

Overview of *GGtools* for investigating genetics of gene expression

VJ Carey <stvjc@channing.harvard.edu>

May 5, 2007

1 Introduction

The *GGtools* package supports data analysis activities that link genotypic information such as SNP configurations to gene expression phenotype. We will attach the library and have a look at a basic demonstration resource.

```
> library(GGtools)
> data(chr20GGdem)
> class(chr20GGdem)
```

```
[1] "racExSet"
attr(,"package")
[1] "GGtools"
```

```
> chr20GGdem
```

```
racExSet instance (SNP rare allele count + expression)
```

```
rare allele count assayData:
```

```
Storage mode: lockedEnvironment
```

```
featureNames: rs4814683, rs6076506, ..., rs6062370, rs6090120 (117417 total)
```

```
Dimensions:
```

```
      racs
```

```
Features 117417
```

```
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

rowNames: NA06985, NA06993, ..., NA12892 (58 total)
varLabels and varMetadata:
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"

The `racExSet` class is an extension of the `eSet` class. It represents expression data from the hgfocus chip on 48 individuals in the CEU CEPH group, and SNP data obtained from their HapMap genotyping results.

The data are organized into an 8793 by 58 matrix of expression values accessible with the `exprs` method, and an 117417 by 58 of rare allele counts:

```
> dim(exprs(chr20GGdem))
```

```
[1] 8793 58
```

```
> dim(snps(chr20GGdem))
```

```
[1] 117417 58
```

```
> snps(chr20GGdem)[1:5, 1:5]
```

	NA06985	NA06993	NA06994	NA07000	NA07022
rs4814683	2	0	0	2	1
rs6076506	0	0	0	0	NA
rs6139074	2	0	0	2	1
rs1418258	2	0	0	2	1
rs7274499	0	0	0	0	NA

We need some genetic metadata about SNPs; these are derived from from SNP genotyping panels released on a chromosome-by-chromosome basis for CEPH participants by HapMap project:

```
> data(chr20meta)
> chr20meta[1:4, ]
```

	pos	strand
rs4814683	9795	+
rs6076506	11231	+
rs6139074	11244	+
rs1418258	11799	+

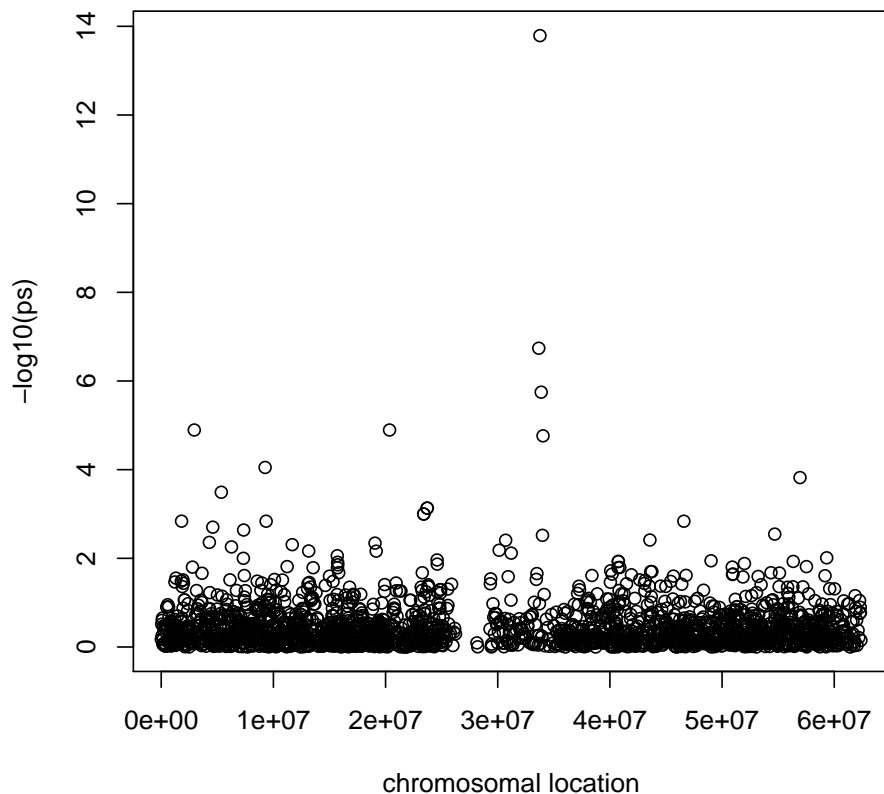
A basic task is to compute a screen (over the genome, or, more practically, if seeking interactive performance with commodity hardware, over a chromosome) of genotypic determination of expression. The `snpScreen` method helps with this; we illustrate an example related to results in Cheung and Spielman 2005:

```
> chr20GGdem = exclMono(chr20GGdem)
> S100 = snpScreen(chr20GGdem, chr20meta, genesym("CPNE1"), ~.,
+      lm, gran = 30)
> S100
```

GGtools snpScreenResult for call:
snpScreen(racExSet = chr20GGdem, snpMeta = chr20meta, gene = genesym("CPNE1"),
formTemplate = ~., fitter = lm, gran = 30)
There were 2125 attempted fits,
and 2125 were successful.

A primitive display is obtained as follows. We know that `lm` was used, so the relevant p-values are in the coefficient component of the summarized fit objects.

```
> ps = as.numeric(sapply(S100, function(x) try(summary(x)$coef[2,
+      4])))
> plot(S100@locs, -log10(ps), xlab = "chromosomal location")
```



2 Performance-oriented specialization

The `snpScreen` method illustrated above is very general (can accommodate and retain results of any R modeling function) but is fairly slow. We have added an R function `fastAGM` for fast fitting of an additive genetic model (equivalent to but much faster than using `lm`).

```
> ut = unix.time(sCPNE1 <- snpScreen(chr20GGdem, chr20meta, genesym("CPNE1"),
+   ~., fastAGM, 75))
> ut

      user  system elapsed
      0.13    0.01    0.14

> sCPNE1
```

GGtools snpScreenResult for call:

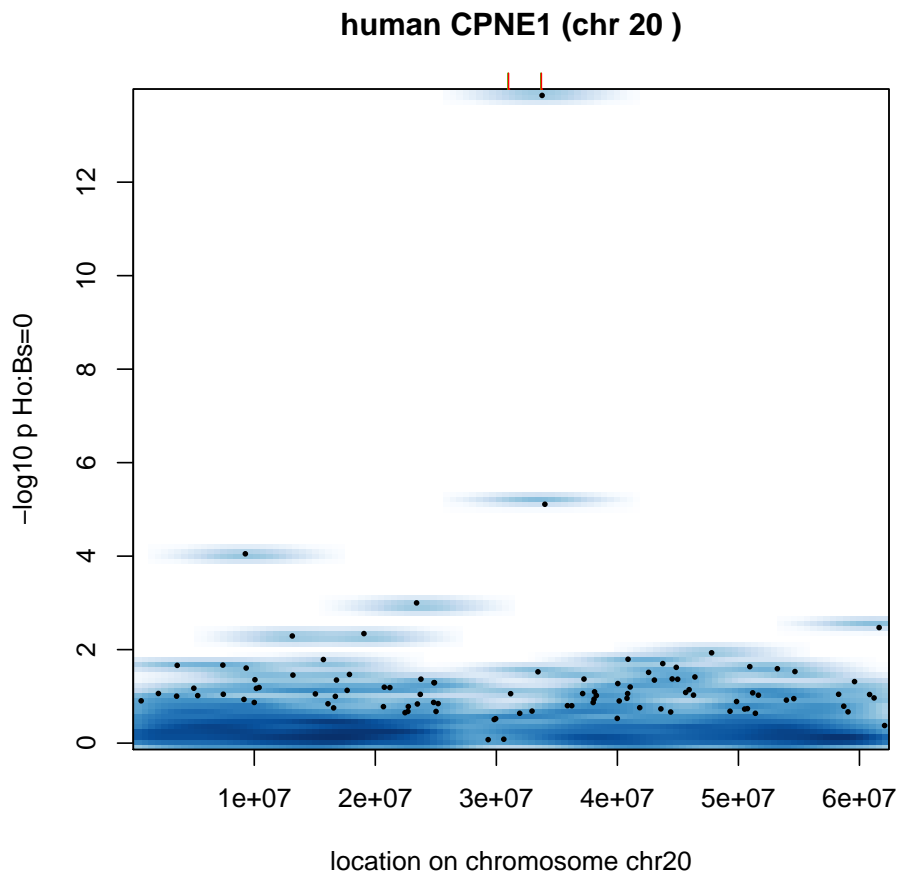
```
snpScreen(racExSet = chr20GGdem, snpMeta = chr20meta, gene = genesym("CPNE1"),  
  formTemplate = ~., fitter = fastAGM, gran = 75)
```

There were 581 attempted fits,
and 581 were successful.

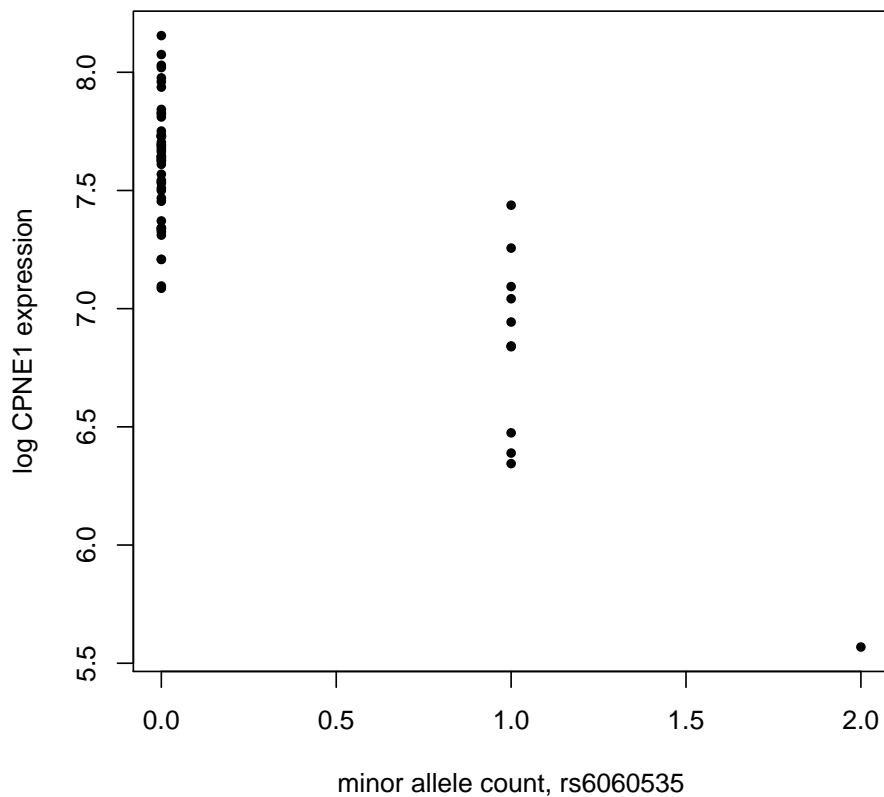
```
> wm = which.min(pp <- extract_p(sCPNE1))  
> pp[wm]
```

```
rs2425109  
1.395072e-14
```

```
> data(geneLocs_hsa)  
> plot_mlp(sCPNE1, chr20meta, geneLocDF = geneLocs_hsa)
```



```
> plot_EvG(chr20GGdem, genesym("CPNE1"), "rs6060535")
```



3 Creating new `racExSet` instances and supporting objects

Genotypes are available for a wide number of reference strains of mice through the Wellcome Trust; expression studies of such mice can be obtained using GEO. The data structure `gse2031GG` is an example:

```
> data(gse2031GG)
> gse2031GG
```

```
racExSet instance (SNP rare allele count + expression)
```

```
rare allele count assayData:
```

```
Storage mode: lockedEnvironment
```

```
featureNames: rs3683945, rs3707673, ..., rs4232414, mCV23022620 (13368 total)
```

```
Dimensions:
```

```
      racs
```

```
Features 13368
Samples    44
```

```
expression assayData
```

```
Storage mode: lockedEnvironment
```

```
featureNames: 100001_at, 100002_at, ..., AFFX-b-ActinMur/M12481_M_at, AFFX-b-ActinMur
```

```
Dimensions:
```

```
exprs
```

```
Features 12488
```

```
Samples    44
```

```
phenoData
```

```
rowNames: GSM36673, GSM36674, ..., GSM36716 (44 total)
```

```
varLabels and varMetadata:
```

```
strain: strain from GEO soft file for series
```

```
gsm: GEO id
```

```
Experiment data
```

```
Experimenter name:
```

```
Laboratory:
```

```
Contact information:
```

```
Title:
```

```
URL:
```

```
PMIDs:
```

```
No abstract available.
```

```
Annotation [1] "mgu74av2"
```

The expression data are obtained by applying RMA to a collection of CEL files readily obtainable from GEO. The strains of the mice giving rise to the samples are obtained from the WebQTL "contacts" link.

The genotype data for a large number of strains can be obtained from the INBREDS distribution at Wellcome Trust. The URL is www.well.ox.ac.uk/mouse/INBREDS. An example of the genotype data can be seen using the following code:

```
> ff = readLines(system.file("fileDemos/StrainInit.txt", package = "GGtools"))
> t(sapply(strsplit(ff[1:5], " "), function(x) x[c(1, 2, 3, 55,
+      56, 57, 101)]))
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	"SNP"	"CHR"	"POSITION"	"AKXD-6/TyJ"	"AKXD-7/TyJ"	"AKXD-8/TyJ"
[2,]	"rs3683945"	"1"	"3157748"	"A"	"A"	"A"
[3,]	"rs3707673"	"1"	"3366533"	"G"	"G"	"G"

```

[4,] "rs6269442" "1" "3451386" "G" "G" "G"
[5,] "rs6336442" "1" "3539826" "G" "G" "G"
      [,7]
[1,] "BXA-12/PgnJ"
[2,] "A"
[3,] "G"
[4,] "A"
[5,] "A"

```

There are various representations provided in various places. Tightening the path to accurate genotyping data for reference strains would be useful.

Given an expression matrix and a genotyping file as provided at the INBREDS site, the `INBREDSworkflow` function will help generate a `racExSet` instance:

```

> args(INBREDSworkflow)

function (inbfile, emat, estrains, pd, mi, anno, fixup = NULL,
          fixchr = function(x) gsub("_random", "", x))
NULL

```

The `estrains` argument is a vector of strings defining the reference strains from which columns of `emat` were obtained through microarray hybridization. This vector will typically be available from a `phenoData` variable on the `ExpressionSet` from which the expression matrix was obtained. For the `gse2031GG`, this is the variable `strain`:

```

> as.character(gse2031GG$strain[1:5])

[1] "BXD6" "BXD6" "BXD8" "BXD8" "BXD11"

```

Now it appears that this nomenclature for strains is not always adopted directly. If we `grep` for a variation of `BXD[nn]` in the INBREDS strains file, we find

```

> grep("BXD-", strsplit(ff[1], " ")[[1]], value = TRUE)[1:5]

[1] "BXD-1/TyJ" "BXD-11/TyJ" "BXD-12/TyJ" "BXD-13/TyJ" "BXD-14/TyJ"

```

and we see there are lexical variations introduced – we want `BXD6`, but the labels in the INBREDS file have hyphens. A `fixup` parameter allows inline reformatting of deviant strain identifiers as column names of the INBREDS file. Additionally, chromosome names are sometimes postpended with “-random”, and some steps may be needed to simplify the chromosome nomenclature. The user must create/find the R code to obtain the desired results, or manually massage the source data files so that the desired regularities are present.

Once genotype codes are available with a strain naming convention that matches that of the expression samples, INBREDSworkflow will compute rare allele counts, bind the genotyping, expression, and phenotype data together, and generate a `racExSet` instance.

In summary, `make_racExSet` is a specification of materials that must be made mutually compatible for creation of a `GGtools` resource with which investigations of genetics of gene expression can be conveniently made. `HMworkflow` and `INBREDSworkflow` are utility functions that support this task for files coming from HapMap and Wellcome repositories.

4 Appendix: Package documentation for *GGtools*

Information on package 'GGtools'

Description:

```
Package:      GGtools
Title:        software and data for genetical genomics (c) 2006 VJ
              Carey
Version:      1.4.0
Author:       Vince Carey <stvjc@channing.harvard.edu>
Description:  dealing with hapmap SNP reports, GWAS, etc.
Depends:      R (>= 2.2.0), methods, Biobase (>= 1.11.26), hgfocus,
              geneplotter(>= 1.11.8), mgu74av2
LazyData:     no
biocViews:    SNPsAndGeneticVariability, Genetics, Statistics
Maintainer:   Vince Carey <stvjc@channing.harvard.edu>
License:      Artistic (see COPYING)
Collate:      snpMeta.R AllClasses.R AllGenerics.R HapMapUtils.R
              exclMono.R countRare.R fastAGM.R genoString.R oneFit.R
              racExSet-methods.R snpScreenResult-methods.R snps3Pto.R
              updateObject.R Strains2rac.R wrapSNPmetaWh.R
              anno2chrband.R ogtes.R zzz.R
Built:        R 2.5.0; i386-pc-mingw32; 2007-05-05 03:10:30; windows
```

Index:

```
HM2rac          compute rare allele count from a hapmap file
HMworkflow      function to bind together HapMap genotyping
                 results and expression data
```

Strains2rac	convert a Wellcome 'Strains' genotyping file to rare allele count form
geneLocs	gene metadata from NCBI
genoStrings	create a character vector of genotype value strings
make_racExSet	create a racExSet from simpler constituents
oGtypeExSet-class	Class "oGtypeExSet" ~~~
plot_EvG	plot expression vs genotype
racExSet-class	Class "racExSet" for combining RareAlleleCount representations of SNPs, gene expression data, and other phenotype data
snpMeta-class	Class "snpMeta" -- HapMap (or Wellcome INBREDS) -based metadata structures for SNPs
snpScreen	compute model fits over a sequence of SNPs
snps	accessor for genotype data in a ggExprSet

Further information is available in the following vignettes in directory 'E:/biocbld/bbs-2.0-bioc/tmpdir/Rinst207699935/GGtools/doc':

GGoverview: GGtools overview (source)

Session information for this vignette build:

```
> sessionInfo()
```

```
R version 2.5.0 (2007-04-23)
i386-pc-mingw32
```

```
locale:
```

```
LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MONETARY=E
```

```
attached base packages:
```

```
[1] "tools"      "stats"      "graphics"   "grDevices"  "utils"      "datasets"
[7] "methods"    "base"
```

```
other attached packages:
```

```
GGtools      mgu74av2  geneplotter  lattice      annotate      hgfocus
"1.4.0"      "1.16.0"  "1.14.0"     "0.15-5"     "1.14.1"     "1.16.0"
Biobase
"1.14.0"
```