Package 'jacpop'

October 13, 2022

Type Package

Title Jaccard Index for Population Structure Identification

Description Uses the Jaccard similarity index to account for population structure in sequencing studies. This method was specifically designed to detect population stratification based on rare variants, hence it will be especially useful in rare variant analysis.

Version 0.6

Author Dmitry Prokopenko

Maintainer Dmitry Prokopenko <dmitry.prokopenko@uni-bonn.de>

License GPL-3

LazyData TRUE

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-06-03 14:40:10 UTC

R topics documented:

`generate_pw_jaccard` *Calculate pairwise Jaccard similarity matrix*

Description

Computes pairwise Jaccard similarity matrix from sequencing data and performs PCA on it. The function is specifically useful to detect population stratification in rare variant sequencing data.

Usage

```
generate_pw_jaccard(geno, pop.label = NULL, n.pcs = 10,
  plot_it = TRUE)
```

Arguments

geno	A numeric nxm matrix of genotypes. Rows are individuals and columns are variants. The genotypes should be coded 0, 1 and 2. Missing entries are coded as NA. The natural input would be a matrix produced by PLINK using the option –recodeA and removing the first row and the first 6 columns.
pop.label	(Optional) A numeric or character vector of n population labels(if known). It is used for plotting purposes.
n.pcs	A numeric scalar. Number of principal components to extract from the Jaccard similarity matrix. Set to NULL, if you want just the similarity matrix.
plot_it	A logical scalar. Should the first 2 principal components be plotted?

Details

In order to account for population structure in sequencing data we propose to calculate the Jaccard similarity instead of the genetic covariance between individuals.

$$Jac(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

This similarity index is most suitable for sparse data, which is the case, when we restrict our analysis to variants with low minor allele frequencies. The pairwise Jaccard similarity matrix can be further used in Principal Component Analysis.

Although the function does basic filtering (singletons, SNPs with missing entries), we recommend to extract a subset of possibly independent SNPs (500k - 1M should be enough) from your initial dataset for population structure identification. You could either extract a random subset of variants or prune your dataset.

Value

A list of 2 elements:

- A nxn numeric matrix, where the entries are Jaccard similarity indicies between a pair of individuals. The order of individuals corresponds to the order in the input genotype matrix.
- A data.frame of principal components, which can be further used in an association analysis. The order of individuals corresponds to the order in the input genotype matrix.

References

Prokopenko, D., Hecker, J., Silverman, E., Pagano, M., Noethen, M. M., Dina, C., Lange, C., Fier, H. L. (2015). Utilizing the Jaccard index to reveal population stratification in sequencing data: A simulation study and an application to the 1000 Genomes Project. *Bioinformatics*, 32, 1366-1372.

Examples

```
#####1)Toy example
#Simulate genotypes in 2 populations
nsnps=10000
fst=0.01
nind=20
maffilter=0.05
p<-runif(nsnps,0,maffilter)
freq1<-sapply(1:length(p),function(x) rbeta(1,p[x]*(1-fst)/fst,(1-p[x])*(1-fst)/fst))
freq2<-sapply(1:length(p),function(x) rbeta(1,p[x]*(1-fst)/fst,(1-p[x])*(1-fst)/fst))

pop1<-sapply(1:nsnps, function(x) sample(c(0,1,2),nind,replace=TRUE,
 prob=c(((1-freq1[x])^2),(2*freq1[x]*(1-freq1[x])),(freq1[x]^2))))
pop2<-sapply(1:nsnps, function(x) sample(c(0,1,2),nind,replace=TRUE,
 prob=c(((1-freq2[x])^2),(2*freq2[x]*(1-freq2[x])),(freq2[x]^2))))
all<-as.matrix(rbind(pop1,pop2))

#Generate the Jaccard similarity index and plot the first 2 principal components
res<-generate_pw_jaccard(geno=all,pop.label=c(rep(1,nind),rep(2,nind)))
## Not run:
#####2)PLINK files
#If you are working with plink files after filtering the dataset consider
#to create a genotype count file by using the option --recodeA.
#After that remove the first row and the first 6 columns. Now you can
# read it in in the following way:
geno<-matrix(scan('sample.geno'),nrow=nind,byrow=T)
# nind is the number of individuals(rows)

## End(Not run)
```

Index