Integration
Self-Study Exercises

Bioconductor Team

Fred Hutchinson Cancer Research Center

17-18 February, 2011

1 Introduction

These exercises pull together the various aspects of data manipulation and class construction and methods development covered in this the Advanced *R* course.

Exercise 1
In this section you will compute linkage disequilibrium on the snp data, filter out snps based on a particular result value and annotate the remaining snps.

One of the challenges of this exercise is to construct your function such that linkage disequilibrium is computed for all snps in the dataset (except the last 'width' number of snps). Note that of the snps input to the `.cld` *function, linkage disequilibrium results are only computed for (# snps - width) snps. This is because the number of linkage disequilibrium comparisons made are equal to the* `width` *argument. Thus if 100 snps are input with a width of 5, the output will be a matrix of the first 95 snps (one row per snp) by 5 columns. The 5 columns represent the comparisons with the five snps to the right of the snp.*

At your disposal you have the `GWASdata` *class that can hold the snp data in the* `AdvancedR2011Data` *package. Also useful may be the method for composite linkage disequilibrium which accepts the* `GWASdata` *as input. See '?cld' and the code defining the generic and method in* `cld.R`.

```
> library(AdvancedR2011)
> library(StudentGWAS)
> dataPath <- system.file("extdata", "snpData.nc",
+                         package="AdvancedR2011Data")
> metadataPath <- system.file("extdata", "metadata.sqlite",
+                             package="AdvancedR2011Data")
> ## The GWASdata class
> data <- GWASdata(dataPath, metadataPath)
> ## The cld method for GWASdata
> res <- cld(data, 1, 20)
```

Question 1
Write a function that computes composite linkage disequilibrium on the snpData.nc file. Iterate through the data in chunks. Filter out snps that have a linkage disequilibrium result greater than 0.1 with any of its neighbors within the specified width. A subset of the filtered snps will be annotated in the next exercise.

Solution:

Exercise 2
At this point a natural question to wonder about is what all these SNPs actually are? Previous to this point, you created a metadata database that mapped the SNPs onto their respective entrez gene IDs (where applicable). In case you ever need to know, the mapping of these SNPs and onto their Entrez Gene IDs was already done for you by using the dbSNP anntoation package along with the GenomicFeatures package. To get other relevant information, you have several options to explore which I will describe briefly here. 1) You could make use of the annotation packages such as org.Hs.eg.db. We didn't cover this in this course since it is an introductory topic, but that also means that it is pretty easy to do. Basically you can just load up a package, and make use of the provided mapping objects. So here is one quick example of how you can do this sort of approach:

```
> ## load a lib
> library(org.Hs.eg.db)
> ## list the mappings:
> head(ls("package:org.Hs.eg.db")) ## etc.

[1] "org.Hs.eg"
[2] "org.Hs.egACCNUM"
[3] "org.Hs.egACCNUM2EG"
[4] "org.Hs.egALIAS2EG"
[5] "org.Hs.egCHR"
[6] "org.Hs.egCHRLENGTHS"

> ## The mappings act like subsettable R objects so you can subset:
> org.Hs.egPATH[c("10","100")]

PATH submap for Human (object of class "AnnDbBimap")

> ## or you can convert that subsetted mapping to a data.frame:
> toTable(org.Hs.egPATH[c("10","100")])

  gene_id path_id
1      10   00232
2      10   00983
3      10   01100
4     100   00230
5     100   01100
6     100   05340
```

```
> ## and you can merge that with other data using merge() etc.
> merge(toTable(org.Hs.egPATH[c("10","100")]),
+       toTable(org.Hs.egSYMBOL[c("10","100")]),
+       by.x ="gene_id", by.y="gene_id",
+       all.x=TRUE, all.y=TRUE)

  gene_id path_id symbol
1      10   00232   NAT2
2      10   00983   NAT2
3      10   01100   NAT2
4     100   00230    ADA
5     100   01100    ADA
6     100   05340    ADA
```

OR you could also do the more advanced approach that we taught you about before in the final exercise and use a cross database SQL join to get your data. This is more difficult than simply using an annotation package, but it also does not require that you have a standard org package in order to use it. So any SQLite database that contains the data you need should work with this approach.

Question 2
Using the getSnps() method that you developed for the StudentGWAS package, retrieve the annotations for all the SNPs that you searched and then subset that using the result from before. After this 1st step, you should have a data.frame containing the SNPs of interest mapped to the entrez gene IDs. And now you can choose how advanced you want to go and do one (or all) of the strategies described above to additionally map these SNPs to gene symbols. To speed things up for those of you who are interested in the 2nd option, be aware that the gene symbols are stored in the gene_info table. Also, it is a good idea while working on this to initially work with just a subset (maybe the 1st 10 or so) SNPs until you get everything working. Then annotate the whole set.

Solution: