Get started with `RegionScan`

Myriam Brossard

February 25, 2024

`RegionScan` is a R package for comprehensive and scalable region-level genome-wide association testing of alternative region-level multiple-variant and single-variant statistics and visualization of the results. It implements various state-of-the-art region-level tests to improve signal detection under heterogeneous genetic architectures and comparison of multiple-variant region-level and single-variant test results. It leverages LD-based genomic partitioning for LD-adaptive region definition. `RegionScan` is compatible with VCF input file format, and accommodates parallel region-level processing and analysis to improve computational time and memory efficiency. It also provides options for analysis of multi-allelic variants, unbalanced binary phenotypes, with detailed outputs for results interpretation, and utility functions for visualization, comparison, and interpretation.

In this vignette, we illustrate basic usage of `RegionScan` functions applied to a reproducible example of dataset provided with the package. The list of main functions and options as well as description of the outputs can be found in our preprint (https://www.biorxiv.org/content/10.1101/2024.03.04.582374v1, Supplementary Information 1 and 2).

Installation

```
library(devtools)
install_github("brossardMyriam/RegionScan")
library(RegionScan)
```

Basic usage of `regscan`

Example dataset

This example dataset is based on 436 biallelic SNPs (MAF>0.05) genotyped in chr16:46382489-47684754 in 40,000 individuals simulated from high coverage whole genome sequenced 1000G European ancestry haplotypes using HAPGEN2 software. For this illustration, we used 20 consecutive regions identified by LD partitioning of chr16 using the BigLD algorithm (https://pubmed.ncbi.nlm.nih.gov/29028986/) implemented in R package *gpart* (https://academic.oup.com/bioinformatics/article/35/21/4419/5487391). However, *RegionScan* accepts any type of user-defined region boundaries (e.g. gene start/end positions etc). We simulated a quantitative trait (*sim_QT* in the `phenocov` input) and a binary trait (*sim_bin*) ; sim_QT was generated under a linear regression model, assuming joint effects of two causal SNPs in region 8 ("chr16.46880510.A.G", "chr16.46889594.G.A"); while *sim_bin* was generated by dichotomizing *sim_QT* (see data_simulation.R for details).

In this vignette, we illustrate how to run the main function *regscan* for region-level and single-SNP analysis and how to visualize the results.

Main inputs

Three main inputs required by regscan

Input 1: REGIONinfo This dataframe must include at least `chr`, `start.bp`, `end.bp`, `region`; regions start and end positions.

```
head(REGIONinfo)
#>   chr start.index end.index         start.rsID           end.rsID start.bp
#> 1  16           4        14 chr16.46382489.T.A chr16.46401970.C.T 46382489
#> 2  16          16       175 chr16.46402078.C.G chr16.46490675.G.A 46402078
#> 3  16         187       230 chr16.46505480.G.A chr16.46568409.T.A 46505480
#> 4  16         235       271 chr16.46568907.C.T chr16.46614446.G.A 46568907
#> 5  16         289       293 chr16.46633030.A.T chr16.46641133.G.A 46633030
#> 6  16         298       439 chr16.46651722.T.A chr16.46793612.G.A 46651722
#>     end.bp region
#> 1 46401970      1
#> 2 46490675      2
#> 3 46568409      3
#> 4 46614446      4
#> 5 46641133      5
#> 6 46793612      6
```

Input 2: geno This dataframe must include genotypes (columns) of the individuals (rows) ; as illustrated below for 4 genetic variants. The individuals must be in the same order as the individuals in input `phenocov`

```
head(geno[,1:4])
#>   chr16.46382489.T.A chr16.46401970.C.T chr16.46402078.C.G chr16.46402735.C.T
#> 1                  0                  0                  0                  0
#> 2                  0                  1                  0                  1
#> 3                  1                  1                  0                  1
#> 4                  0                  0                  1                  0
#> 5                  0                  0                  0                  0
#> 6                  0                  0                  1                  0
```

Input 3: SNPinfo This dataframe includes the variant positions and information for the variants in `geno` input ; all the following columns are required

```
head(SNPinfo)
#>      chr       bp            variant multiallelic    ref    alt      maf
#>    <int>    <int>             <char>        <num> <char> <char>    <num>
#> 1:    16 46382489 chr16.46382489.T.A            0      T      A 0.132125
#> 2:    16 46401970 chr16.46401970.C.T            0      C      T 0.148238
#> 3:    16 46402078 chr16.46402078.C.G            0      C      G 0.481212
#> 4:    16 46402735 chr16.46402735.C.T            0      C      T 0.147688
#> 5:    16 46404976 chr16.46404976.G.C            0      G      C 0.054150
#> 6:    16 46405393 chr16.46405393.C.A            0      C      A 0.110875
```

Input 4: phenocov This dataframe must includes phenotypes and covariates for all the individuals of input `geno`. The individuals must be in the same order as in input 'geno.

```
head(phenocov)
#>   ID   sim_QT sim_bin
#> 1  1 3.192062       1
#> 2  2 2.367709       1
#> 3  3 3.814162       1
#> 4  4 2.656099       1
#> 5  5 3.343224       1
#> 6  6 2.705045       1
```

Single & Region-level analysis

Example of run of `regscan` main function for region & single-SNP level analysis with the continuous outcome `sim_QT` .

```
results<-RegionScan::regscan(phenocov = phenocov, pheno="sim_QT",  REGIONinfo=REGIONinfo,
                             geno_type="D", pheno_type="C", data = geno,  SNPinfo = SNPinfo )
```

Main Outputs

The main output of `regscan` function is a list including the region-level, Bin-level, variant-level outputs and filtered variant lists; and optionally an additional output (sinleSNPall) with single-SNP results for all the variants analyzed (including filtered variants in region-level analysis).

Output 1: Region-level results This is the main output which includes results from all the region-level test implemented in RegionScan for all regions from REGIONinfo.

```
row.names(results$regionout)<-NULL
head(results$regionout)
#>   chr region start.bp   end.bp nSNPs nSNPs.kept       maxVIF         Wald
#> 1  16      1 46382489 46401970     2          2 7.631184.... 8.578549....
#> 2  16      2 46402078 46490675    33         29 89.30684.... 78.06813....
#> 3  16      3 46505480 46568409    16          9 36.15805.... 55.18799....
#> 4  16      4 46568907 46614446    11          7 38.44783.... 54.27399....
#> 5  16      5 46633030 46641133     2          1            1 9.499954....
#> 6  16      6 46651722 46793612    79         35 180.7569.... 89.59961....
#>   Wald.df       Wald.p         MLCB MLCB.df       MLCB.p          LCB LCB.df
#> 1       2 0.013714.... 7.657003....       1 0.005655.... 7.657003....      1
#> 2      29 2.213318.... 57.85440....       9 3.466171.... 26.03930....      1
#> 3       9 1.121493.... 52.59930....       4 1.033627.... 7.999335....      1
#> 4       7 2.075915.... 44.62355....       4 4.760959.... 39.11439....      1
#> 5       1 0.002054.... 9.499954....       1 0.002054.... 9.499954....      1
#> 6      35 1.118290.... 61.90856....       8 1.965451.... 23.87505....      1
#>          LCB.p         PC80 PC80.df       PC80.p       SKAT.p      SKATO.p
#> 1 0.005655.... 7.633874....       1 0.005728.... 0.005596.... 0.005781....
#> 2 3.345371.... 56.24105....       5 7.248759.... 6.140714.... 2.317563....
#> 3 0.004679.... 46.93972....       3 3.579744.... 5.750429.... 2.366122....
#> 4 3.996853.... 43.81066....       3 1.655610.... 5.427249.... 9.882212....
#> 5 0.002054.... 9.499954....       1 0.002054.... 0.002057.... 0.002057....
#> 6 1.027953.... 60.72948....       4 2.037991.... 8.985011.... 7.432988....
#>      simes.p.p simes.p.Meff   simpleM.df    simpleM.p      GATES.p      uMinP.p
#> 1 0.003797.... 1.067820....            2 0.007579.... 0.004687.... 0.003797....
```

```
#> 2 1.687836.... 9.910942....           22 9.594103.... 6.335776.... 4.361016....
#> 3 0.000433.... 4.783002....            8 0.001138.... 0.000592.... 0.000142....
#> 4 1.363957.... 3.747198....            6 5.364175.... 4.595361.... 8.940270....
#> 5 0.002056....           1 0.002056.... 0.002056.... 0.002056....           16
#> 6 3.324259.... 10.58963....           22 3.834710.... 2.922570.... 1.748155....
```

Output 2: Bin-level results This output includes the bin-level association results (deltaB, deltaB.se, deltaB.pvalue) for each LD bin identified in the regions for the MLC test. It also includes the bin sizes (ie. number of SNPs in each LD bin before and after LD-based pruning. The bin-level results can help to investigate some MLC region-level test results.

```
row.names(results$binout)<-NULL
head(results$binout)
#>   chr region start.bp   end.bp binstart.bfP.bp binend.bfP.bp binstart.afP.bp
#> 1  16      1 46382489 46401970        46382489      46401970        46382489
#> 2  16      2 46402078 46490675        46402735      46466492        46402735
#> 3  16      2 46402078 46490675        46402078      46490675        46402078
#> 4  16      2 46402078 46490675        46405393      46467231        46405393
#> 5  16      2 46402078 46490675        46411236      46461260        46411236
#> 6  16      2 46402078 46490675        46459836      46460534        46459836
#>   binend.afP.bp binsize.bfP binsize.afP bin NSNPs.kept               deltaB
#> 1      46401970           2           2   1          2  -0.0215576196362694
#> 2      46466492           9           6   1          6 -0.00695681158277963
#> 3      46490675           6           6   2          6  0.00612789212287157
#> 4      46467231           6           5   3          5 -0.00609765307800171
#> 5      46461260           6           6   4          6 -0.00825024396270651
#> 6      46460534           2           2   5          2 -0.00359760118888102
#>              deltaB.se       deltaB.pvalue
#> 1 0.00779060830215809 0.00565523397003551
#> 2 0.00412151111104078  0.0914251638747339
#> 3 0.00381185505383036   0.107925460468975
#> 4 0.00614512191644068   0.321063218322989
#> 5 0.00437189961499551  0.0591458862943667
#> 6  0.011676307491812   0.757997714217638
```

Output 3: variant-level results This output provides detailed results at the variant-level for all variants kept for the region-level analysis. The variant-specific results from multiple regression at the region level and single-SNP regression models are reported. Investigation of this output can help to investigate the region-level results (particularly for the MLC region-level test).

```
row.names(results$snpout)<-NULL
head(results$snpout)
#>   chr region start.bp   end.bp bin       bp multiallelic ref alt       maf
#> 1  16      1 46382489 46401970   1 46382489            0   T   A 0.1321250
#> 2  16      1 46382489 46401970   1 46401970            0   C   T 0.1482375
#> 3  16      2 46402078 46490675   1 46402735            0   C   T 0.1476875
#> 4  16      2 46402078 46490675   1 46407733            0   G   A 0.1472250
#> 5  16      2 46402078 46490675   1 46424244            0   T   C 0.1471250
#> 6  16      2 46402078 46490675   1 46452260            0   C   T 0.1516500
#>   MLC.codechange LC.codechange           variant           sglm.beta
#> 1              0             0 chr16.46382489.T.A -0.0397690215740153
#> 2              0             0 chr16.46401970.C.T -0.0433298393987113
```

```
#> 3              0             1 chr16.46402735.C.T -0.0435931113805227
#> 4              0             1 chr16.46407733.G.A -0.041954008635747 4
#> 5              0             1 chr16.46424244.T.C -0.0472985035084181
#> 6              0             1 chr16.46452260.C.T -0.0477073817009268
#>              sglm.se          sglm.pvalue  mglm.vif   mglm.beta    mglm.se
#> 1 0.0156776073126558  0.0111947922846746  7.631184  0.01933877 0.04330814
#> 2 0.0149686516864239   0.003797194991454  7.631184 -0.06054226 0.04135071
#> 3 0.0149849003574682  0.0036262880244314 45.566160 -0.12919510 0.10109958
#> 4 0.0149992673531146 0.00515928997059794 69.116155  0.24487683 0.12463231
#> 5 0.0150117535272298 0.00162963605341296 41.839892 -0.10999110 0.09705300
#> 6 0.0148273882560012 0.00129408055605998 89.306848  0.08655029 0.14005273
#>   mglm.pvalue
#> 1  0.65521047
#> 2  0.14317001
#> 3  0.20129230
#> 4  0.04944499
#> 5  0.25709103
#> 6  0.53658967
```

Output 4: variants excluded This output lists all the variants excluded within each region and the reasons of exclusion (MAF, LD pruning etc); the LD bin information is also reported.

```
row.names(results$filterout)<-NULL
head(results$filterout)
#>   chr region    start      end       bp            variant bin reason
#> 1  16      2 46402078 46490675 46418341 chr16.46418341.G.A   1   rcut
#> 2  16      2 46402078 46490675 46435570 chr16.46435570.C.T   1   rcut
#> 3  16      2 46402078 46490675 46439309 chr16.46439309.G.A   1   rcut
#> 4  16      2 46402078 46490675 46444013 chr16.46444013.T.C   3   rcut
#> 5  16      3 46505480 46568409 46539340 chr16.46539340.T.C   1   rcut
#> 6  16      3 46505480 46568409 46539341 chr16.46539341.G.A   1   rcut
```

Visualisation of the results

Locus plot Illustration of the LocusPlot function which plots the region-level results in a set of consecutive regions. The plot is directly saved as a pdf in the local directory.

```
results$regionout$chr<-as.numeric(as.character(results$regionout$chr))
LocusPlot(chr=16,pheno="sim_QT",regscanout=results,regionlist=c(1:20),outname="LocusPlot",
          region_tests=c("Wald.p","PC80.p","MLCB.p","SKATO.p"))
```

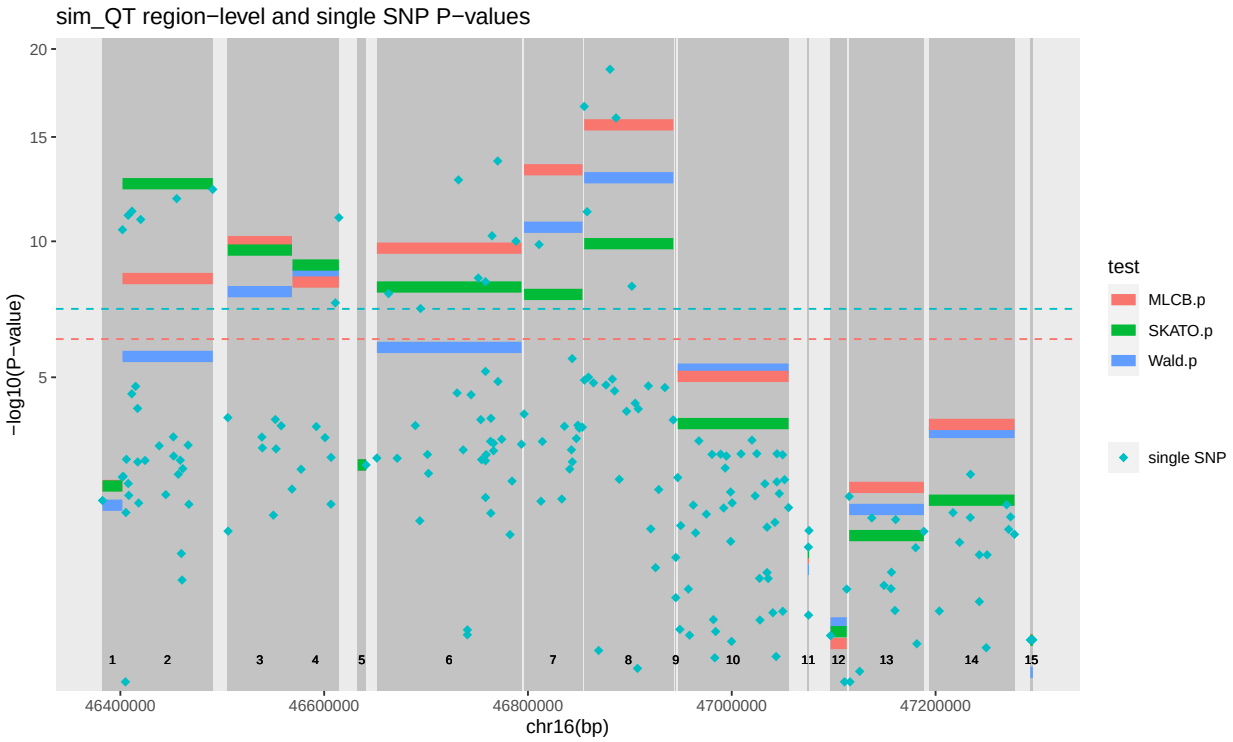

Figure 1: Locus plot of consecutive regions

LD heatmaps The following function produces LD heatmaps for a specified region (here region "8") to visualize the correlation structure within region, before and after pruning, as well as the dependencies between the LD bins. Plots are saved as pdf files in the local directory.

```
regscan(phenocov = phenocov, pheno="sim_QT",  REGIONinfo=REGIONinfo,
        geno_type="D", pheno_type="C", data = geno,  SNPinfo = SNPinfo,
        MLCheatmap = TRUE, regionlist ="8" )
```

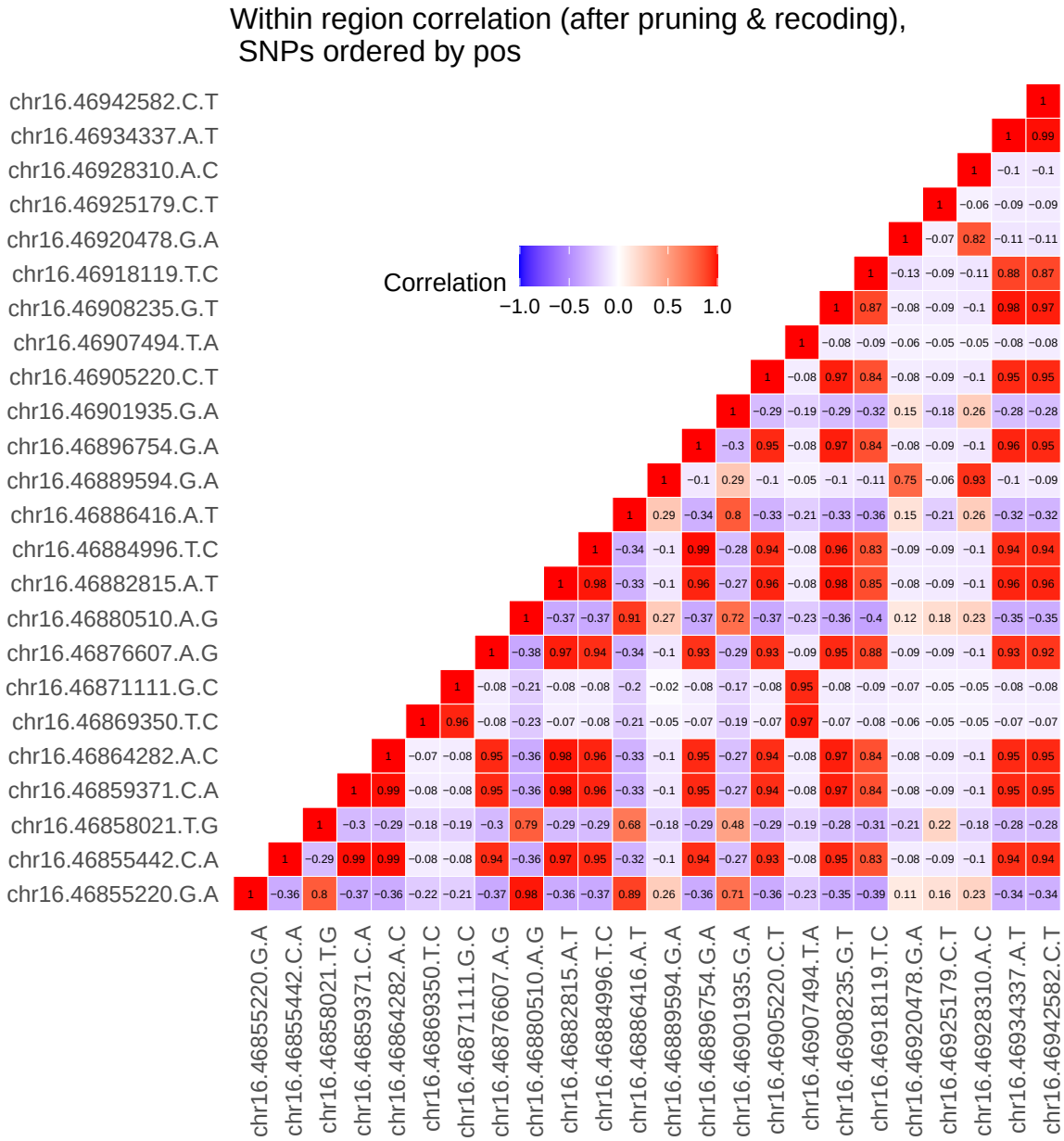

Figure 2: Example 1 of LD heatmap plot for region 8 (after LD pruning), ordered by variant positions

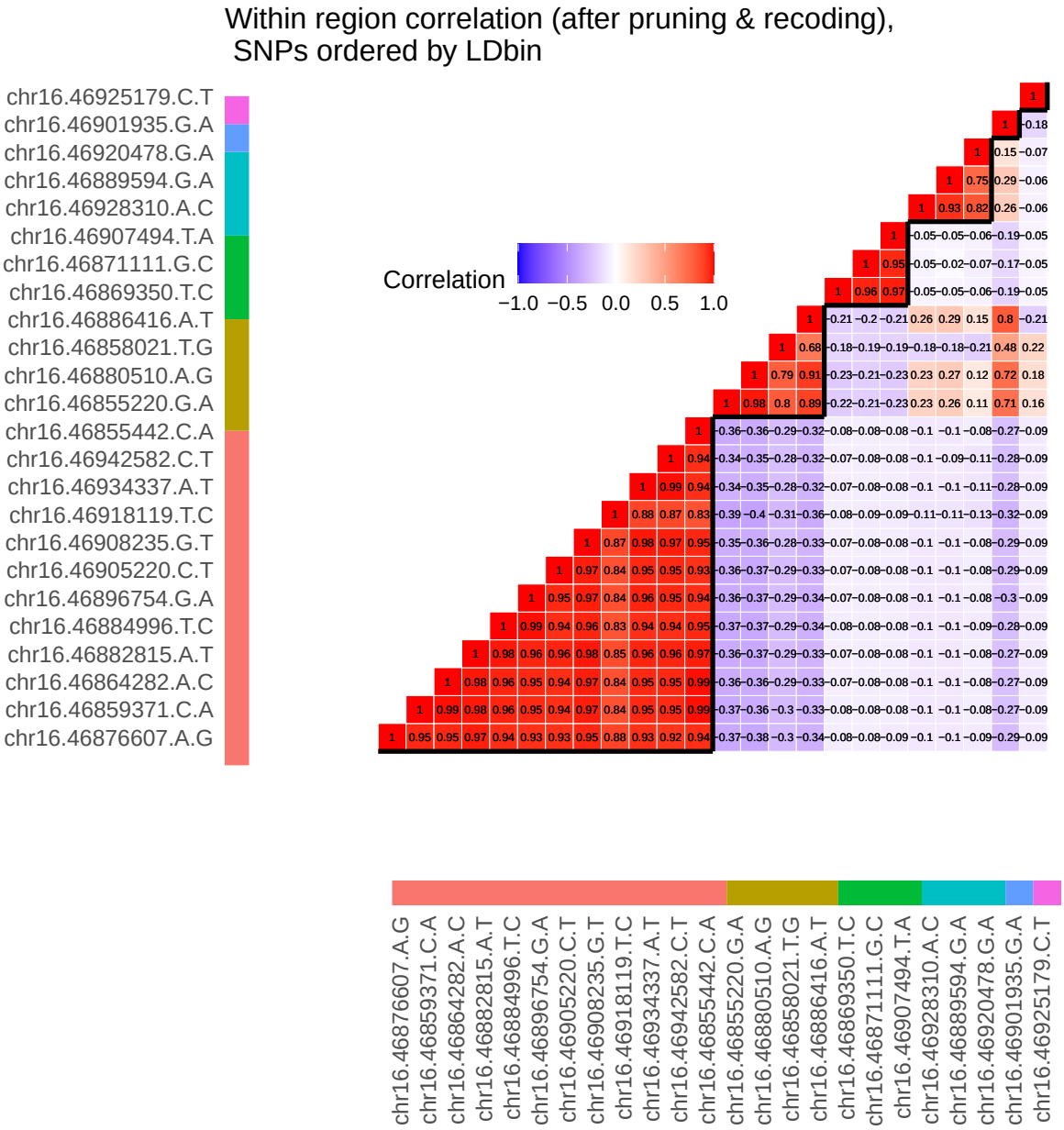

Figure 3: Example 2 of LD heatmap plot for region 8 (after LD pruning), ordered by LD bins

SNP LD bin positions Visualization of the variant positions (X axis) along the LD bins (Y axis) for the variants kept after LD pruning (and in grey, removed by LD pruning). Plots are saved as pdf files in the local directory.

```
MLCbinsnpPlot(rscanout = results, chr_=16, region_ =8 )
```

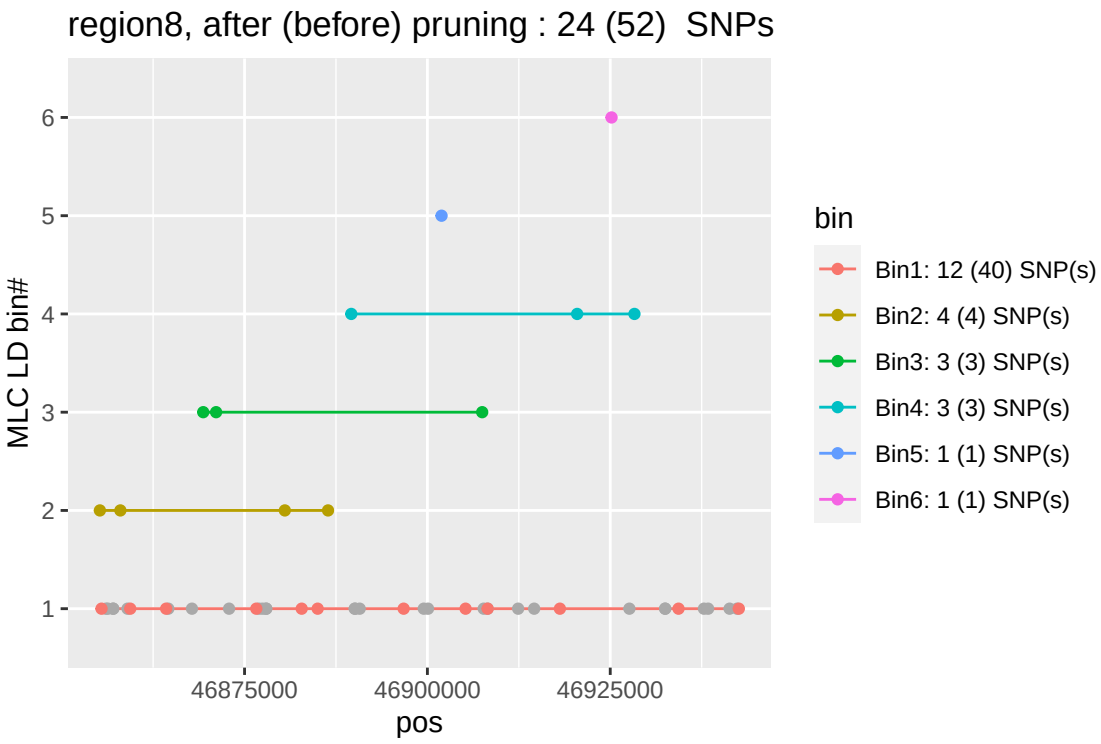

Figure 4: variant positions (X axis) along LD bins (Y axis) in region 8

Citation

RegionScan : A comprehensive R package for region-level genome-wide association testing with integration and visualization of multiple-variant and single-variant hypothesis testing. Brossard M, Roshandel D, Luo K, Yavartanoo F, Paterson AD, Yoo YJ, Bull SB. https://www.biorxiv.org/content/10.1101/2024.03.04.582374v1.

License

This package is released under the GNU General Public License (GPL) v3.0.