Using the DNaseI hypersensitivity data from encode in R

VJ Carey

April 15, 2008

1 Introduction

Annotation tracks from UCSC hg18 can be used with Bioconductor to help establish genomic contexts of events or alterations. The CD4-based hypersensitivity assays are collected in the structure rawCD4 in package encoDnaseI:

```r
> library(encoDnaseI)
> data(rawCD4)
> rawCD4
```

hg18track (storageMode: lockedEnvironment)
assayData: 382713 features, 1 samples
  element names: dataVals
phenoData
  sampleNames: 1
  varLabels and varMetadata description: none
featureData
  featureNames: 1, 2, ..., 382713 (382713 total)
  fvarLabels and fvarMetadata description:
    bin: given bin
    chrom: chr..
    chromStart: numeric origin
    chromEnd: numeric close
experimentData: use 'experimentData(object)'
  pubMedIds: 16791207
Annotation:

At present, we can subset the data by casting a chromosome number:

```r
> c19g = rawCD4[chrnum(19)]
> c19g
```
hg18track (storageMode: lockedEnvironment)
assayData: 11158 features, 1 samples
  element names: dataVals
phenoData
  sampleNames: 1
  varLabels and varMetadata description: none
featureData
  featureNames: 129572, 129573, ..., 140729 (11158 total)
  fvarLabels and fvarMetadata description:
    bin: given bin
    chrom: chr..
    chromStart: numeric origin
    chromEnd: numeric close
experimentData: use 'experimentData(object)'
  pubMedIds: 16791207
Annotation:

  And we can get a trace of values along the chromosome:

  > c19gxy = getTrkXY(c19g)
  > plot(c19gxy)
2 Coupling the DnaseI series to genetics of gene expression

We would like to subset a racExSet from GGdata and look at snps that are in regions of high DNaseI sensitivity. Some infrastructure to help with this is:

```r
> clipSnps = function(sms, chrn, lo, hi) {
+   allp = getSnpLocs(sms)
+   allp = allp - allp[1]
+   ok = allp >= lo & allp <= hi
+   thesm = smList(sms)[[1]]
+   rsn = colnames(thesm)
+   rid = rsn[which(ok)]
+   thesm = thesm[, rid, drop = FALSE]
+   nn = new.env()
+   tmp = list(thesm)
```
+ names(tmp) = as.character(chrn)
+ assign("smList", tmp, nn)
+ sms@smlEnv = nn
+ sms@activeSnpInds = which(ok)
+ sms
+ }
> rangeX = function(htrk) {
+ range(getTrkXY(htrk)$x)
+ }

So we get the information on expression and SNPs in chr19g and filter:

> library(GGtools)
> library(GGdata)

GGdata loading...

> data(hmceuB36)
> rs19g = rangeX(c19g)
> c19gf = clipSnps(hmceuB36[chrnum(19), ], chrnum(19), rs19g[1],
+ rs19g[2])
> c19gf

snp.matrix-based genotype set:
number of samples:  90
number of snp.matrix:  1
annotation:
  exprs: illuminaHumanv1.db
  snps: snp locs package: GGdata ; ncdf ref: GGdata_hmceuLocs.nc
Expression data:  47293 x 90
Phenodata: An object of class "AnnotatedDataFrame"
  rowNames: NA06985, NA06991, ..., NA12892 (90 total)
  varLabels and varMetadata description:
    famid: hapmap family id
    persid: hapmap person id
    ...
    isAdad: logical TRUE if person is a father
    (9 total)

A gene-specific screen can be computed as follows:

> smxi1 = gwSnpScreen(genesym("MXI1"), c19gf, chrnum(19))

[1] "GI_18641367-A" "GI_18641367-I" "GI_18641369-I"
> plot(smxil)

We’d like to look at the SNP screen results juxtaposed with the DnaseI results.

> print(juxtaPlot(c19g, smxil))
Another example:

```r
> sOSR2 = gwSnPScreen(genesym("OSR2"), c19gf, chrnum(19))
> print(juxtaPlot(c19g, sOSR2))
```
We can score the highly associated SNPs for closeness to a highly DNaseI sensitive region using ALICOR:

> ALICOR(sOSR2, c19g)

[1] 0.3453289

> ALICOR(smx11, c19g)

[1] -0.339013

> if (interactive()) {
+   if (!exists("mads"))
+     mads = apply(exprs(c19gf), 1, mad)
+   if (interactive())
+     fn = featureNames(c19gf)[which(mads > quantile(mads, +
+       0.6))]
+   if (!interactive())
With these scores, we can find gene-snp combinations for which association is at least partly synchronized with DHS. Algorithms for systematically assessing synchronicity are in development.