

# Overview of GGtools for expression genetics

VJ Carey stvjc at channing.harvard.edu

August 17, 2010

## 1 Introduction

The *GGtools* package contains infrastructure and demonstration data for joint analysis of transcriptome and genome through combination of DNA expression microarray and high-density SNP genotyping data. For Bioconductor 2.2 we adopted a representation of genotypes due to Clayton (in package *snpMatrix*) allowing reasonably convenient storage and manipulation of 4 megaSNP phase II HapMap genotypes on all the CEPH CEU samples. This contrasts with the previous version of *GGtools* which was limited to 550 kiloSNP and 58 CEU founders.

To give an immediate taste of the capabilities, we attach the package and load some test data.

```
> library(GGtools)
```

```
Loading package ff 2.1-2.1
```

```
- getOption("fftempdir")== "D:/biocbld/bbs-2.6-bioc/tmpdir/RtmpzMDVyb"
- getOption("ffextension")== "ff"
- getOption("ffdrop")== TRUE
- getOption("fffinonexit")== TRUE
- getOption("ffpagesize")== 65536
- getOption("ffcaching")== "mmnoflush" -- consider "ffeachflush" if your system stalls
- getOption("ffbatchbytes")== 42928701 -- consider a different value for tuning your sys
```

```
Attaching package ff
```

```
> data(hmceuB36.2021)
```

```
> hmceuB36.2021
```

```
snp.matrix-based genotype set:
number of samples: 90
number of chromosomes present: 2
annotation: illuminaHumanv1.db
Expression data dims: 47293 x 90
```

```
Phenodata: An object of class "AnnotatedDataFrame"
  sampleNames: NA06985, NA06991, ..., NA12892 (90 total)
  varLabels and varMetadata description:
    famid: hapmap family id
    persid: hapmap person id
    ....: ...
    male: logical TRUE if male
    (7 total)
```

Expression data are recoverable in a familiar way:

```
> exprs(hmceuB36.2021)[1:5, 1:5]
```

	NA06985	NA06991	NA06993	NA06994	NA07000
GI_10047089-S	5.983962	5.939529	5.912270	5.891347	5.906675
GI_10047091-S	6.544493	6.286516	6.244446	6.277397	6.330893
GI_10047093-S	9.905235	10.353804	10.380972	9.889223	10.155686
GI_10047099-S	7.993935	7.593970	8.261215	6.598430	6.728085
GI_10047103-S	11.882265	12.204753	12.249708	11.798415	12.015252

Genotype data have more complex representation.

```
> smList(hmceuB36.2021)
```

```
$`20`
```

```
A snp.matrix with 90 rows and 119921 columns
```

```
Row names: NA06985 ... NA12892
```

```
Col names: rs4814683 ... rs6090120
```

```
$`21`
```

```
A snp.matrix with 90 rows and 50165 columns
```

```
Row names: NA06985 ... NA12892
```

```
Col names: rs885550 ... rs10483083
```

```
> class(smList(hmceuB36.2021)[["20"]])
```

```
[1] "snp.matrix"
```

This shows that we use a named list to hold items of the *snp.matrix* class from *snpMatrix*.

It will generally be unnecessary to probe to this level, but it is instructive to check the underlying representation:

```
> schunk = smList(hmceuB36.2021)[["20"]]
```

```
> schunk@.Data[1:4, 1:4]
```

	rs4814683	rs6076506	rs6139074	rs1418258
NA06985	03	03	03	03
NA06991	02	03	02	02
NA06993	01	03	01	01
NA06994	01	03	01	01

The leading zeroes show that a raw byte representation is used. We can convert to allele codes as follows:

```
> as(schunk[1:4, 1:4], "character")
```

	rs4814683	rs6076506	rs6139074	rs1418258
NA06985	"B/B"	"B/B"	"B/B"	"B/B"
NA06991	"A/B"	"B/B"	"A/B"	"A/B"
NA06993	"A/A"	"B/B"	"A/A"	"A/A"
NA06994	"A/A"	"B/B"	"A/A"	"A/A"

The primary method of interest is the genome-wide association study, here applied with expression as the phenotype. Here we execute a founders-only analysis, adjusting for gender, confining attention to chromosome 20:

```
> pd = pData(hmceuB36.2021)
> hmFou = hmceuB36.2021[, which(pd$mothid == 0 & pd$fathid == 0)]
> f1 = gwSnpTests(genesym("CPNE1") ~ male, hmFou, chrnum(20))
```

## 2 Conversion to nucleotide codes

This is currently somewhat cumbersome. Suppose we want to know the specific nucleotide assignments for a given genotype call. For example, rs4814683 for subject NA06985.

```
> schunk["NA06985", "rs4814683"]
```

```
Autosomal snp(s):
[1] "B/B"
```

We need to know a) that the A/B tokens map in lexical order to the nucleotides (A will be the alphabetically first nucleotide for the diallelic call).

Using the `SNPlocs.Hsapiens.dbSNP.*` package, we can get the nucleotides:

```
> library(SNPlocs.Hsapiens.dbSNP.20080617)
> s20 = getSNPlocs("chr20")
> s20[s20[, 1] == 4814683, ]
```

Now we need to translate the IUPAC code to the nucleotides:

```
> library(Biostrings)
> IUPAC_CODE_MAP
```

A	C	G	T	M	R	W	S	Y	K	V
"A"	"C"	"G"	"T"	"AC"	"AG"	"AT"	"CG"	"CT"	"GT"	"ACG"
H	D	B	N							
"ACT"	"AGT"	"CGT"	"ACGT"							