

Using the DNaseI hypersensitivity data from encode in R

VJ Carey

January 17, 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`:

```
> 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:
```

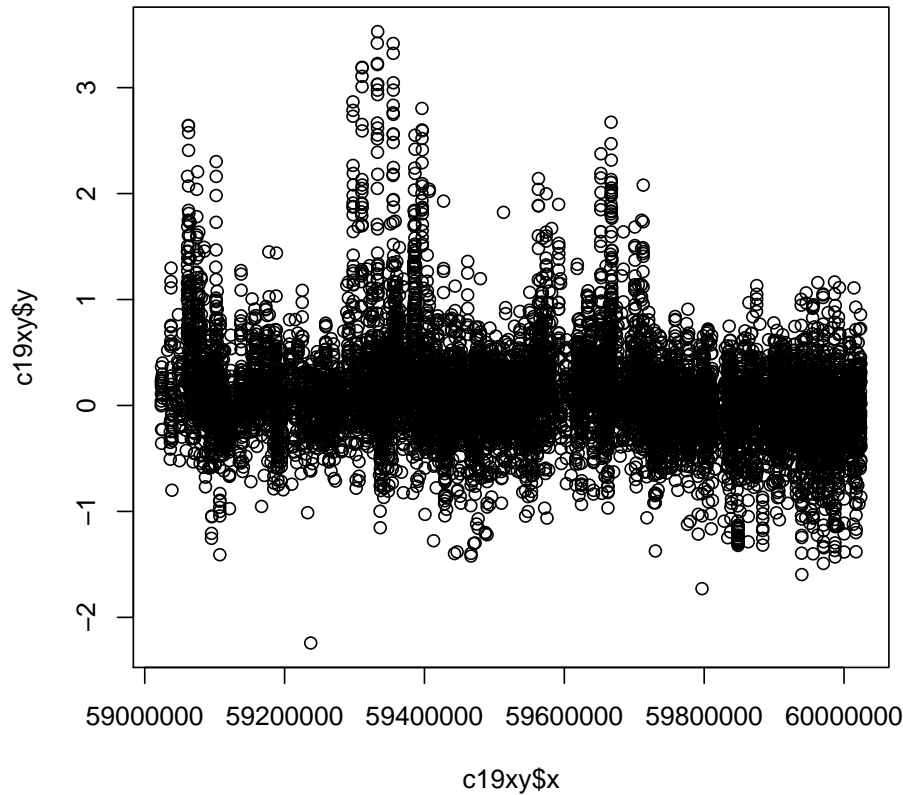
At present, we can subset the data by casting a chromosome number:

```
> c19 = rawCD4[chrnum(19)]
> c19
```

```
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:

```
> c19xy = getTrkXY(c19)
> plot(c19xy)
```



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:

```
> clipSnps = function(rexset, chrmeta, lo, hi) {
+   allp = pos(chrmeta)
+   ok = allp >= lo & allp <= hi
+   rid = names(allp[ok])
+   rexset[snpID(rid), ]
+ }
> rangeX = function(htrk) {
+   range(getTrkXY(htrk)$x)
+ }
```

So we get the information on expression and SNPs in chr19 and filter:

```
> library(GGtools)
> library(GGdata)
> data(chr19GGceuRMA)
> data(chr19meta)
> rs19 = rangeX(c19)
> c19f = clipSnps(chr19GGceuRMA, chr19meta, rs19[1], rs19[2])
> c19f
```

racExSet instance (SNP rare allele count + expression)

rare allele count assayData:

Storage mode: lockedEnvironment

featureNames: rs7258432, rs7259148, ..., rs678846, rs581623 (987 total)

Dimensions:

 racs

Features 987

Samples 58

expression assayData

Storage mode: lockedEnvironment

featureNames: 1007_s_at, 1053_at, ..., AFFX-r2-P1-cre-3_at, AFFX-r2-P1-cre-5_at (8793 total)

Dimensions:

 exprs

Features 8793

Samples 58

phenoData

An object of class "AnnotatedDataFrame"

rowNames: NA06985, NA06993, ..., NA12892 (58 total)

varLabels and varMetadata description:

sample: hapmap id

Experiment data

Experimenter name: Cheung VG

Laboratory: Department of Pediatrics, University of Pennsylvania, Philadelphia, Penns

Contact information:

Title: Mapping determinants of human gene expression by regional and genome-wide assoc

URL:

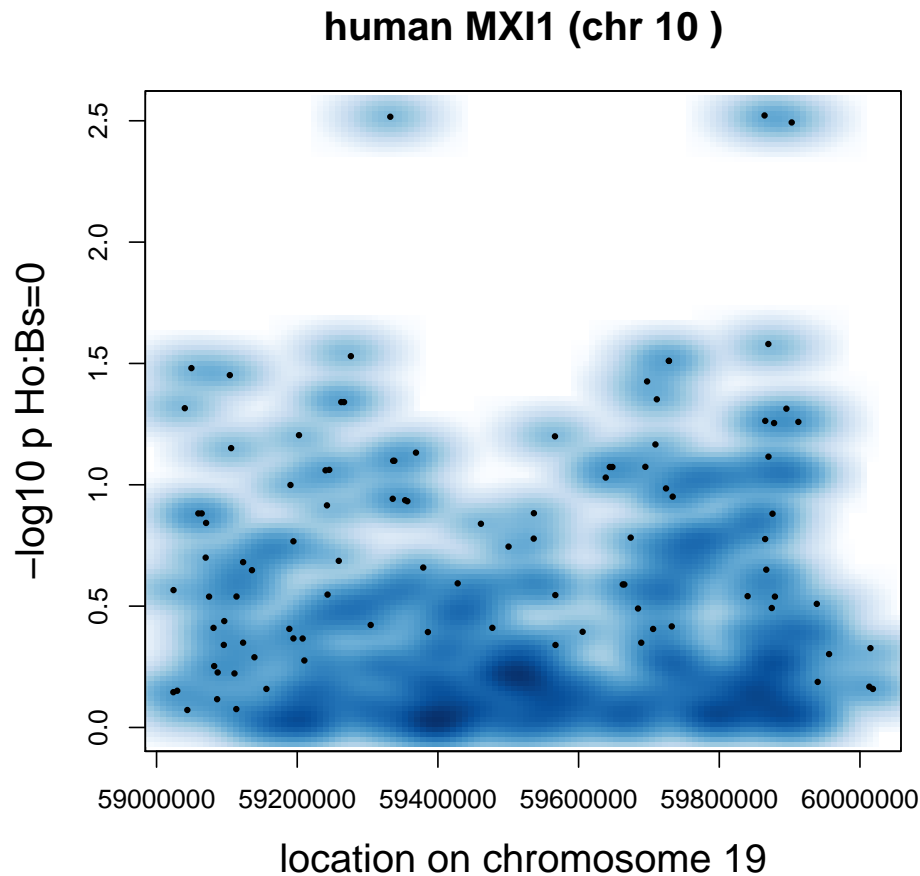
PMIDs: 16251966

Abstract: A 180 word abstract is available. Use 'abstract' method.

Annotation [1] "hgfocus"

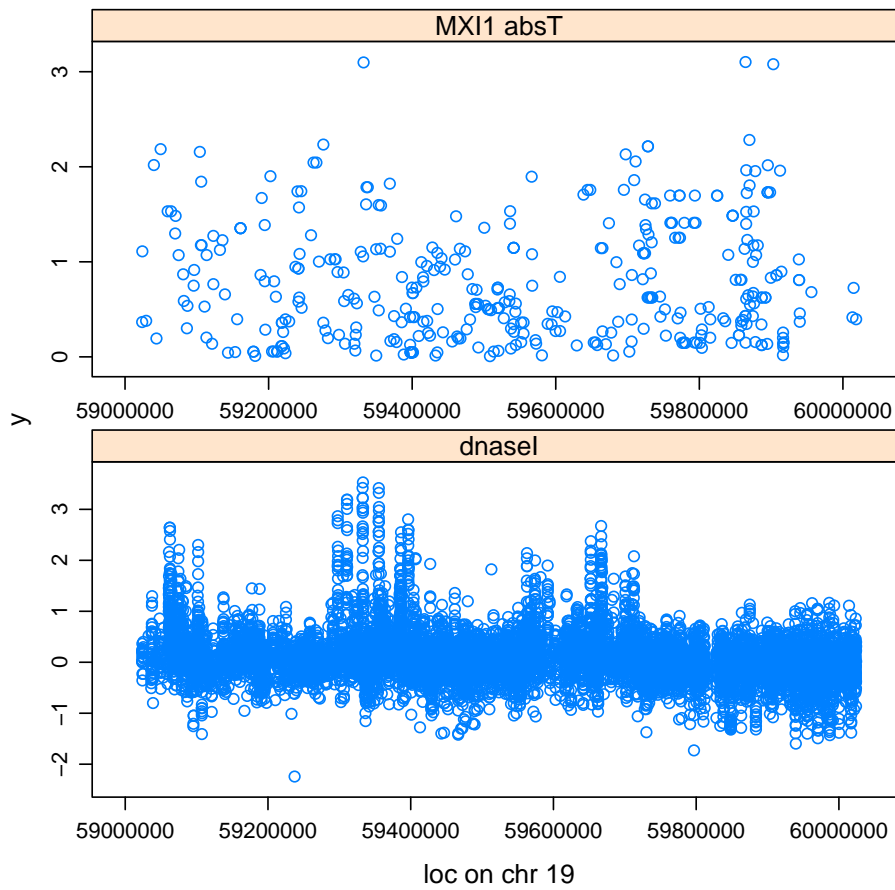
A gene-specific screen can be computed as follows:

```
> smxi1 = snpScreen(c19f, chr19meta, genesym("MXI1"), ~., fastAGMfitter)
> plot_mlp(smxi1, chr19meta, "10")
```



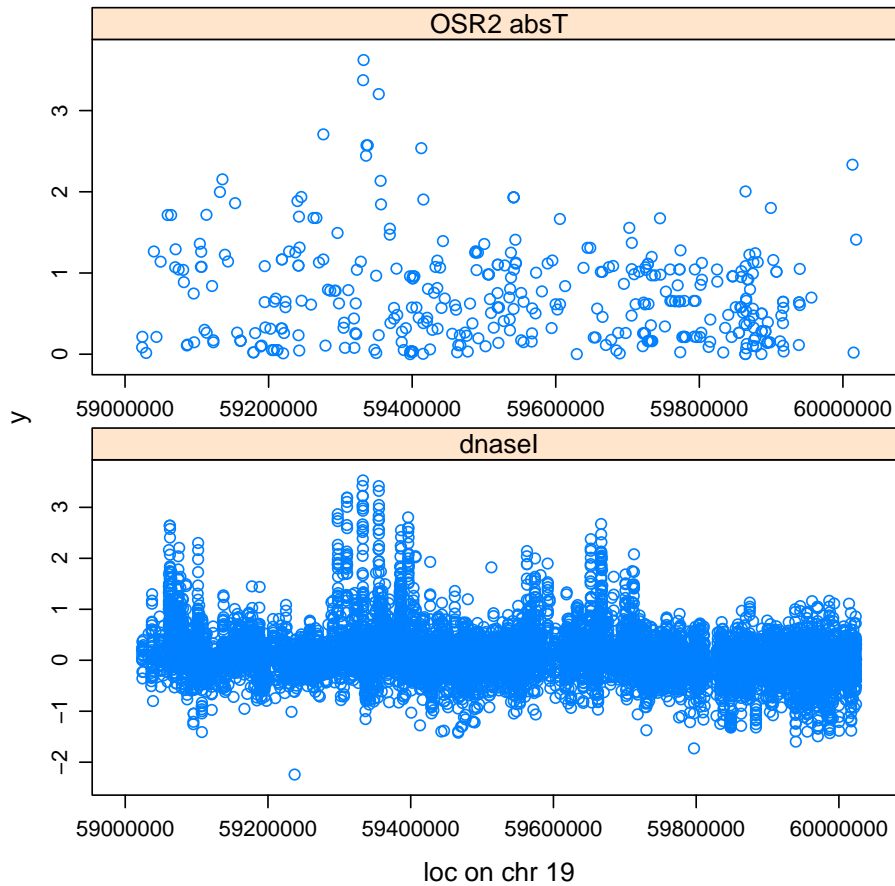
We'd like to look at the SNP screen results juxtaposed with the DnaseI results.

```
> print(juxtaPlot(c19, smxi1))
```



Another example:

```
> sOSR2 = snpScreen(c19f, chr19meta, genesym("OSR2"), ~., fastAGMfitter)
> print(juxtaPlot(c19, sOSR2))
```



We can score the highly associated snps for closeness to a highly DnaseI sensitive region using ALICOR:

```
> ALICOR(sOSR2, c19)
```

```
[1] 0.7993318
```

```
> ALICOR(smx11, c19)
```

```
[1] 0.2340942
```

To do this in the context of a transcriptome wide screen, we filter the genes in the expression set non-specifically.

The following code takes about 15 minutes to run on a decent linux box, and is suppressed from the standard build.

```
> if (interactive()) {
+   if (!exists("mads"))
```

```

+      mads = apply(exprs(c19f), 1, mad)
+    if (interactive())
+      fn = featureNames(c19f)[which(mads > quantile(mads, 0.6))]
+    if (!interactive())
+      fn = featureNames(c19f)[which(mads > quantile(mads, 0.97))]
+    n19 = c19f[exFeatID(fn), ]
+    if (file.exists("tw19.rda"))
+      load("tw19.rda")
+    if (!exists("tw19"))
+      tw19 = twSnpScreen(n19, chr19meta, ~., fastAGMfitter)
+    if (!file.exists("tw19.rda"))
+      save(tw19, file = "tw19.rda")
+    if (file.exists("allscor.rda"))
+      load("allscor.rda")
+    if (!exists("allscor"))
+      allscor = sapply(tw19, function(x) {
+        if (inherits(x, "try-error"))
+          return(NA)
+        else return(ALICOR(x, c19))
+      })
+    if (!file.exists("allscor.rda"))
+      save(allscor, file = "allscor.rda")
+  }

```

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.