Bioconductor's SNPath package

TBD
Fred Hutchinson Cancer Research Center
email: TBD

October 13, 2010

Contents

1 Overview

Genome-wide association studies (GWAS) assess the association between individual SNPs and disease risk, and have successfully identified susceptibility loci for various complex diseases. In addition, many methods have been proposed to evaluate the association between disease risk and a set of SNPs that belongs to functional gene sets or pathways. The `SNPath` package contains four different algorithms in the literature: `grass` [1], `gseaSnp` [4], `plinkSet` [3] and `aligator` [2]. Users can use any one of them to identify pathways that are associated with disease risk; meanwhile, this package provides a nice and convenient platform for comparison of different algorithms as well.

This document provides a tutorial for using the `SNPath` package, as well as detailed description of each algorithm in this package. Specifically, the function `grass` calculate the pvalues for *a prior* defined pathways (sets of genes), based on the *Gene set Ridge regression in ASsociation Studies* (GRASS) algorithm [1]. The algorithm summarizes the genetic structure using singular value decomposition for each gene as eigenSNPs and uses a novel form of regularized regression technique, termed group ridge regression, to select representative eigenSNPs for each gene and assess their joint association with disease risk. The function `gseaSnp` implements the phenotype-permution algorithm proposed in Wang *et al.* (2007) [4]. This algorithm modifies the gene-set enrichment analysis approach for expression studies and is considered the first approach for gene-set enrichment analysis in association studies. The function `plinkSet` implements the set-based tests in the popular whole genome association analysis toolset, PLINK software. The function `aligator` performs a simple and fast ALIGATOR algorithm [2], which is a method for testing

overrepresentation of pathways, in lists of significant SNPs from GWAS. With a simulated example, we will show how to use each function with various options.

2 Simulated Data

We simulated a simple data set to demonstrate the use of functions with various options in the `SNPath` package. The simulated data is also included in the package as the dataset `simDat`. It consists of 5 elements. A SNP data `snpDat` consists of the genotype data for 300 SNPs and 1000 samples (500 cases and 500 controls). Each row represents 1 SNP and is coded in the trinary fashion (0,1,2 corresponding to homozygotes for the major allele, heterozygotes, and homozygotes for the minor allele, respectively). `snp.info` is a 300 by 3 matrix, with each row listing the name, chromosome and genome location (base pair) of a SNP. `gene.info` is a 20 by 4 matrix showing the genome information for 20 genes. The four columns are gene name, chromosome number, start and end coordinate of the gene (start<end). `y` is the case control status for the 1000 samples, with cases coded as 1 and controls coded as 0. `sim.pathway` is a list of gene names in two simulated pathways with 5 and 10 genes, respectively.
One can load the data set by `data(simDat)`.

```
> library(SNPath)
> data(simDat)
> ls()

[1] "snpDat" "snp.info" "gene.info" "sim.pathway" "y"

> dim(snpDat)

[1] 300  1000

> dim(snp.info)

[1] 300  3
```

3 The grass function

The `grass` function calculates the p-values of disease-association for pathways by the algorithm GRASS [1]. GRASS summarizes the genetic structure by SVD for each gene as eigenSNPs and uses a novel form of regularized regression technique, termed group ridge regression, to select representative eigenSNPs for each gene and assess their joint association with disease risk.

In the following example, `grass` function estimates p-values for two simulated pathways. Because most of the pathway analysis algorithms are nonparametric and requires a large number of permutations (or resamplings). It is highly recommended that one uses parallel computing to make best use of all available resources. Please see the R library `snow` and `Rmpi` for more information on parallel computing. If `cl=NULL`, clusters are not provided, all computation will be done on the local computer. If `cl` is generated, please also load the `SNPath` library on each cluster.

The following code loaded the simulated data, generated two clusters on the local computer, and loaded the `SNPath` library on each clusters. The example implemented the unweighted analysis of GRASS and obtained the pvalues for the two simulated pathways. In this example,

`gene.def="ref"`, i.e., relative gene definition is used, with default distance `k=1`. That is, a SNP is assigned to the gene it physically located in, if it does, and one nearest gene from either end of gene (or SNP).

```
> library(SNPath)
> data(simDat)
> library(snow)
> cl <- makeCluster(c("localhost","localhost"), type = "SOCK")
> clusterEvalQ(cl, {
>    library(SNPath)
> })
> path.pval <- grass(cl=cl, snp.dat=snpDat, snp.info=snp.info, gene.info=gene.info,
                     gene.set=sim.pathway, y=y, gene.def="rel", B=1000,
                     nominal.p=FALSE)
> path.pval

path1 path2
0.622 0.356
```

The option `nominal.p` indicates if p-values should be calculated based on normal approximation of null statistics. By default, it is set to be `FALSE`, and p-values are calculated non-parametrically by counting the frequency of null pathway statistics exceeding the observed one. When computation load is heavy, one can use a small number of permutations, `B`, and set `nominal.p = TRUE`.

```
> path.pval <- grass(cl=cl, snp.dat=snpDat, snp.info=snp.info, gene.info=gene.info,
                     gene.set=sim.pathway, y=y, gene.def="rel", nominal.p=TRUE)
> path.pval

    path1     path2
0.7612674 0.4150053
```

4 The gseaSnp function

The function `gseaSnp` calculates the p-value of disease-association for pathways by the algorithm proposed in Wang *et al.* (2007) [4]. Briefly, the idea behind the algorithm is: first assign SNPs to genes based on absolute or relative genome location, and use the top individual SNP association statistic within the gene as the statistic of the gene and rank all the genes by significance. The algorithm then compares the distribution of the ranks of genes from a given pathway to that of the remaining genes via a weighted Kolmogorov-Smirnov test, with greater weight given to genes with more extreme statistic values. To compute the p-values for pathways of interest, one can permute the case/control status and repeat the above procedure to obtain null pathway statistics, and count the frequency of null statistics exceeding the observed ones.

There are many ways to calculate the individual SNP association statistics. Here we provide two options: the default option is to use logistic regression by setting `snp.method = "logiReg"`, and if `weights` are provided, weighted logistic regression will be performed (requires loading the R library `Zelig` and `survey`). Another option is to use chi-square test by setting `snp.method = "chiSq"`.

Here we show an example of using logistic regression without weight. In this example, SNPs within 5 base pair of either end of a gene are assigned to the gene (`gene.def="abs"` and `dist=5`).

```
> path.pval <- gseaSnp(cl=cl, snp.dat=snpDat, snp.info=snp.info,
                       gene.info=gene.info, gene.set=sim.pathway, y=y,
                       snp.method="logiReg", gene.def="abs",dist=5)
> path.pval

path1 path2
0.125 0.865
```

If sampling weights are available, one can also specify the `weights` option to perform a weighted analysis. In the following example, we use a equal weight on each sample, and should ideally get the same results as above. The small difference between the two p-values for `path1` is due to rounding error.

```
> path.pval <- gseaSnp(cl=cl, snp.dat=snpDat, snp.info=snp.info,
                    gene.info=gene.info,gene.set=sim.pathway, y=y,
                    weights=rep(1,length(y)),
                    snp.method="logiReg", gene.def="abs",dist=5)
> path.pval

path1 path2
0.120 0.865
```

Also one can use chi-square test to assess individual SNP associations, by setting `snp.method="chiSq"`.

```
> path.pval <- gseaSnp(cl=cl, snp.dat=snpDat, snp.info=snp.info,
                    gene.info=gene.info, gene.set=sim.pathway, y=y,
                    snp.method="chiSq", gene.def="abs",dist=5)
> path.pval

path1 path2
0.345 0.725
```

5 The `plinkSet` function

`PLINK` is a popular software for analyzing whole genome association studies [3]. It provides an option to perform SNP set-based tests. We implement the idea in `R` to facilitate users who want to compare different pathway analysis methods.

This algorithm works as follows:

1. Assign SNPs to each pathway of interest.
2. Perform standard single SNP analysis (logistic regression or chi-square test).
3. For each set, select the top "independent" SNPs with p-values below `snp.pcut`. The best SNP is selected first; subsequent SNPs are selected in order of descreasing statistical significance, after removing SNPs in linkage disequilibrium (r^2 above `snp.r2cut`) with previously selected SNPs.

4. The statistic for each pathway is calculated as the mean of these single SNP statistics from the top "independent" SNPs within the pathway.

5. Permute the case/control status `y` B times, keeping SNP dataset unchanged.

6. For each permuted dataset, repeat steps 2 to 4 above.

7. The p-value for a pathway is the number of times the permuted pathway statistic exceeds the observed one.

In the following example, we use the set-based tests in PLINK to calculate the p-values for the two simulated pathway. Again we use logistic regression to perform standard single SNP analysis. SNPs with p-values below 0.05 are retained, and their pairwise r^2s are less than 0.5.

```
> path.pval <- plinkSet(cl=cl, snp.dat=snpDat, snp.info=snp.info,
                  gene.info=gene.info, gene.set=sim.pathway, y=y,
                  snp.method="logiReg", gene.def="abs", dist=10,
                  snp.pcut=0.05, snp.r2cut=0.5)
> path.pval

path1 path2
 0.56  0.44
```

6 The `aligator` function

The `aligator` function implements the ALIGATOR algorithm [2]. The algorithm takes the individual SNP association p-values as input and use a preselected p-value threshold `snp.pcut` to define a set of significantly associated SNPs. It then counts the number of genes in a pathway that contains these SNPs, with each gene counted only once, regardless of the number of significant SNPs in the gene. Instead of permuting phenotypes to establish the null distribution as in PLINK, ALIGATOR uses resampling of SNPs. Thus it only requires a p-value or summary statistic from each SNP as input, and can be used when individual level SNP data are not available.

In the following example, we calculated the SNP association p-values by logistic regression, using the function `calc.fun`. We defined SNPs with p-values less than 0.05 as significant SNPs (`snp.pcut=0.05`) and performed ALIGATOR on the data.

```
> pval <- calc.fun(cl=cl, snp.dat=snpDat, y=y, snp.method="logiReg")$pval

> path.pval <- aligator(cl=cl, snp.info=snp.info, gene.info=gene.info,
                    gene.set=sim.pathway, snp.pval=pval, gene.def="rel", snp.pcut=0.05)
> path.pval

path1 path2
0.5686 0.9008
```

References

[1] Lin S. Chen, Carolyn M. Hutter, John D. Potter, Yan Liu, Ross L. Prentice, Ulrike Peters, and Li Hsu. Insights into colon cancer etiology via a regularized approach to gene set analysis of gwas data. *The American Journal of Human Genetics*, 86(6):860–871, 2010.

[2] P. Holmans, E. K. Green, J. S. Pahwa, M. A. Ferreira, S. M. Purcell, P. Sklar, the Wellcome Trust Case-Control Consortium, M. J. Owen, M. C. O'Donovan, and N. Craddock. Gene ontology analysis of gwa study data sets provides insights into the biology of bipolar disorder. *The American Journal of Human Genetics*, 85(1):13–24, 2009.

[3] S. M. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. de Bakker, and M. J. Daly *et al.* Plink: a toolset for whole-genome association and population-based linkage analysis. *The American Journal of Human Genetics*, 81:559–575, 2007.

[4] K. Wang, M. Li, and M. Bucan. Pathway-based approaches for analysis of genomewide association studies. *The American Journal of Human Genetics*, 81(6):1278–1283, 2007.