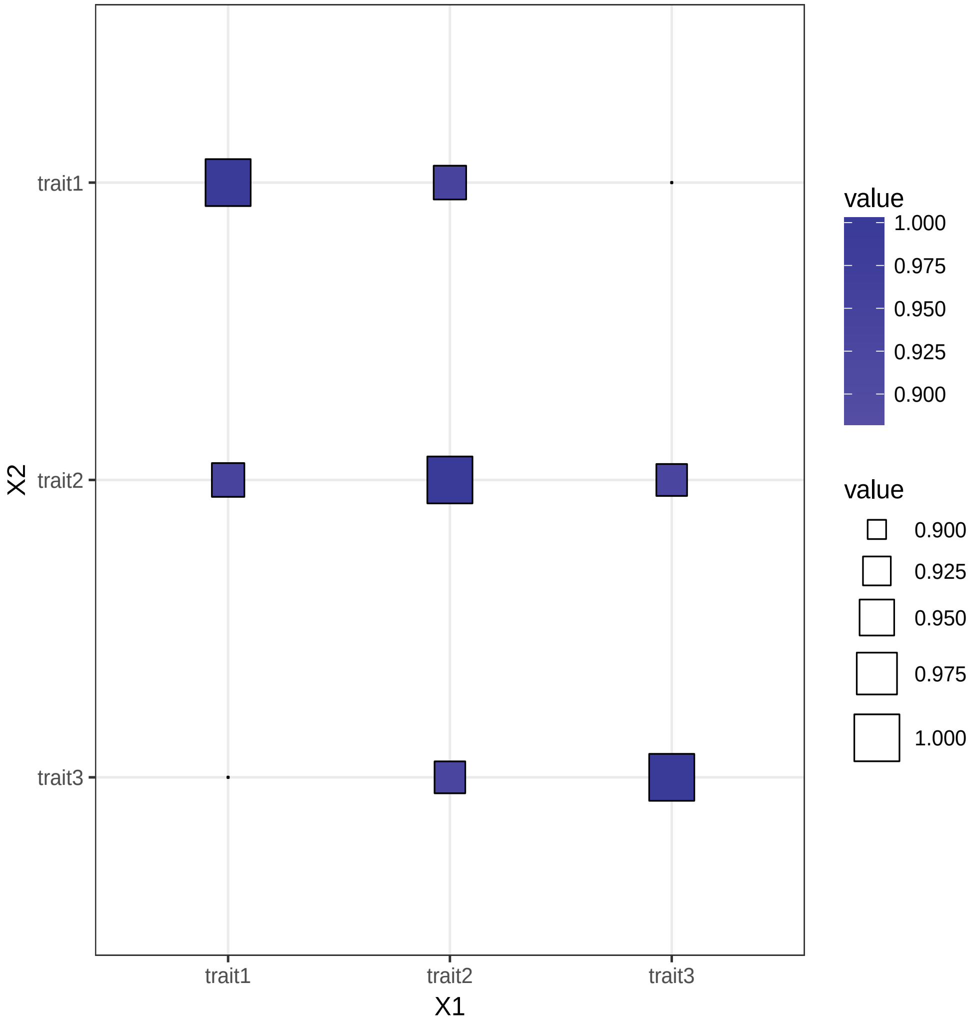

Figure 11: Averaged trait-by-trait correlation from the sampled correlation matrices.

```
R version 3.4.0 (2017-04-21)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS Sierra 10.12.5

Matrix products: default
BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0
LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapac

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
 [1] grid      stats4    parallel  stats     graphics  grDevices utils
 [8] datasets  methods   base

other attached packages:
 [1] LDheatmap_0.99-2          GenomicRanges_1.28.1
 [3] GenomeInfoDb_1.12.0       IRanges_2.10.1
 [5] S4Vectors_0.14.1          ggbio_1.24.0
 [7] BiocGenerics_0.22.0       gridExtra_2.2.1
 [9] scales_0.4.1              ROCR_1.0-7
[11] gplots_3.0.1              ggplot2_2.2.1
[13] RiVIERA_0.9.3             reshape_0.8.6
[15] RcppArmadillo_0.7.800.2.0

loaded via a namespace (and not attached):
 [1] ProtGenerics_1.8.0             bitops_1.0-6
 [3] matrixStats_0.52.2             RColorBrewer_1.1-2
 [5] httr_1.2.1                     tools_3.4.0
 [7] backports_1.0.5                R6_2.2.1
 [9] rpart_4.1-11                   KernSmooth_2.23-15
[11] Hmisc_4.0-3                    DBI_0.6-1
[13] lazyeval_0.2.0                 colorspace_1.3-2
[15] nnet_7.3-12                    GGally_1.3.0
[17] compiler_3.4.0                 graph_1.54.0
[19] Biobase_2.36.2                 htmlTable_1.9
[21] DelayedArray_0.2.2             rtracklayer_1.36.0
[23] labeling_0.3                   caTools_1.17.1
[25] checkmate_1.8.2                RBGL_1.52.0
[27] stringr_1.2.0                  digest_0.6.12
[29] Rsamtools_1.28.0               foreign_0.8-68
[31] XVector_0.16.0                 base64enc_0.1-3
[33] dichromat_2.0-0                htmltools_0.3.6
[35] ensembldb_2.0.1                BSgenome_1.44.0
[37] htmlwidgets_0.8                RSQLite_1.1-2
```

```
[39] BiocInstaller_1.26.0          shiny_1.0.3
[41] BiocParallel_1.10.1           gtools_3.5.0
[43] acepack_1.4.1                 VariantAnnotation_1.22.0
[45] RCurl_1.95-4.8                magrittr_1.5
[47] GenomeInfoDbData_0.99.0       Formula_1.2-1
[49] Matrix_1.2-10                 Rcpp_0.12.11
[51] munsell_0.4.3                 stringi_1.1.5
[53] yaml_2.1.14                   SummarizedExperiment_1.6.1
[55] zlibbioc_1.22.0               plyr_1.8.4
[57] AnnotationHub_2.8.1           gdata_2.17.0
[59] lattice_0.20-35               Biostrings_2.44.0
[61] splines_3.4.0                 GenomicFeatures_1.28.0
[63] knitr_1.15.1                  reshape2_1.4.2
[65] biomaRt_2.32.0                XML_3.98-1.7
[67] biovizBase_1.24.0             latticeExtra_0.6-28
[69] data.table_1.10.4             httpuv_1.3.3
[71] gtable_0.2.0                  mime_0.5
[73] xtable_1.8-2                  AnnotationFilter_1.0.0
[75] survival_2.41-3               tibble_1.3.0
[77] OrganismDbi_1.18.0            GenomicAlignments_1.12.1
[79] AnnotationDbi_1.38.0          memoise_1.1.0
[81] cluster_2.0.6                 interactiveDisplayBase_1.14.0
```

References

[1] Kyle Kai-How Farh, Alexander Marson, Jiang Zhu, Markus Kleinewietfeld, William J Housley, Samantha Beik, Noam Shoresh, Holly Whitton, Russell J H Ryan, Alexander A Shishkin, Meital Hatan, Marlene J Carrasco-Alfonso, Dita Mayer, C John Luckey, Nikolaos A Patsopoulos, Philip L De Jager, Vijay K Kuchroo, Charles B Epstein, Mark J Daly, David A Hafler, and Bradley E Bernstein. Genetic and epigenetic fine mapping of causal autoimmune disease variants. *Nature*, 518(7539):337–343, February 2015.

[2] Yue Li and Manolis Kellis. Riviera-mt: A bayesian model to infer risk variants in related traits using summary statistics and functional genomic annotations. *bioRxiv*, 2016.

[3] Joseph K Pickrell. Joint Analysis of Functional Genomic Dataand Genome-wide Association Studies of 18 Human Traits. *The American Journal of Human Genetics*, 94(4):559–573, April 2014.