invClust: inference of polymorphic inversions from multivariate analysis on SNP data

Alejandro Cáceres, Juan R. González
Center for Research in Environmental Epidemiology (CREAL), Barcelona, Spain
Bioinformatics Research Group in Epidemiolgy (http://www.creal.cat/brge.htm)

May 18, 2020

1 Introduction

invClust is a method to detect inversion-related haplotypes in SNP data [1]. Under the presence of an inversion polymorphism in the population sample, one can expect to find three clearly differentiated haplotype-genotypes supported by the suppression of recombination associated with the inversion [2]. The haplotype-genotypes are the combination of two haplotypes each of which tag an inversion state (standard: NI or inverted: I).

The algorithm first performs a multidimensional scaling (MS) of SNP genotypes within the inverted segment [3]. Using the first two components of the MS analysis, the algorithm then decides whether there is a three-cluster pattern in the data, each corresponding to a haplotype-genotype. If this is the case, then inversion status is given for each individual, assuming that the discovered haplotypes tag the inversion. The result is a list with the inferred inversion genotypes per individual (NI/NI: non-inverted homozygous, NI/I: inverted heterozygous, I/I: inverted homozygous).

More technically, the clustering of the three genotypes is based on a mixture modelling which assumes that the homozygous groups are equidistant from the heterozygous group, and that the frequency of the inversion alleles are in Hardy-Weinberg Equilibrium.

The inference is performed on a region of interest (ROI), internal to inversion being interrogated. Optimally, the ROI should extend to the breakpoints of the inversion to include all the SNPs within the inverted segment. However, since not all SNPs are informative, the analyses are robust under some variation in the brake-points

2 Preparing data

Load the package

```
> library(invClust)
```

We have developed the method to read genotypes in the SnpMatrix format from the snpStats package from bioconductor which can be installed as

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("snpStats")
```

The main advantage of this format is that binary PLINK files (http://pngu.mgh.harvard.edu/ purcell/plink/), supported by most dbGap (http://www.ncbi.nlm.nih.gov/gap) studies, can be readily loaded. Having the three PLINK files, like genosPLINK.bed, genosPLINK.bim, genosPLINK.fam, the genotype are loaded as

```
> library(snpStats)
> path<-system.file("extdata", package = "invClust")
> genofile<-file.path(path,"genosPLINK")
> #genofile is the path where the genoPLINK demo files are stored
> #use your own path for your own data
> geno.data<-read.plink(genofile)
```

The SNP genotypes and annotation to be used by `invClust` are simply coded in two variables "geno" (as a SnpMatrix object)

```
> geno<-geno.data$genotypes
> geno

A SnpMatrix with  165 rows and  2862 columns
Row names:  NA06989 ... NA12865
Col names:  rs13266763 ... rs12719915
```

and "annot"; a variable with three columns (chromosome, snp.name and position)

```
> annot.read<-geno.data$map
> annot<-annot.read[,c(1,2,4)]
> head(annot)

           chromosome   snp.name position
rs13266763          8 rs13266763  8006806
rs2980436           8  rs2980436  8129435
rs2945254           8  rs2945254  8130650
rs2980437           8  rs2980437  8132173
rs10092295          8 rs10092295  8139051
rs712253            8   rs712253  8141192
```

please keep the strict order of the columns and the increasing order on the SNP positions. Genotype SNP names must match the ordering of the SNP names in the annotation.

```
> identical(annot[,2],colnames(geno))

[1] TRUE
```

The genotypes in the current example correspond to the SNPs within the inv-8p23 for the CEU subjects of the HapMap that we have sub-selected for illustration. The complete data set can be found in (ftp://ftp.ncbi.nlm.nih.gov/hapmap/genotypes/2009-01_phaseIII/plink_format/) and conversion from .ped .map to .bin, .fam and .map formats can be done with PLINK.

Note that if you do not have your genotypes in PLINK format you can still create a SnpMatrix with the function `new` using your genotypes coded as 0,1 and 2

```
> geno.mat<-matrix(c(0,0,1,2,0,0),ncol=2)
> rownames(geno.mat)<-c("sub1","sub2","sub3")
> colnames(geno.mat)<-c("rs1","rs2")
> geno.raw<-matrix(as.raw(geno.mat+1),ncol=ncol(geno.mat))
> geno.new<-new("SnpMatrix", geno.mat)

coercing object of mode  numeric  to SnpMatrix

> geno.new

A SnpMatrix with  3 rows and  2 columns
Row names:  sub1 ... sub3
Col names:  rs1 ... rs2
```

3 Running `invClust`

Now an ROI data.frame should be provided with the inversion brake-points where we would like to run the analysis. The ROI can be passed as a `data.frame` variable or as a file with the list of regions to be tested. Here we illustrate the ROI defined by inv-8p23

```
> roi<-data.frame(chr=8,LBP=7934925, RBP=11824441, reg= "inv1")
```

the algorithm is called with the function `invClust`

```
> invcall<-invClust(roi=roi, wh = 1, geno=geno, annot=annot, dim=2)
```

where the parameters `wh` refer to the ROI we want to analyze, in case that the roi `data.frame` has more than one ROI, and dim is the number of dimensions kept in the MDS analysis. If you have more than one ROI to test, the roi argument can also be the file name (e.g "ROI.txt") where the ROIs are stored under the format

```
  chr   LBP       RBP       reg
1 8     7934925   11824441  inv1
2 17    41026708  41685507  inv2
..
```

in this case `invClust` can be called like

```
> invcall<-invClust(roi="ROI.txt", wh = 1, geno=geno, annot=annot, dim=2)
```

The algorithm computes the mixture model using an expectation maximization routine with two main initial conditions; one close to inversion frequency near zero and other around 50%. From these two models and one with no inversion signal (a Gaussian model with no clusters), it selects the model with highest Bayes Information Criterion (BIC).

The result is a object of class `invClust`

```
> invcall
```

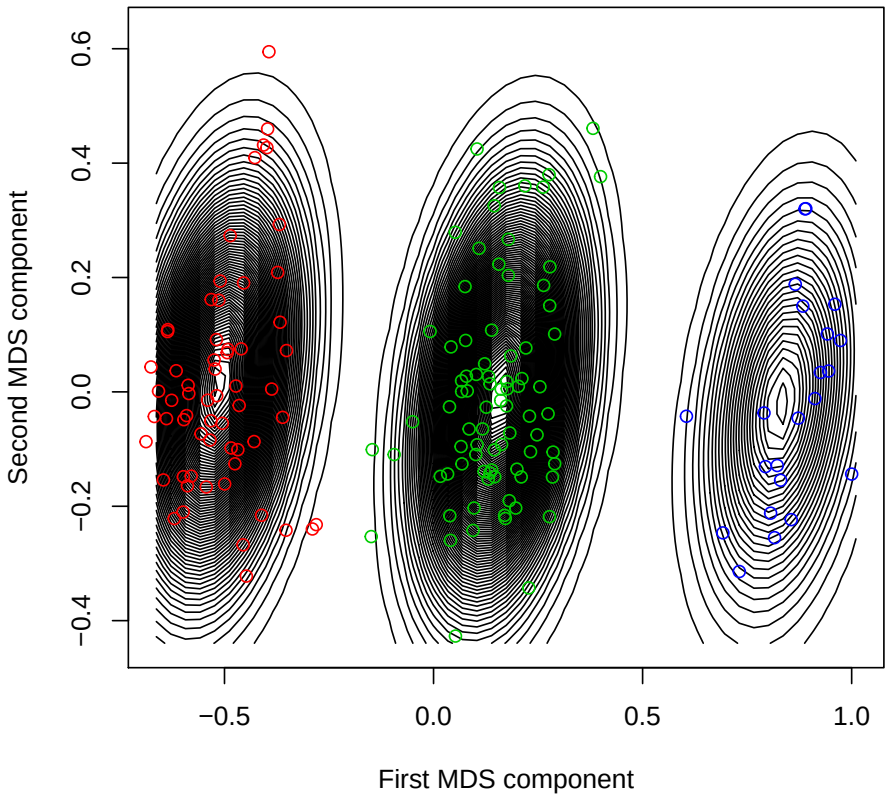
```
Inversion genotype clustering
-object of class invClust-
  fields: $EMestimate: mixture model parameters
          $datin: fitted data
  subjects: 165
  groups fitted: 1
  overall inversion allele frequency: 0.3817809
  variance explained by  5  MDS componet(s): 0.8642412
```

the print of the object gives you a brief description on the main parameters in the computation. The MDS analysis with the mixture model can be plotted

```
> invcall
```

```
Inversion genotype clustering
-object of class invClust-
  fields: $EMestimate: mixture model parameters
          $datin: fitted data
  subjects: 165
  groups fitted: 1
  overall inversion allele frequency: 0.3817809
  variance explained by  5  MDS componet(s): 0.8642412
```

```
> plot(invcall)
```



The inversion-genotypes are easily extracted with

```
> inv<-invGenotypes(invcall)
> head(inv)
```

```
NA06989 NA11891 NA11843 NA12341 NA12739 NA10850
    I/I    NI/I    NI/I   NI/NI   NI/NI   NI/NI
Levels: NI/NI NI/I I/I
```

Note that the inversion genotype NI/NI is by default the most frequent allele in the population. To compute the reference allele as the ancestral allele we recommend to 1) extract the predicted ancestral alleles for each SNP in the inversion (1000 genomes) 2) add a homozygous subject for all the ancestral SNP alleles in your genotypes and 3) identify the homozygous group in which this subject falls as the ancestral configuration.

Finally the inversion genotypes can also be computed from their probabilities of belonging to each genotype cluster

```
> invUnc<-invcall["genotypes"]
> head(invUnc)
```

```
        NI/NI NI/I I/I
NA06989     0    0   1
NA11891     0    1   0
NA11843     0    1   0
NA12341     1    0   0
NA12739     1    0   0
NA10850     1    0   0
```

4 Association Tests

As an example we show how to perform association tests of these genotypes using the package SNPassoc that can be accessed in r-cran

```
> install.packages("SNPassoc")
```

```
> library(SNPassoc)
```

We have simulated a normalized BMI that can be found in the file BMI (separated by TAB).

```
> path<-system.file("extdata", package = "invClust")
> phenofile<-file.path(path,"BMI.txt")
> BMI<-read.delim(phenofile,as.is=TRUE)
> head(BMI)
```

```
       ID        BMI
1 NA06989 -0.6221268
2 NA11891  0.5964190
3 NA11843 -1.2691413
4 NA12341 -0.9126277
5 NA12739  0.3573106
6 NA10850  1.7389315
```

We merge the genotype and phenotype data checking the correct order for the subjects' IDs.

```
> identical(BMI$ID, names(inv))

[1] TRUE

> data<-cbind(BMI,inv)
```

Before association tests SNPassoc requires setting up the genotype columns as class snp

```
> data.end<-setupSNP(data,colSNPs=3)
> head(data.end)

       ID        BMI   inv
1 NA06989 -0.6221268   I/I
2 NA11891  0.5964190  NI/I
3 NA11843 -1.2691413  NI/I
4 NA12341 -0.9126277 NI/NI
5 NA12739  0.3573106 NI/NI
6 NA10850  1.7389315 NI/NI

> class(data.end$inv)

[1] "snp"    "factor"
```

We can then test association for all the genetic models of the polymorphic inversion

```
> association(BMI~inv,data.end)

SNP: inv  adjusted by:
               n       me      se     dif    lower   upper p-value   AIC
Codominant
NI/NI         61  0.03966 0.12219  0.0000                   0.02914 447.3
NI/I          82 -0.10198 0.09982 -0.1416 -0.44802 0.16473
I/I           22  0.49435 0.19539  0.4547  0.00406 0.90532
Dominant
NI/NI         61  0.03966 0.12219  0.0000                   0.91888 452.5
NI/I-I/I     104  0.02417 0.09165 -0.0155 -0.31324 0.28225
Recessive
NI/NI-NI/I   143 -0.04156 0.07736  0.0000                   0.01227 446.2
I/I           22  0.49435 0.19539  0.5359  0.12117 0.95066
Overdominant
NI/NI-I/I     83  0.16018 0.10541  0.0000                   0.07289 449.3
NI/I          82 -0.10198 0.09982 -0.2622 -0.54680 0.02248
log-Additive
0,1,2                             0.1304 -0.08359 0.34443 0.23406 451.1
```

References

[1] Cáceres and González, J. R. (2015) Following the footprints of polymorphic inversions on SNP data: from detection to association tests. *NAR*, **doi**:10.1093/nar/gkv073 .

[2] Ma, J. and Amos, C. I. (2012) Investigation of Inversion Polymorphisms in the Human Genome Using Principal Components Analysis. *PLOS ONE*, **7**, e40224.

[3] Salm, M. P., Horswell, S. D., Hutchison, C. E., Speedy, H. E., Yang, X., Liang, L., Schadt, E. E., Cookson, W. O., Wierzbicki, A. S., Naoumova, R. P., et al. (2012) The origin, global distribution, and functional impact of the human 8p23 inversion polymorphism. *Genome Res.*, **22**, 1144.