cgmisc: package tutorial

Marcin Kierczak, Jagoda Jablonska, Simon Forsberg, Matteo Bianchi, Katarina Tengvall, Mats Pettersson, Jennifer Meadows, Patric Jern, Orjan Carlborg, Kerstin Lindblad-Toh

2 Feb 2015

cgmisc is an R package for enhanced genome-wide association studies (GWAS) and visualisation. This document aims at guiding you through the installation process and to demonstrate package capabilities in a series of practical examples based on an example data included in the package.

Package installation

The **cgmisc** package can be installed in the same way as any other R package. One way is to issue the following command in R console:

```
install.packages("cgmisc")
```

Other possibilities include using graphical user interface (GUI) of, e.g. R console or RStudio.

After the package has been installed, to use the package, it is necessary to load it into environment:

```
library("cgmisc")
```

```
## Loading required package: GenABEL
## Loading required package: MASS
## Loading required package: GenABEL.data
##
##  Package cgmisc contains miscellaneous functions, useful for extending
## genome-wide association study (GWAS) analyses.
##
## Package Name: cgmisc
##  Version: 2.9.8
##  Date: 2015-01-28
##  Author: Marcin Kierczak <marcin.kierczak@imbim.uu.se>
##  License GPL (>=2.10)
##
##  Package contains various functions useful in computational
##     genetics, especially in genome-wide association studies.
```

Loading data

Whenever possible, the **cgmisc** package uses data structures used by the GenABEL (Aulchenko et al., 2007) package. In particular, the `gwaa.data-class` and the `gwaa.scan-class` structures are used. The package is shipped with an example dataset called **cgmisc_data** that contains genotyping data (Illumina, canFam2) for N=207 German shepherds originally collected for the project described in (Tengvall et al., 2012). The phenotypes, though, have been simulated in order to be able to illustrate various features of **cgmisc**. To load the example dataset, use the following command:

```
data("data")
```

Example analyses

In order to illustrate how to use particular functions, we will perform a very much simplified GWAS analysis. We begin by initial quality control where we prune the data with per marker of per individual call rates below 95%. Based on 2000 randomly selected markers, we remove one (with lower call rate) from each pair of too similar (more than 95% similarity) individuals. We also set very low (10^{-3}) threshold for pruning on minor allele frequency (in practise only the monomorphic markers will be removed) and turn off checks based on the departure form Hardy-Weinberg equilibrium (p.level=10e-18)

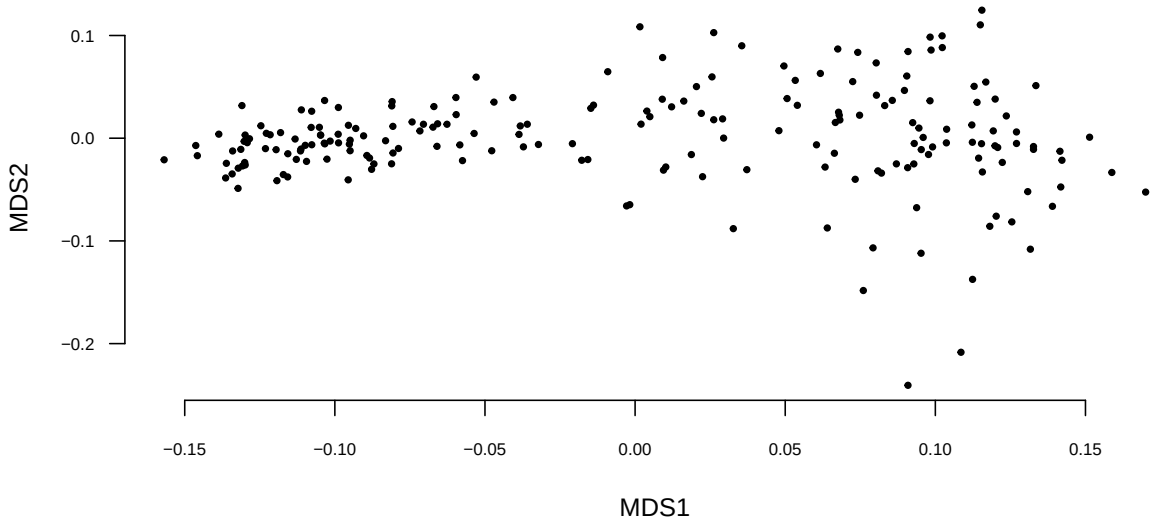
```
qc1 <- check.marker(data, callrate = .95, perid.call = .95, ibs.threshold = .95, ibs.mrk=2000, ibs.exclu
```

```
## Excluding people/markers with extremely low call rate...
## 174375 markers and 207 people in total
## 0 people excluded because of call rate < 0.1
## 1069 markers excluded because of call rate < 0.1
## Passed: 173306 markers and 207 people
##
## RUN 1
## 173306 markers and 207 people in total
## 42743 (24.66331%) markers excluded as having low (<0.1%) minor allele frequency
## 1468 (0.8470567%) markers excluded because of low (<95%) call rate
## 650 (0.3750591%) markers excluded because they are out of HWE (P <1e-17)
## 0 (0%) people excluded because of low (<95%) call rate
## Mean autosomal HET is 0.2658536 (s.e. 0.01917125)
## 0 people excluded because too high autosomal heterozygosity (FDR <1%)
## Mean IBS is 0.7753457 (s.e. 0.01530644), as based on 2000 autosomal markers
## 2 (0.9661836%) people excluded because of too high IBS (>=0.95)
## In total, 128942 (74.40135%) markers passed all criteria
## In total, 205 (99.03382%) people passed all criteria
##
## RUN 2
## 128942 markers and 205 people in total
## 0 (0%) markers excluded as having low (<0.1%) minor allele frequency
## 0 (0%) markers excluded because of low (<95%) call rate
## 0 (0%) markers excluded because they are out of HWE (P <1e-17)
## 0 (0%) people excluded because of low (<95%) call rate
## Mean autosomal HET is 0.2660211 (s.e. 0.01918304)
## 0 people excluded because too high autosomal heterozygosity (FDR <1%)
## Mean IBS is 0.7766199 (s.e. 0.01427284), as based on 2000 autosomal markers
## 0 (0%) people excluded because of too high IBS (>=0.95)
## In total, 128942 (100%) markers passed all criteria
## In total, 205 (100%) people passed all criteria
```

```
data.qc1 <- data[qc1$idok, qc1$snpok]
```

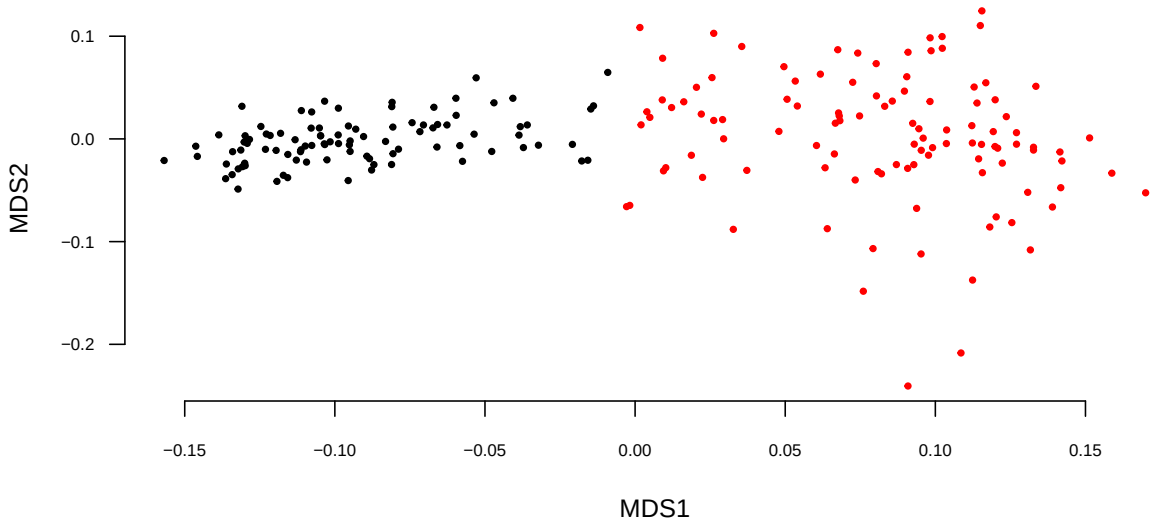
Next, we analyse population structure by means of genomic-kinship:

```
autosomal <- which(data.qc1@gtdata@chromosome != 39)
data.qc1.gkin <- ibs(data.qc1, snpsubset = autosomal, weight = 'freq')
data.qc1.dist <- as.dist(0.5 - data.qc1.gkin)
data.qc1.mds <- cmdscale(data.qc1.dist)
plot(data.qc1.mds, pch=19, cex=.5, las=1, xlab="MDS1", ylab="MDS2", cex.axis=.7, bty='n')
```



We can see that there is possible population structure here. We should investigate this further, but for our purposes, let's just run simple K-means clustering with the number of clusters *a priori* set to $K = 2$

```
kclust <- kmeans(data.qc1.mds, centers = 2)
plot(data.qc1.mds, pch=19, cex=.5, las=1, xlab="MDS1", ylab="MDS2", cex.axis=.7, bty='n', col=kclust$clu
```

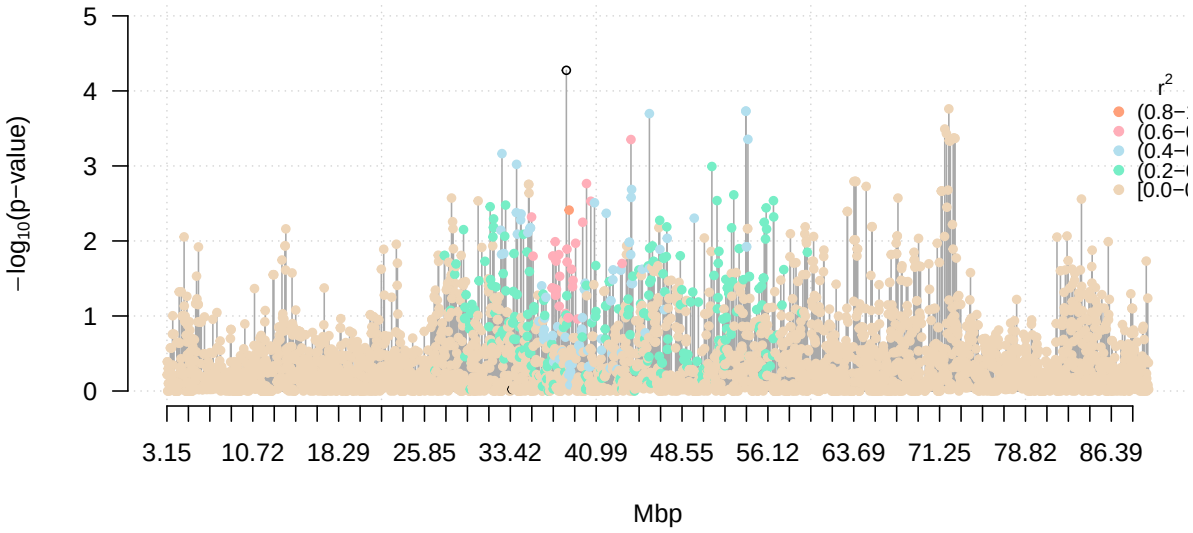



```
pop <- kclust$cluster
```

We can compare subpopulations looking at the differences in the reference allele frequency using the `pop.allele.counts` and the `plot.pac` functions. Here, we just focus on chromosome 2. # Comparing subpopulations

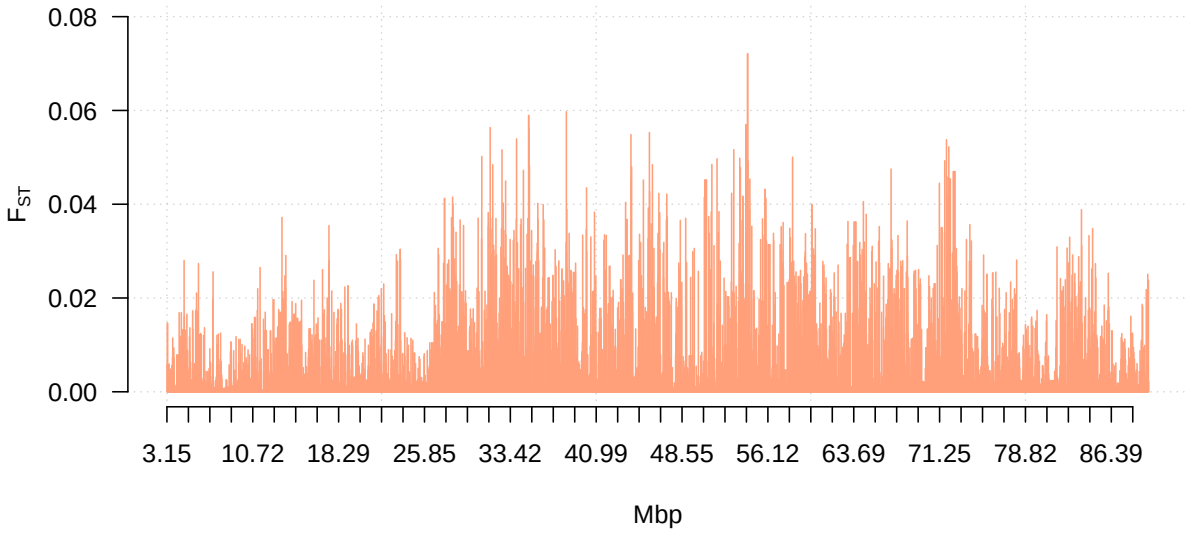
```
pac <- pop.allele.counts(data.qc1[,data.qc1@gtdata@chromosome==2], pop, progress=F)
plot.pac(data.qc1[,data.qc1@gtdata@chromosome==2], allele.cnt = pac, plot.LD = T)
```

```
## Loading required package: wesanderson
```



In a similar way, we can compute and plot fixation index F_{ST}:

```
fst <- compute.fstats(data.qc1[,data.qc1@gtdata@chromosome==2], pop)
plot.fstats(data.qc1[,data.qc1@gtdata@chromosome==2], fst)
```



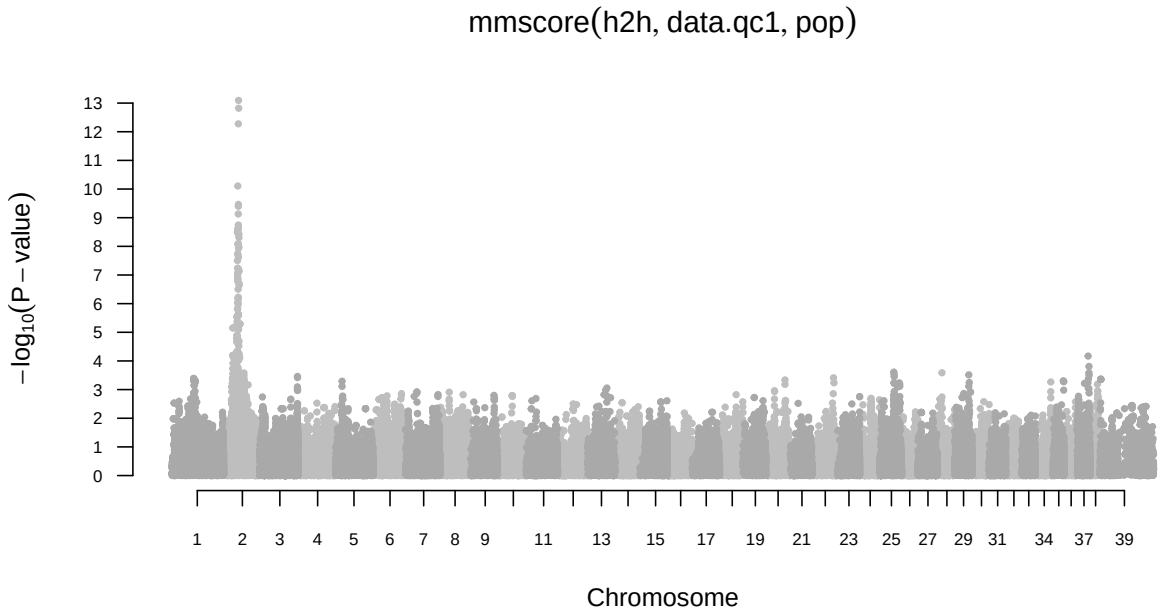
Having defined subpopulations, we can proceed to association analyses using mixed model with genomic kinship as random effect.

```
h2h <- polygenic_hglm(formula = ct ~ sex, data.qc1.gkin, data.qc1)
```

```
## Loading required package: hglm
```

```
## Loading required package: Matrix
## Loading required package: hglm.data
##
## hglm: Hierarchical Generalized Linear Models
## Version 2.0-11 (2014-10-30) installed
## Authors: Xia Shen, Moudud Alam, Lars Ronnegard
## Maintainer: Xia Shen <xia.shen@ki.se>
##
## Use citation("hglm") to know how to cite our work.
##
## Discussion: https://r-forge.r-project.org/forum/?group_id=558
## BugReports: https://r-forge.r-project.org/tracker/?group_id=558
## VideoTutorials: http://www.youtube.com/playlist?list=PLn1OmZECD-n15vnYzvJDy5GxjNpVV5Jr8
```

```
mm <- mmscore(h2h, data.qc1, strata = pop)
par(las=1, cex.axis=.7) # Tweak graphics
plot(mm, cex=.5, pch=19, col=c("darkgrey","grey"))
```



As we can see, there is a very strong association signal on chromosome 2. We can examine it a bit closer using the `plot.manhattan.ld` function.

Visualization and analyses of linkage structure

Say, we would like to zoom in on chromosome 2 and visualise LD to the top-associated marker. First, we need the name and cooridinates of the marker:

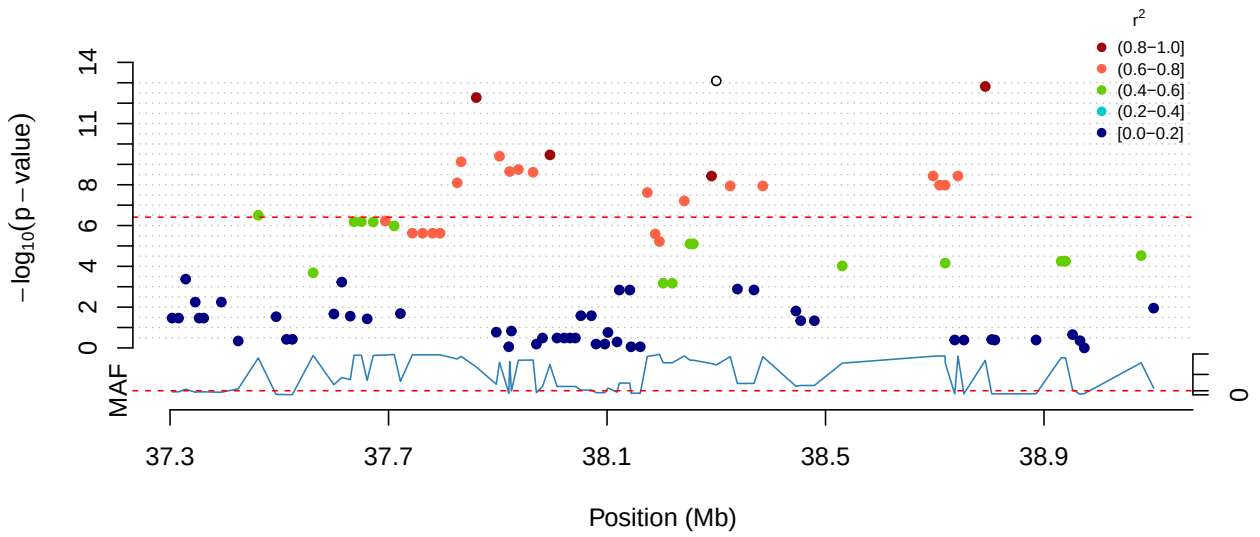
```
summary(mm, top=1)
```

```
## Summary for top 1 results, sorted by P1df
```

```
##                Chromosome Position Strand A1 A2   N     effB   se_effB
## BICF2S2365880           2 38256927      u  T  C 205 1.095257 0.1466398
##                chi2.1df         P1df        Pc1df effAB effBB chi2.2df P2df
## BICF2S2365880 55.78642 8.078765e-14 3.540639e-13    NA    NA        0   NA
```

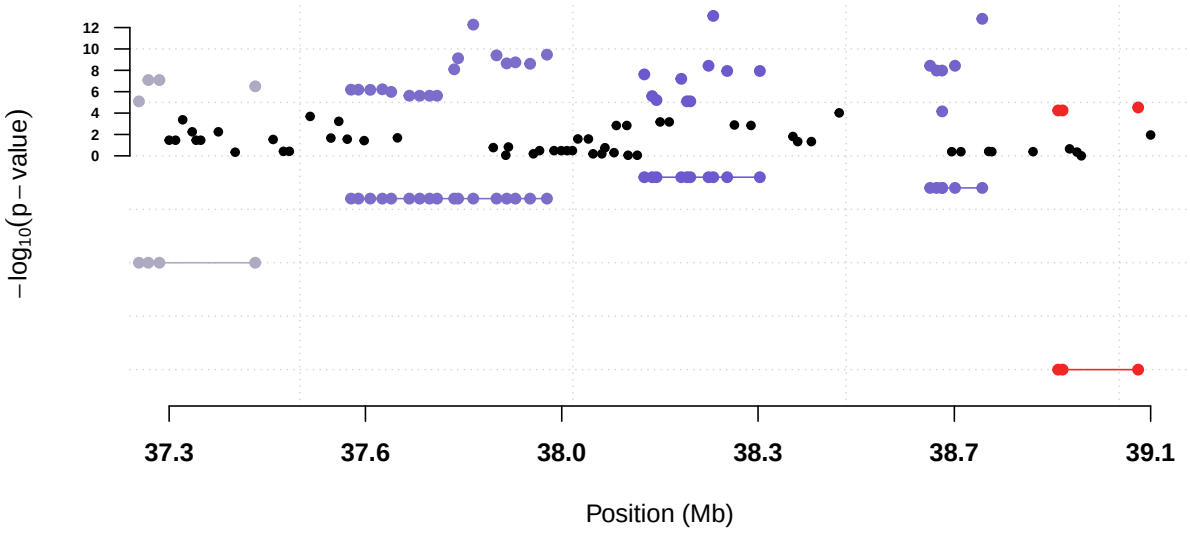
We see that the top-associated marker is **BICF2S2365880** and its position is **38256927bp**. We will zoom in on a 2Mbp region centered on the marker:

```
plot.manhattan.LD(data = data.qc1, gwas.result = mm, chr = 2, region = c(37256927,39256927), index.snp =
```



We can also use the clumping procedure as outlined in PLINK documentation [cite] to single out regions of interest.

```
clumps <- clump.markers(data, mm, chr = 2)
plot.clumps(mm, clumps, 2, c(37256927,39256927))
```

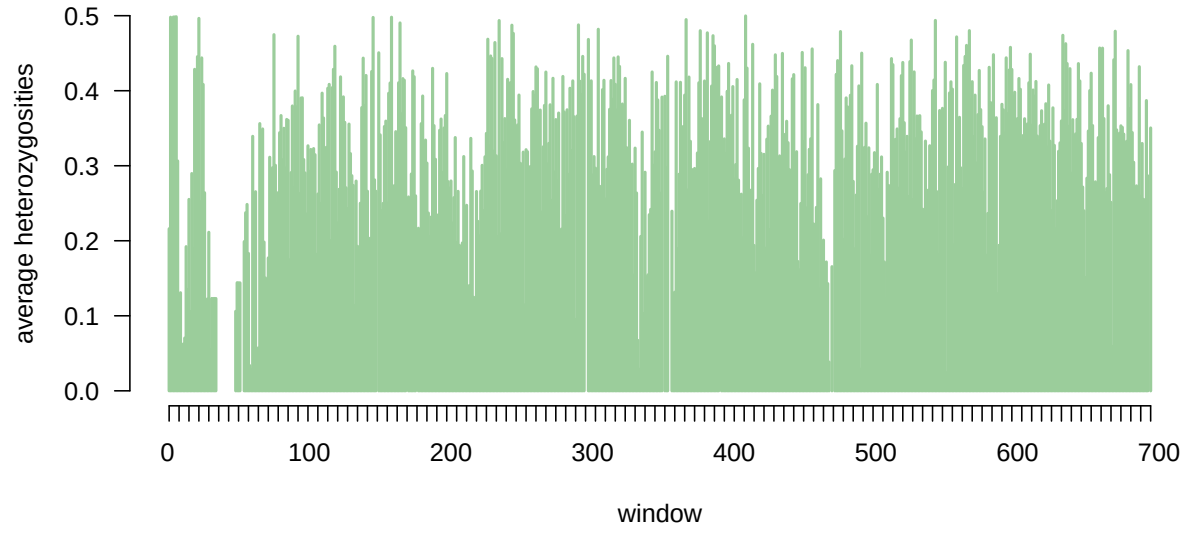


Computing average heterozygosity using overlapping windows approach.

```
LW <- get.overlap.windows(data.qc1, 2, 125e3, 2500)
het.windows <- het.overlap.wind(data.qc1, LW, F)
```

Now, having calculated average heterozygosity we can visualize them with `plot.overlap`

```
plot.overlap(LW,het.windows)
```



We can use calculated heterozygosity to examine runs of homozygosity across selected chromosome. Let's use `get.roh` and check if we have any stretches of reduced heterozygosity on chromosome 2. Below a given threshold, all windows will be treated as homozygous.

```
get.roh(data.qc1, chr=2, LW=LW, hetero.zyg=het.windows, threshold = 0.30, strict = TRUE)
```

```
##      window    begin      end length
## [1,]      8  4010290  5360290     87
## [2,]    197 27162790 28022790     54
## [3,]    316 41740290 42477790     52
## [4,]    436 56440290 57667790     77
```

As a result we get a matrix with start window, coordinates of run and length in windows

To visualise LD decay on chromosome 2, one can call the `plot.ld.decay` function.

```
plot.LD.decay(data.qc1[,data.qc1@gtdata@chromosome==2])
```