Get started with `RegionScan`

Myriam Brossard

December 1st, 2024

`RegionScan` is designed for scalable genome-wide association testing of both region-level multiple-variant and single-variant statistics, with visualization of the results. For detection of association under various regional architectures, it implements three classes of state-of-the-art region-level tests, including multiple-variant linear/logistic regression (with and without dimension reduction), variance-component score tests, and region-level minP tests. `RegionScan` also supports the analysis of multi-allelic variants and unbalanced binary phenotypes and is compatible with widely used variant call format (VCF) files for both genotyped and imputed variants. Association testing leverages linkage disequilibrium (LD) structure in pre-defined regions, for example, LD-adaptive regions obtained by genomic partitioning, and accommodates parallel processing to improve computational and memory efficiency. Detailed outputs (with allele frequencies, variant-LD bin assignment, single/joint variant effect estimates and region-level results) and utility functions are provided to assist comparison, visualization, and interpretation of results. Thus, `RegionScan` analysis offers valuable insights into region-level genetic architecture which supports a wide range of potential applications.

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 implemented in `RegionScan`, options as well as description of the outputs of main function regscan are in Annex of this vignette.

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
#> 4   16 46382489 chr16.46382489.T.A            0   T   A 0.132125
#> 14  16 46401970 chr16.46401970.C.T            0   C   T 0.148238
#> 16  16 46402078 chr16.46402078.C.G            0   C   G 0.481212
#> 19  16 46402735 chr16.46402735.C.T            0   C   T 0.147688
#> 35  16 46404976 chr16.46404976.G.C            0   G   C 0.054150
#> 38  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 -0.8418605       0
#> 2  2  2.5931863       1
#> 3  3  4.9720520       1
#> 4  4  3.4298926       1
#> 5  5  3.5351976       1
#> 6  6  3.3847384       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<-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.

```
head(results$regionout)
#>   chr region start.bp   end.bp nSNPs nSNPs.kept           maxVIF
#> 1  16      1 46382489 46401970     2          2 7.63118409358028
#> 2  16      2 46402078 46490675    33         29 89.3068478742558
#> 3  16      3 46505480 46568409    16          9  36.158052040736
#> 4  16      4 46568907 46614446    11          7 38.4478358507763
#> 5  16      5 46633030 46641133     2          1             <NA>
#> 6  16      6 46651722 46793612    79         35 180.756972532698
#>                Wald Wald.df              Wald.p             PC80 PC80.df
#> 1 13.2183097619898       2  0.00134797086396498 11.4158641131233       1
#> 2 126.944786126531      29 3.21618445117108e-14 84.3542012194867       5
#> 3 58.5717069136177       9 2.52433174409188e-09 28.5345978528676       3
#> 4 102.804756484661       7 2.84007861980665e-19 95.8824234218916       3
#> 5 11.1273685632092       1 0.000850631832676347 11.1273685632029       1
#> 6 136.694141275958      35 5.95571393547455e-14 84.6911418092466       4
#>                  PC80.p             MLCB MLCB.df              MLCB.p
#> 1 0.000728196198525501 11.4552092034544       1 0.000712938624286533
#> 2 1.02813148101739e-16 87.1426177690514       9 6.08481865525033e-15
#> 3 2.80483261162339e-06 36.2666724304519       4 2.55025052900061e-07
#> 4 1.19307154961529e-20 97.8991119543123       4 2.75428945545249e-20
#> 5 0.000850631832679229 11.1273685632092       1 0.000850631832676347
#> 6 1.76399337813415e-17 99.3399418631948       8 5.82398514579283e-18
#>                LCB LCB.df              LCB.p              SKAT.p
#> 1 11.4552092034544      1 0.000712938624286533  0.00104947556871415
#> 2 34.9763378622398      1 3.33736410188684e-09 9.74634425131443e-15
#> 3 7.08311103881546      1  0.00778137995814597    0.215689691960729
#> 4 48.6354481085035      1  3.0824701317004e-12 3.03026009628285e-17
#> 5 11.1273685632092      1 0.000850631832676347  0.000851924538777572
#> 6 22.1885012754748      1 2.47149570476665e-06 8.42588081355409e-18
#>               SKATO.p              simes.p simpleM.df           simpleM.p
#> 1 0.000895448518125508 0.000359829455696319          2 0.000719529434155541
#> 2  6.8224409759201e-14 1.19993233644567e-13         22 6.81454892514921e-13
#> 3   0.0534490049341949 0.000171294066412414          8 0.000318336642977646
#> 4 2.12118206739799e-16 6.22561765526543e-17          6                   0
#> 5  0.000851924538777572 0.000851405665034688          1 0.000851405665034688
#> 6 5.89811656948778e-17 1.08416039850315e-16         22 2.44249065417534e-15
#>               GATES.p        single_Wald.p
#> 1 0.000444211392039609 0.000359829455696319
```

```
#> 2  4.5042899698509e-13 3.10037494646998e-14
#> 3 0.00024984476956063 1 3.97976234088524e-05
#> 4 2.80476976663764e-16 5.45667595824671e-17
#> 5 0.00085140566503468 8 0.00085140566503468
#> 6 1.81250162120005e-15 1.08416039850315e-16
```

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.

```
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  -0.0263044227412342
#> 2      46466492           9           6   1          6  -0.0024261415198229
#> 3      46490675           6           6   2          6  0.00857041275832189
#> 4      46467231           6           5   3          5 -0.00192016759715904
#> 5      46461260           6           6   4          6 -0.00109835921410416
#> 6      46460534           2           2   5          2   0.0642642715636263
#>             deltaB.se        deltaB.pvalue
#> 1 0.00777190105679396 0.000712938624286533
#> 2 0.00410934586819968    0.554925166125585
#> 3 0.00380060380613128   0.0241324836127966
#> 4 0.00612698369034169    0.753980391088406
#> 5 0.00435899531386413    0.801060144650654
#> 6  0.0116418431625167 3.38784749048239e-08
```

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).

```
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 -0.0480730798258817
#> 2              0             0 chr16.46401970.C.T -0.0532834375150429
#> 3              0             1 chr16.46402735.C.T -0.0524756816395952
#> 4              0             1 chr16.46407733.G.A -0.0570326612654028
#> 5              0             1 chr16.46424244.T.C  -0.051518678296245
#> 6              0             1 chr16.46452260.C.T -0.0531965054431854
```

```
#>              sglm.se          sglm.pvalue  mglm.vif   mglm.beta    mglm.se
#> 1  0.015640279358937  0.00211584432310794  7.631184  0.03012700 0.04320415
#> 2 0.0149327616684696 0.000359829455696319  7.631184 -0.08009788 0.04125142
#> 3 0.0149490636258694 0.000448095479251166 45.566160  0.07890890 0.10080117
#> 4 0.0149628641470136 0.000138269175752732 69.116155 -0.18959396 0.12426444
#> 5 0.0149762180348197 0.000582237338420528 41.839892  0.12426043 0.09676653
#> 6 0.0147921645682756 0.000323210773092559 89.306848 -0.28737353 0.13963935
#>   mglm.pvalue
#> 1  0.48560835
#> 2  0.05218043
#> 3  0.43373903
#> 4  0.12708509
#> 5  0.19910525
#> 6  0.03959952
```

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.

```
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.

```
LocusPlot(chr=16,pheno="sim_QT",regscanout=results,regionlist=c(1:15),outname="LocusPlot_region1_15",
          region_tests=c("Wald.p","PC80.p","MLCB.p","SKATO.p"))
```

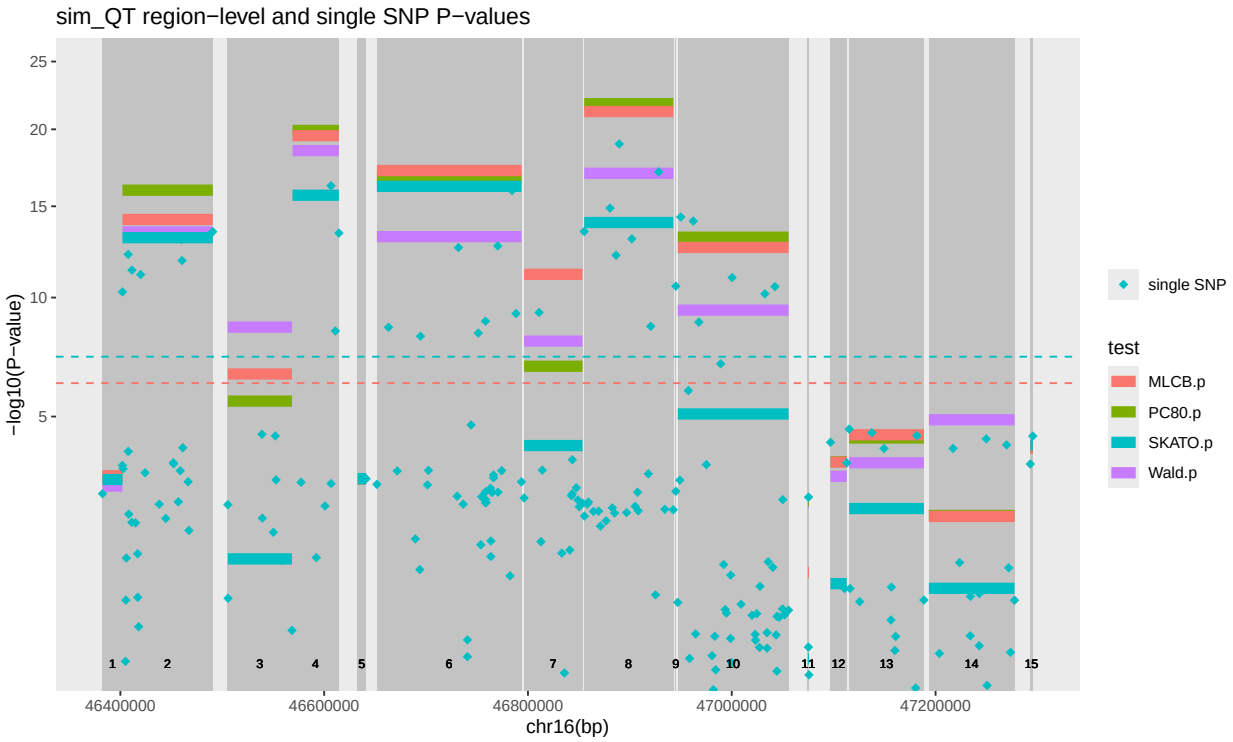

Figure 1: Locus plot of consecutive regions

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" )
```

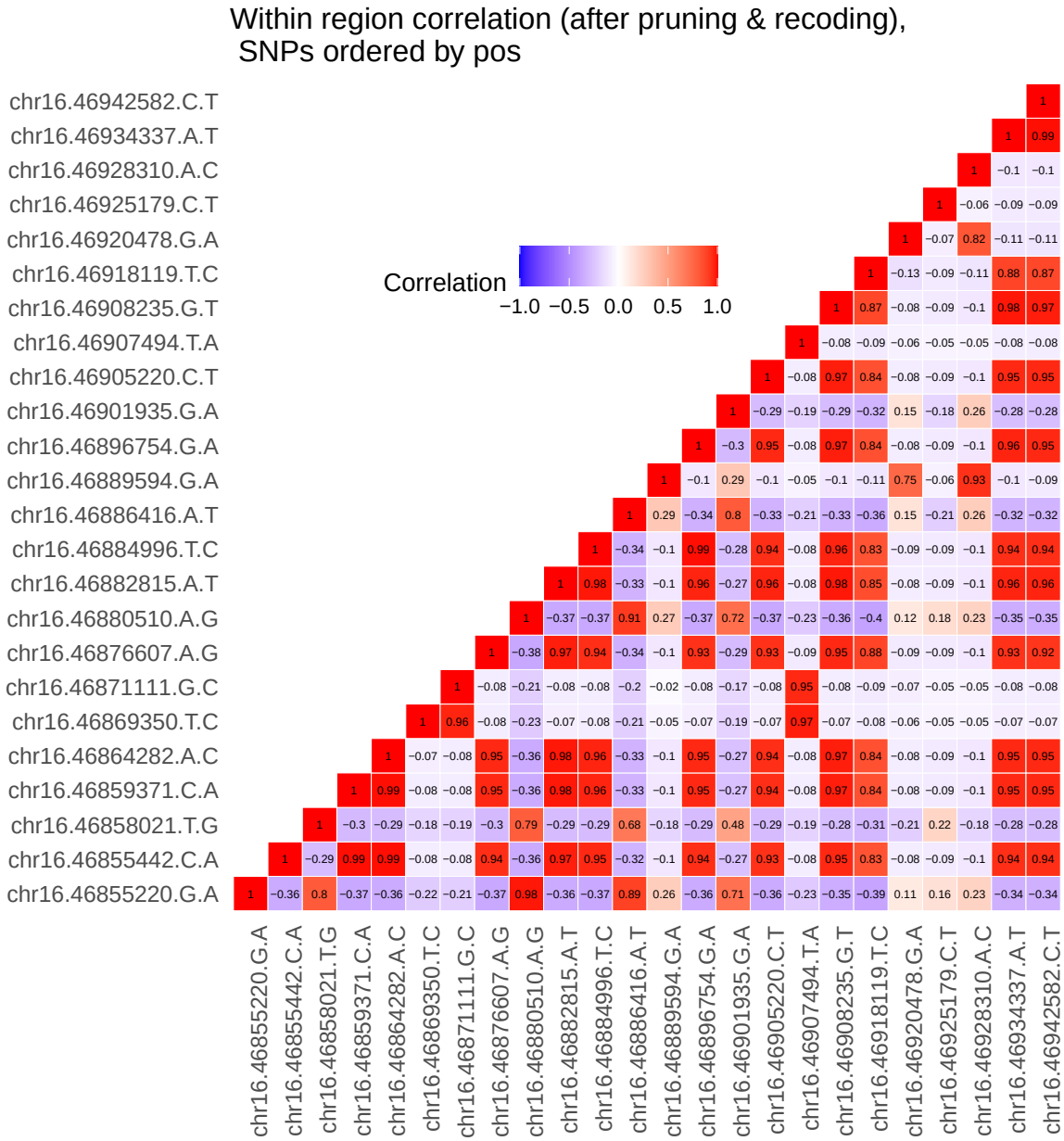

Figure 2: Example 1 of LD heatmap plot for region 8 (after LD pruning), ordered by variant positions

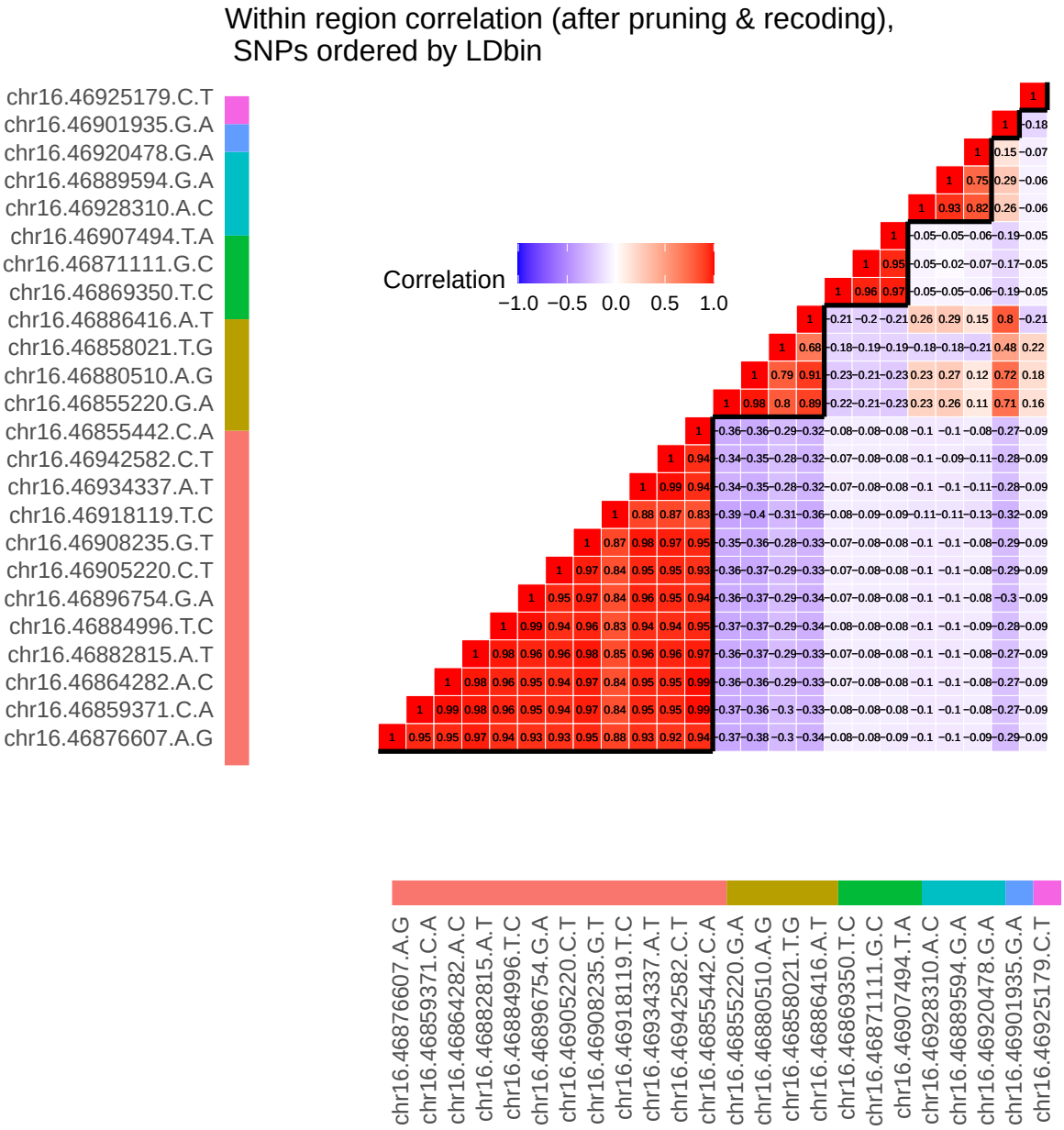

Figure 3: Example 2 of LD heatmap plot for region 8 (after LD pruning), ordered by LD bins

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 )
```

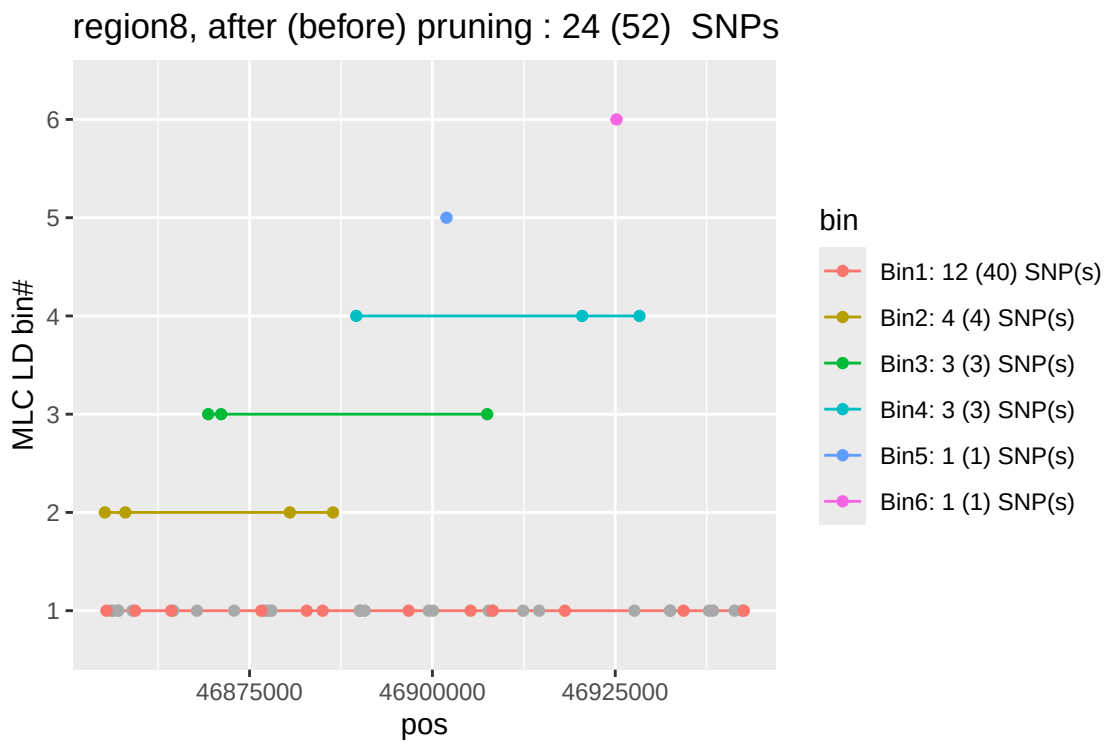

Figure 4: variant positions (X axis) along LD bins (Y axis) in region 8

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.

ANNEX

Table 1. List of *Main* and *Utility* Functions in RegionScan

Functions	Short description
regscan	Main function for region processing, single-SNP and region-level analysis.
recodeVCF	Extracts information from VCF file for each region, recode variants (including multiallelic variants) and returns genotype and info files used by *regscan.*
MiamiPlot	Produces a Miami plot genome-wide (or in a subset of regions) for any specified pair of tests; can also be used to visualize potential colocalization between region-level tests obtained for two independent studies.
qqregscanPlot	Creates a Quantile-Quantile plot for a specified region-level test and calculates genomic inflation factor.
LocusPlot	Plots region-level test results for a specified set of contiguous regions.
MLCbinsnpPlot	Visualization of genomic positions of the variants assigned to within-region LD bins.

Table 2. List of the main arguments of *regscan*

Arguments	Definition	Default value
REGIONinfo	dataframe including region positions. This dataframe must include the four following columns: "chr", "region", "start.bp", "end.bp". This file can include LD block regions positions as generated using for example, by the "BigLD" function from the *gpart* R package or any type of genes/regions positions.	Required
phenocov	dataframe including the covariates (if applicable) and the phenotype (s) in columns. Individuals (in rows) must be in the same order as in the *data* input.	
covlist	list of covariates to account for in single-SNP and multiple-SNP regression models, including for example, ancestry PCs and/or non-genetic covariates (must be included in *data* or *phenocov* if applicable)	NULL
covout	option to output the covariate coefficients, standard errors and their *P*-values from multi-SNP region-level regression models (used to construct MLC, LC and Generalized Wald tests)	FALSE
pheno	name of the phenotype column in *phenocov* to consider for the analysis	Required
pheno_type	pheno_type="C" if phenotype is **c**ontinuous, or pheno_type="D" if phenotype is **d**ichotomous	Required
geno_type	Format of the genotypes: geno_type="D" if genotypes are in allele dosage format , or genotype format (geno_type="G")	Required

	– argument required for SKAT/SKATO tests	
***If data/info data frames are specified* (in this case, recodeVCF is not used)**		
data	dataframe that includes the genotypes (in columns) and individuals (in rows)	Required if *vcfname* is left empty
SNPinfo	dataframe that includes the information for each variant (in rows), must include the following columns: "chr", "pos", "variant", "ref", "alt", "multialSNP", "multialSNP.rec","freq.alt" "multialSNP": binary variable=0, if the SNP is bi-allelic (two alleles), or 1 if more than two alleles "multialSNP.rec": indicates if the multi-allelic SNP has been recoded b recodeVCF (such that baseline allele = major allele) "freq.alt": frequency of the alternate allele (minor allele) This input can be generated using the auxiliary function *recodeVCF*.	Required if *vcfname* is left empty
Used only by the auxiliary function recodeVCF ***if data and SNPinfo are not specified***		
vcfname	name of the VCF file (if required)	NULL
qcmachr2	threshold used to filter out SNPs with low Mach R2 imputation quality score	NULL
qcinput	dataframe that includes at least 2 columns: variant & info_score (for filtering based on imputation quality score)	NULL
info_score	threshold used to filter out SNPs with low Info imputation quality score – If this option is specified, "qcinput" must be specified, and "info_score" must be a column in "qcinput"	NULL
Options to include multiallelic SNPs		
multiallelic	If FALSE: extract & process only the biallelic SNPs, If TRUE: include multiallelic SNPs in addition to the biallelic SNPs.	FALSE
multial_nmaxalleles	For extraction of SNPs with less than multial_nmaxalleles alleles (e.g. 2 to extract bi-allelic SNPs (default), 3 to extract bi-allelic and tri-allelic SNPs, etc).	2
Options for pruning/filtering of the SNPs in each region		
mafcut	threshold to filter out the SNPs with MAF ≤ *mafcut*	0.05
LDpruning	option to prune out the SNPs within each region based on the absolute value of the region-level genotypes correlation matrix; recommended to reduce the multi-collinearity issues in the multi-SNP region-level regression models (see **section 2.1.2**).	TRUE
rcut	threshold to prune out the correlated SNPs (works only if LDpruning=TRUE)	0.99
Other options		
singleSNPall	Produces an additional output including the single-SNP results (and LD-bin level information) for all the SNPs before LD pruning/alias identification. – can increase computational time.	FALSE

firthreg	If firthreg ="TRUE", use a Jeffreys-prior penalized-likelihood regression implemented in the R package **brlmgm2**(Kosmidis *et al.*, 2020; Kosmidis and Firth, 2021) instead of the default maximum-likelihood logistic regression. This option is recommended for unbalanced case-control data / small minor allele count.	FALSE
parallel	By default, process & run region-level analysis sequentially; if parallel ="TRUE" proceed in parallel.	FALSE
regionlist	list of the region names from REGIONinfo input to be analyzed for the analysis of a subset of regions.	NULL
alltests	By default, output region-level tests: Wald, MLCB, PC80, SKAT, SKATO, LCB, GATES, SimpleM; If alltests="TRUE" reports additional region-level tests: MLCZ, LCZ	FALSE
edgecut	parameter for clustering of the SNPs in LD bins (based on SNP correlation) using the CLQ algorithm applied before MLC test.	0.5
tol	tolerance parameter to deal with convergence issues in regression models.	1e-16
MLCheatmap	For each region, produces four heatmap plots of the region-level SNP correlation matrix, with SNPs ordered by positions & by LD bin, before and after LD pruning (if applicable) as illustrated in **Fig S2.** We recommend using this option for investigation of a subset of regions of interest specified with the option regionlist.	FALSE
SKAT_kernel	type of kernel used for SKAT and SKATO tests. There are 6 types of pre-specified kernels: "linear", "linear.weighted", "IBS", "IBS.weighted", "quadratic" and "2wayIX". See "kernel" argument in SKAT function in for details https://cran.r-project.org/web/packages/SKAT/SKAT.pdf.	linear.kernel
SKAT_weights	a numeric vector of weights for the weighted kernels. When it is NULL, the beta weight with the "SKAT_weights_beta" parameter is used.	NULL
SKAT_weights_beta	a numeric vector of parameters for the beta weights for the weighted kernels. If you want to use your own weights, please use the "weights" parameter. It will be ignored if "weights" parameter is not null.	c(1,25)

Table 3. Region-level output

Column name	Description
chr	chromosome # (as provided in *REGIONinfo* input*)*
region	region # or name (as provided in *REGIONinfo* input*)*
start.bp	start region position in bp (as provided in *REGIONinfo* input)
end.bp	end region position in bp (as provided in *REGIONinfo* input)
NSNPs	NSNPs in region (before pruning on LD/perfect linear dependency)
NSNPs.kept	NSNPs analyzed in region (after pruning on LD/perfect linear dependency)
max.VIF	Maximum VIF value among the NSNPs.kept
Wald	Generalized Wald statistic
Wald.df	degree of freedom of the Generalized Wald statistic (= NSNPs.kept)
MLCB	MLCB test statistic
MLCB.df	degree of freedom of the MLCB test (= # of LD bins in the region)
MLCB.p	*P*-value for the MLCB test
LCB	LCB test statistic
LCB.df	degree of freedom of the LCB test (=1 for all the regions)
LCB.p	*P*-value for the LCB test
PC80	PC80 test statistic
PC80.df	degree of freedom of the PC80 test
PC80.p	*P*-value for the PC80 test
SKAT	SKAT statistic
SKAT.pDavies	*P*-values from SKAT computed using Davies's method
SKAT.pLiu	*P*-values from SKAT computed using Liu's method
SKATO.p	SKATO test statistic
GATES.df	degree of freedom for the GATES test
GATES.p	*P*-value for the GATES test
SimpleM.p	*P*-value for the simpleM test
single_Wald.p	Minimum *P*-value from single-SNP analysis among the NSNPs.kept
If alltests=TRUE	
MLCZ	MLC statistic based on Z-scores from multi-SNP region-level regression model (rather than on SNP effects as used for MLCB)
MLCZ.p	*P*-value for MLCZ test statistic from multi-SNP region-level regression model (rather than on SNP effects as used for MLCB)
LCZ	LC statistic based on Z-scores (rather than SNP effects) from multi-SNP region-level regression model (rather than on SNP effects as used for MLCB)
LCZ.p	*P*-value for LCZ test statistic

Table 4. Bin-level output (specific to the MLC test)

Column name	Description
chr	chromosome # (as provided in input *REGIONinfo)*
region	region # or name (as provided in input *REGIONinfo)*
start.bp	start region position in bp (as provided in input *REGIONinfo)*
end.bp	end region position in bp (as provided in input *REGIONinfo)*
bin	LDbin # assigned within each region (bin #1 corresponds to the bin with largest bin.size)
bin.size	# of SNPs in LD bin
bin.size.keptSNPs	# of SNPs in LD bin (after pruning on MAF & LD (if applicable), and complete linear dependency)
deltabinB	Bin-level test statistic
deltabinB.p	*P*-value for 1 df Wald test of deltabinB
If alltests=TRUE	
deltabinZ	Same as deltabinB but using Z statistics instead of Betas
deltabinZ.p	Same as deltabinB but using Z statistics instead of Betas

Table 5. Variant-level output

Column name	Description
chr	chromosome # (as provided in *REGIONinfo* input)
region	region # or name (as provided in *REGIONinfo* input)
start.bp	start region position in bp (as provided in *REGIONinfo* input)
end.bp	end region position in bp (as provided in *REGIONinfo* input)
variant	variant name (as provided in *SNPinfo* input)
pos	SNP position (as provided in *SNPinfo* input)
multiallelicSNP	indicator variable to flag multiallelic SNPs (as provided in *SNPinfo* input)
ref	Reference allele (as specified in *SNPinfo* input)
alt	Alternate allele (as specified in *SNPinfo* input)
maf	Minor allele frequency
LDbin	LDbin # assigned within each region (numbered by decreasing # of SNPs)
LDbin.size	# of SNPs analyzed in each LDbin (kept after pruning on MAF & LD)
MLC.flip	Flag the SNPs recoded for MLC/LCbin tests
LC.flip	Flag the SNPs recoded for the LCbin tests
sg.beta	SNP effect estimate from single-SNP regression models
sg.pval	*P*-value for 1 *df* Wald test of SNP effect from single-SNP regression models
VIF	Variance inflation factor (VIF) values based on *all SNPs* analyzed in each region
glm.beta	SNP effect estimate from the region-level multi-SNP regression model
glm.pval	*P*-value for 1 *df* Wald test of SNP effect the region-level multi-SNP regression model

Table 6. List of SNPs excluded from the region-level tests and reasons for exclusion

Column name	Description
chr	Chromosome # (as provided in input *REGIONinfo)*
region	region # (as provided in input *REGIONinfo)*
start.bp	end position in bp (as provided in input *REGIONinfo)*
end.bp	start position in bp (as provided in input *REGIONinfo)*
variant	variant name (as provided in input *REGIONinfo)*
multiallelicSNP	indicator variable to flag multiallelic SNPs (as provided in input *REGIONinfo)*
pos	SNP position (as provided in input *SNPInfo)*
MAF	Minor allele frequency
reason	Reason of exclusion: "mafcut" – MAF < *mafcut* (if applicable) "rcut" – high correlation (if applicable) "alias" – complete linear dependency "multial" – multiallelic SNP (if applicable)

Table 7. Optional output produced if singleSNPall=TRUE. This output includes single-SNP results and bin-level information for all SNPs in regions before LD pruning

Column name	Description
chr	Chromosome # (as provided in input *REGIONinfo)*
region	region # (as provided in input *REGIONinfo)*
start.bp	end position in bp (as provided in input *REGIONinfo)*
end.bp	start position in bp (as provided in input *REGIONinfo)*
variant	variant name (as provided in input *SNPinfo*)
pos	SNP position (as provided in input *SNPinfo*)
multiallelicSNP	indicator variable to flag multiallelic SNPs (as provided in input *SNPinfo*)
major.allele	Major allele (as provided in input *SNPinfo*)
minor.allele	Minor allele (as provided in input *SNPinfo*)
maf	Minor allele frequency
LDbin	Bin # within each region (numbered by decreasing # of SNPs)
sg.beta	SNP effect estimate from single-SNP regression models
sg.pval	*P*-value for 1 *df* Wald test of SNP effect from single-SNP regression models
rmcorr	Indicator variable to flag SNPs excluded because of LD pruning