

Introduction to recombClust

Carlos Ruiz, Alejandro Caceres, Juan R Gonzalez

2020-09-15

Contents

1	Overview.	2
2	Getting data	3
3	Main function	4

1 Overview

`recombClust` is a R package that classifies chromosomes in different subpopulations based on recombination patterns. `recombClust` works with chromosomes, so we need phased data. To start, we will load the chromosomes from a VCF file into a `SnpMatrix`. To get `recombClust` input, we need to run some preprocessing steps:

```
library(recombClust)

## Packages to load SNP data
library(VariantAnnotation)
library(GenomicRanges)

## Function to load a VCF file, convert it to a snpMatrix
## and removing SNPs with a low MAF
getVCFmatrixChr <- function(range = NULL, samples = NULL, snps.names = NULL,
                           minmaf = 0.1, Remove.Granges = NULL, ...){
  vcf <- loadVCFrange(range, samples, ...)

  snpsVCF <- genotypeToSnpMatrix(vcf)
  snpsVCF$genotypes <- snpsVCF$genotypes[, !snpsVCF$map$ignore]
  sums <- col.summary(snpsVCF$genotypes)
  snps <- colnames(snpsVCF$genotypes)[sums$MAF > minmaf]

  vcf <- vcf[snps, ]
  geno <- geno(vcf)$GT
  phase <- lapply(1:ncol(geno), function(x){
    chr1 <- as.numeric(sapply(geno[, x], substring, 1, 1))
    chr2 <- as.numeric(sapply(geno[, x], substring, 3, 3))
    matrix(c(chr1, chr2), nrow = 2, byrow = TRUE)
  })
  phase <- Reduce(function(...) rbind(...), phase)
  rownames(phase) <- paste(rep(colnames(geno), each = 2), 1:2, sep = "_")
  colnames(phase) <- rownames(geno)
  snpsVCF <- list(genotypes = new("SnpMatrix", 2*phase + 1),
                 map = data.frame(name = colnames(phase)))
  ## Conversion from VCF to SNP matrix produces some
  ## SNPs to be NA (multiallelic or bigger than 1)
  snpsVCF$map$position <- start(rowRanges(vcf))
  snpsVCF$map$chromosome <- as.character(seqnames(rowRanges(vcf)))
  snpsVCF$map$allele.2 <- unlist(snpsVCF$map$allele.2)
  rownames(snpsVCF$map) <- rownames(geno)
  snpsVCF
}

## Load VCF selecting samples
loadVCFrange <- function(range = NULL, samples = NULL, vcffile){
  vcfsamples <- samples(scanVcfHeader(vcffile))
  if (!is.null(samples)){
    samples <- vcfsamples[vcfsamples %in% samples]
```

Introduction to recombClust

```
} else{
  samples <- vcfsamples
}
if (!is.null(range)){
  param <- ScanVcfParam(samples = samples, which = range)
} else {
  param <- ScanVcfParam(samples = samples)
}
vcf <- readVcf(vcffile, "hg19", param)
vcf
}
```

2 Getting data

Once we have loaded these functions in our workspace, we are ready to load SNP data. In this example, we will use example data from scoreInvHap package:

```
## Get vcf path
vcf_file <- system.file("extdata", "example.vcf", package = "recombClust")

## Load vcf file
snps <- getVCFmatrixChr(vcf = vcf_file)
#> coercing object of mode numeric to SnpMatrix
```

`snps` is a list that has two elements: `genotypes` and `map`. `genotypes` is a `SnpMatrix` object that contains the SNP data. Chromosomes are in rows and SNPs in columns. Each individual has two chromosomes, so row names have the id number followed by `'_1'` or `'_2'`:

```
snps$genotypes
#> A SnpMatrix with 60 rows and 167 columns
#> Row names: HG00096_1 ... HG00128_2
#> Col names: rs113928679 ... rs77407299
```

You can have a look at the data by coercing it to a numeric matrix. Chromosomes having the reference allele have a 0, and those having the alternative allele a 2:

```
as(snps$genotypes, "numeric")[1:10, 1:5]
#>      rs113928679 rs73387199 rs75989725 rs7800308 rs7776878
#> HG00096_1      0          0          0          0          0
#> HG00096_2      0          0          0          0          0
#> HG00097_1      0          0          0          0          0
#> HG00097_2      0          0          0          0          0
#> HG00099_1      0          0          0          0          0
#> HG00099_2      0          0          0          0          0
#> HG00100_1      0          0          0          0          0
#> HG00100_2      2          2          2          2          2
#> HG00101_1      0          0          0          0          0
#> HG00101_2      0          0          0          0          0
```

`Map` contains the name, chromosome and position of the SNPs:

Introduction to recombClust

```
head(snps$map)
#>           name position chromosome
#> rs113928679 rs113928679 54301673      7
#> rs73387199  rs73387199 54301951      7
#> rs75989725  rs75989725 54302232      7
#> rs7800308   rs7800308 54302736      7
#> rs7776878   rs7776878 54302737      7
#> rs7796456   rs7796456 54302784      7
```

You should convert them to a GenomicRanges prior passing them to recombClust:

```
GRsnps <- makeGRangesFromDataFrame(snps$map, start.field = "position",
                                   end.field = "position")
```

```
GRsnps
#> GRanges object with 167 ranges and 0 metadata columns:
#>           seqnames      ranges strand
#>           <Rle> <IRanges> <Rle>
#> rs113928679      7 54301673      *
#> rs73387199      7 54301951      *
#> rs75989725      7 54302232      *
#> rs7800308       7 54302736      *
#> rs7776878       7 54302737      *
#> ...           ...           ...
#> rs111454158      7 54369666      *
#> rs10278526       7 54369815      *
#> rs62451163       7 54373509      *
#> rs62451164       7 54373654      *
#> rs77407299       7 54376118      *
#> -----
#> seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

3 Main function

Now, we are ready to run recombClust. runRecombClust applies the LDmixture model to each SNP-block pair, selects pairs having a recombination plot, computes the probabilities matrix and clusters the individuals. runRecombClust requires the matrix with SNP data and the annotation in a GRanges. Notice that the matrix is divided by 2 to have a matrix with only 0s and 1s (0: reference allele, 1: alternative allele). To speed up the example, we will include only the first 20 SNPs:

```
## Create matrix with snps
snpMat <- as(snps$genotypes, "numeric")/2 ## Divide by 2 to have 0
                                           ## as reference and 1 as alternative

snpMat[1:10, 1:5]
#>           rs113928679 rs73387199 rs75989725 rs7800308 rs7776878
#> HG00096_1           0           0           0           0           0
#> HG00096_2           0           0           0           0           0
#> HG00097_1           0           0           0           0           0
#> HG00097_2           0           0           0           0           0
```

Introduction to recombClust

```
#> HG00099_1      0      0      0      0      0
#> HG00099_2      0      0      0      0      0
#> HG00100_1      0      0      0      0      0
#> HG00100_2      1      1      1      1      1
#> HG00101_1      0      0      0      0      0
#> HG00101_2      0      0      0      0      0
```

```
## Pass snpMat and GRanges to runRecombClust
res <- runRecombClust(snpMat[, 1:20], annot = GRsnps[1:20])
```

runRecombClust might take a long time to finish. res is a list with two main elements: class (cluster classification of each chromosome) and pc (PCA of the probabilities matrix).

```
table(res$class)
#>
#>  1  2
#> 14 46
plot(res$pc$x, col = res$class, pch = 16)
```

