Prepare annotation of SNPs in W

Izel Fourie Sørensen, Pernille Merete Sarup, Palle Duun Rohde

April 6, 2018

In this script we prepare a data frame of the *Drosophila melanogaster* annotations. The annotations are used for preparing marker sets.

Download and read the annotation data

Variant annotation (based on FB5.49) can be found at http://dgrp2.gnets.ncsu.edu/data.html at the bottom of the page under "Other useful files".

```
download.file("http://dgrp2.gnets.ncsu.edu/data/website/dgrp.fb549.annot.txt",
              destfile = "C:/Users/Izel/Dropbox/qgg-usersguide/data/dgrp.fb549.annot.txt")
```

```
annotation <- read.table(file = "./data/dgrp.fb549.annot.txt", sep="\t",
                         colClasses="character", quote="")
```

The file is tab separated. The separators used in the site class column (column 3) are explained here: http://dgrp2.gnets.ncsu.edu/faq.html under item "3. What are the output files?; GeneAnnotation".

Look at the data

```
dim(annotation)
```

```
## [1] 4438427       4
```

```
str(annotation)
```

```
## 'data.frame':    4438427 obs. of  4 variables:
##  $ V1: chr  "2L_10000016_SNP" "2L_10000023_SNP" "2L_10000029_SNP" "2L_10000033_SNP" ...
##  $ V2: chr  "C" "C" "G" "G" ...
##  $ V3: chr  "SiteClass[FBgn0051875|CG31875|INTRON|0;FBgn0051755|SoYb|NON_SYNONYMOUS_CODING|0],Transc
##  $ V4: chr  "-" "-" "-" "-" ...
```

Edit the annotation data frame

The first column of the `annotation` data frame is used as row names. `as.character()` is used to avoid factor levels.

```
rownames(annotation) <- as.character(annotation[,1])
```

With `sapply` split each of the row names (e.g. 2L_10000016_SNP) at the underscore "_". Resulting in e.g. 2L 10000016 SNP. The resulting third element of the row name (`[3]`, here"SNP") is saved in a vector called `vtype` (variant type).

```
vtype <- sapply(rownames(annotation), function(x){
  strsplit(x,split="_")[[1]][3]
  } )
head(vtype)
```

```
## 2L_10000016_SNP 2L_10000023_SNP 2L_10000029_SNP 2L_10000033_SNP
##           "SNP"           "SNP"           "SNP"           "SNP"
## 2L_10000089_SNP 2L_10000133_SNP
##           "SNP"           "SNP"
```

Rownames of the `annotation` data frame (SNP ids) are kept as names for each element in the `vtype` vector. See:

```
head(names(vtype))
```

```
## [1] "2L_10000016_SNP" "2L_10000023_SNP" "2L_10000029_SNP" "2L_10000033_SNP"
## [5] "2L_10000089_SNP" "2L_10000133_SNP"
```

Look at unique terms in `vtype`.

```
unique(vtype)
```

```
## [1] "SNP" "DEL" "INS" "MNP"
```

`snpA_raw` is a vector containing the information from column 3 (site class) of data frame `annotation`, including only SNPs. Look at the first element of the `snpA_raw` vector.

```
snpA_raw <- as.character(annotation[vtype=="SNP",3])
length(snpA_raw)
```

```
## [1] 3963420
```

```
snpA_raw[1]
```

```
## [1] "SiteClass[FBgn0051875|CG31875|INTRON|0;FBgn0051755|SoYb|NON_SYNONYMOUS_CODING|0],TranscriptAnno
```

Give names to the `snpA_raw` vector: the row names of the `annotation` data frame, i.e. SNP id. Remove the redundant suffix "_SNP" from the names of the vector, such that it only contains the SNP id.

```
names(snpA_raw) <- gsub("_SNP","",rownames(annotation)[vtype=="SNP"])
```

Only the "SiteClass" information (i.e. information in the square brackets directly following "SiteClass") is used. This includes flybase gene id, gene symbol, mapped sequence ontology (site class) and base pair distance to gene and is separated by "," from the transcript annotation information (see: http://dgrp2.gnets.ncsu.edu/faq.html).

Select the site class information by splitting `snpA_raw` at "," and keep the first string. Remove "SiteClass" and square brackets from this string and split the string at the semicolons.

```
snpA_raw <- lapply(snpA_raw, function(x) {
 x <- strsplit(x,",")[[1]][1]
 x <- gsub("SiteClass","",x)
 x <- gsub("[","",x, fixed=TRUE)
 x <- gsub("]","",x, fixed=TRUE)
 x <- strsplit(x,";")[[1]]
 x
  })

head(snpA_raw)
```

```
## $`2L_10000016`
## [1] "FBgn0051875|CG31875|INTRON|0"
## [2] "FBgn0051755|SoYb|NON_SYNONYMOUS_CODING|0"
##
## $`2L_10000023`
```

```
## [1] "FBgn0051875|CG31875|INTRON|0"
## [2] "FBgn0051755|SoYb|NON_SYNONYMOUS_CODING|0"
##
## $`2L_10000029`
## [1] "FBgn0051875|CG31875|INTRON|0"
## [2] "FBgn0051755|SoYb|NON_SYNONYMOUS_CODING|0"
##
## $`2L_10000033`
## [1] "FBgn0051875|CG31875|INTRON|0"
## [2] "FBgn0051755|SoYb|SYNONYMOUS_CODING|0"
##
## $`2L_10000089`
## [1] "FBgn0051875|CG31875|INTRON|0"
## [2] "FBgn0051755|SoYb|NON_SYNONYMOUS_CODING|0"
##
## $`2L_10000133`
## [1] "FBgn0051875|CG31875|INTRON|0"
## [2] "FBgn0051755|SoYb|SYNONYMOUS_CODING|0"
```

```
length(snpA_raw)
```

```
## [1] 3963420
```

Create a vector (`nA_raw`) which contains the length of each of the SNPs (the number of separated segments per SNP) in `snpA_raw`.

```
nA_raw <- sapply(snpA_raw,length)
```

`table(nA_raw)` shows the number of SNP loci that contain 1, 2, 3 or up to 12 separate annotations in flybase.

```
table(nA_raw)
```

```
## nA_raw
##       1       2       3       4       5       6       7       8       9
## 3241772  602432  100328   13818    2828     963     494     440     170
##      10      11      12
##     130      27      18
```

Look at the first three of 18 elements in `nA_raw` that has a length of e.g. 12.

```
nA_raw[nA_raw==12][1:3]
```

```
## 2L_20418974 2L_20418976 2L_20418995
##          12          12          12
```

The vector `snpNames` contains the names of snpA_raw repeated as many times as they appeared in nA_raw, i.e. once for each flybase annotation.

```
snpNames <- rep(names(snpA_raw), times=nA_raw)
length(snpNames)
```

```
## [1] 4833131
```

Unlist snpA_raw and remove the names of the vector so that the vector only contains the separated segments. Unlisting `snpA_raw` creates a vector with the same length as `nA_raw`.

```
snpA_raw <- unlist(snpA_raw, use.names=FALSE)
head(snpA_raw)
```

```
## [1] "FBgn0051875|CG31875|INTRON|0"
## [2] "FBgn0051755|SoYb|NON_SYNONYMOUS_CODING|0"
## [3] "FBgn0051875|CG31875|INTRON|0"
## [4] "FBgn0051755|SoYb|NON_SYNONYMOUS_CODING|0"
## [5] "FBgn0051875|CG31875|INTRON|0"
## [6] "FBgn0051755|SoYb|NON_SYNONYMOUS_CODING|0"
```

```
length(snpA_raw)
```

```
## [1] 4833131
```

Split the segments of snpA_raw at the "|" symbol. The result is a list where the element names correspond to the separated segments prepared above. "GBgn" (flybase gene id), gene id, mapped sequence ontology terms ("SeqOnt") and base pair distance to gene ("distance") is the contents of each element. As earlier the vector `nA_raw` gives the length of each of the elements (the number of separated segments for each element) in `snpA_raw`. `table(nA_raw)` shows that 919806 elements in `snpA_raw` contain 3 segments and 3913325 elements contain 4 segments.

```
snpA_raw <- sapply(snpA_raw,function(x) {  unlist(strsplit(x,split="|",fixed=TRUE))  })
nA_raw <- sapply(snpA_raw,length)
table(nA_raw)
```

```
## nA_raw
##       3       4
##  919806 3913325
```

Create an empty matrix `snpA`, the number of rows equal to the length of snpNames and 4 columns. The `snpNames` vector is used as the row names. This corresponds to element names of the `snpA_raw` list as prepared earlier.

```
snpA <- matrix(NA,nrow=length(snpNames),ncol=4)
rownames(snpA) <- snpNames
head(snpA)
```

```
##             [,1] [,2] [,3] [,4]
## 2L_10000016   NA   NA   NA   NA
## 2L_10000016   NA   NA   NA   NA
## 2L_10000023   NA   NA   NA   NA
## 2L_10000023   NA   NA   NA   NA
## 2L_10000029   NA   NA   NA   NA
## 2L_10000029   NA   NA   NA   NA
```

Unlist `snpA_raw`. This creates a vector of "FBgn", gene names, mapped sequence ontology terms ("SeqOnt") and base pair distance to gene ("distance"). By transposing the vector, the four different types of information are each put into a column of snpA. Only the elements of snpA_raw that contain 4 segments (thus the elements of `nA_raw` with length 4) are included.

```
snpA[nA_raw==4,] <- t(sapply(snpA_raw[nA_raw==4],function(x) {  unlist(x) }))
head(snpA)
```

```
##             [,1]          [,2]      [,3]                    [,4]
## 2L_10000016 "FBgn0051875" "CG31875" "INTRON"                "0"
## 2L_10000016 "FBgn0051755" "SoYb"    "NON_SYNONYMOUS_CODING" "0"
## 2L_10000023 "FBgn0051875" "CG31875" "INTRON"                "0"
```

```
## 2L_10000023 "FBgn0051755" "SoYb"    "NON_SYNONYMOUS_CODING" "0"
## 2L_10000029 "FBgn0051875" "CG31875" "INTRON"                "0"
## 2L_10000029 "FBgn0051755" "SoYb"    "NON_SYNONYMOUS_CODING" "0"
```

The final annotation data frame will only include SNPs that are in the centered and scaled genotype matrix (**W**), as prepared earlier in the qgg user guide. Here we load the `W` matrix.

```
load(file="./genotypes/dgrp2_W2.Rdata")
```

Prepare a logical vector, `inW` . `TRUE` depends on whether the SNPs in this dataset is also found in `W`.

```
inW <- rownames(snpA)%in%colnames(W)
```

Keep only SNPs that were `TRUE` in the logical vector `inW`.

```
snpA <- snpA[inW,]
```

Give the data frame snpA relevant column names.

```
colnames(snpA) <- c("FBid","GeneName","SeqOnt","distance")
snpA[1:5,]
```

```
##             FBid          GeneName  SeqOnt                  distance
## 2L_10000016 "FBgn0051875" "CG31875" "INTRON"                "0"
## 2L_10000016 "FBgn0051755" "SoYb"    "NON_SYNONYMOUS_CODING" "0"
## 2L_10000033 "FBgn0051875" "CG31875" "INTRON"                "0"
## 2L_10000033 "FBgn0051755" "SoYb"    "SYNONYMOUS_CODING"     "0"
## 2L_10000089 "FBgn0051875" "CG31875" "INTRON"                "0"
```

Save the snpA matrix

```
save(snpA, file="./annotation/snpA_W2.Rdata")
```