Replicating selected results of Cheung 2010: Combining hgfocus arrays with HapMap (phase 3, r3) genotypes

VJ Carey

April 13, 2011

Contents

1 Introduction

GEO GSE16778 is a collection of 695 Affymetrix HG-Focus arrays run on RNA from immortalized B-cells from the CEPH CEU cohort. Our objective is to connect the transcript abundance data in these arrays to DNA variation data in HapMap genotype archives, and then to reproduce a finding of expression variation determined by a distal SNP. Specifically, table 1 of Cheung asserts that expression of PECAM1 on chromosome 17 is associated with the number of C alleles in SNP rs2074981, which is on chromosome 19.

2 Obtaining subject identifiers from GEO GSE metadata

We will assume that the GM number from the GEO documentation of the arrays corresponds to the NA number used in HapMap labeling. We can get all the GM numbers as follows. We assume that the Bioconductor GEOmetadb SQLite database has been obtained with code like the following:

```
> library(GEOmetadb)
> geometadbfile <- getSQLiteFile()  # writes 1.4Gb to current working folder
> con = dbConnect(SQLite(), geometadbfile)
```

In this case, the `geometadbfile` is a string naming a local file, so the live connection for this document is:

```
> library(RSQLite)
> con = dbConnect(SQLite(), "GEOmetadb.sqlite")
> dbListTables(con)
```

```
 [1] "gds"               "gds_subset"        "geoConvert"
 [4] "geodb_column_desc" "gpl"               "gse"
 [7] "gse_gpl"           "gse_gsm"           "gsm"
[10] "metaInfo"          "sMatrix"
```

We can get metadata on all the CEL files via

```
> allcel16778 = dbGetQuery(con, "select * from gsm where series_id = 'GSE16778'")
```

and then retrieve the description fields.

```
> alldesc = allcel16778$desc
```

The description fields are often quite messy but are reasonably systematic in this series.

```
> alldesc[1]
[1] "http://ccr.coriell.org/Sections/Search/Sample_Detail.
        aspx?Ref=GM06980&PgId=166;\tGene expression data at baseline
        from lymphoblastoid cells of individual GM06980."
```

We can extract the important GM number and transform to the NA identifier:

```
> gnum = gsub(".*individual (.*).", "\\1", alldesc)
> nanum = gsub("GM", "NA", gnum)
> nanum[1:10]
```

```
 [1] "NA06980" "NA06981" "NA06982" "NA06983" "NA06985" "NA06985" "NA06986"
 [8] "NA06987" "NA06987" "NA06988"
```

3 Retrieving HapMap genotypes and associated identifiers

The *chopsticks* package has facilities for direct import from HapMap. We will use the file

```
http://hapmap.ncbi.nlm.nih.gov/downloads/genotypes/
   hapmap3_r3/hapmap_format/polymorphic/
   genotypes_chr19_CEU_phase3.3_nr.b36_fwd.txt.gz
```

which we assign to variable `fn`. We read the data and obtain the identifiers:

```
> library(chopsticks)
> if (!exists("c19")) c19 = read.HapMap.data(fn)
> hids = rownames(c19[[1]])
> hids[1:10]

 [1] "NA06989" "NA06984" "NA12341" "NA12340" "NA12336" "NA12343" "NA12335"
 [8] "NA12342" "NA12146" "NA12239"
```

4 Connecting the two resources

The number of IDs shared between GSE16778 and this HapMap import is

```
> length(intersect(hids, nanum))

[1] 149
```

Some of the arrays are replicated:

```
> table(table(nanum[nanum %in% intersect(hids,nanum)]))

 1  2 
53 96 
```

The replication was conducted to allow an estimate of variance in transcript abundance measurement, for gene filtering purposes (p11). For exploratory statistics selection of a single replicate may be adequate.

```
> ok = which(nanum %in% hids)
> length(ok)

[1] 245

> ok2 = intersect(grep("rep1", allcel16778$title), ok)
> length(ok2)

[1] 149

> gsms = allcel16778$gsm[ok2]
```

Additional tasks required to build the expression set include:

```
> cf = paste(gsms, ".CEL.gz", sep="")
> setwd("../CELFILES")
> library(affy)
> # two of the files didn't exist
> options(verbose=TRUE); celwhap = ReadAffy(filenames=cf[-c(146,147)], verbose=TRUE)
> library(affyio)
> hh = lapply(cf[-c(146,147)], function(x)read.celfile.header(x, info="full"))
> uu = sapply(hh, function(x) x$ScanDate)
> library(chron)
> uuu = strsplit(uu, " ")
> ds = sapply(uuu,"[",1)
> ts = sapply(uuu,"[",2)
> ccc = chron(ds,ts)
> dates(ccc)
> table(dates(ccc))
> rmawhap = rma(celwhap)
> save(rmawhap, file="rmawhap.rda")
> rmagsms = gsub(".CEL.gz", "", sampleNames(rmawhap))
> match(rmagsms, allcel16778$gsm) -> gsminrma
> summary(gsminrma)
> nainrma = nanum[gsminrma]
> all(nainrma %in% hids)
```