Key practical concerns of the tutorial on genetics of gene expression

Thematic overview (1)

- Data representations, strategies
  - Upstream: expression, genotype, sample-level
    - `SnpMatrix`, `smlSet`
  - Downstream: `eqtlTestsManager` class
    - Managing (m|b)illions of test results with, e.g., ff
    - Coordinating test results with location metadata
    - Creating tracks, RDBMS when needed
- Focused tests, visualizations: `gwSnpTests`
  - You know the gene of interest and seek variants associated with expression

Thematic overview (2)

- Mass testing for cis associations, using location information
  - PRINCIPLE 1: Pre-compute and filter later: computing *all* same-chromosome tests is simple
  - PRINCIPLE 2: Divide and conquer: abandon petty holisms
- `eqtlTests` can run quickly by setting `geneApply` and `chromApply` to work concurrently on distinct structures
- `cisProxScores` can filter `eqtlTestsManager` instances using gene and SNP location metadata couched in IRanges structures

Thematic overview (3)

- Interpreting scores:
  - Using permutations parsimoniously to get approximate FDR
  - Structural coincidence with genomic features
- Enhancing power of eQTL search with post-hoc variance reduction: PCA, SVA
  - Caveat – PCA exquisitely tuned to the given data
- Enhancing resolution using imputation on the basis of population genetic models

Thematic overview (4)

- Conducting and representing trans searches
  - 20000 genes x 10 million SNP: 200bb tests, most of which are null
  - `scratch pad’ approach with multiple ff archives and several types of indirection
- Short reads for RNA-seq-based assessment of allelically imbalanced expression

Packages of interest

- *GGtools, GGBase*: class/method definitions; import and transformation tools
- *snpStats*: key representations of observed and imputed genotypes; import tools; fast execution of snp-specific tests; efficient execution and representation of SNP imputation
- *GenomicFeatures*, *VariantAnnotation*: structural metadata
- *SVA*: (distributed by Leek) reducing expression heterogeneity

Getting acquainted with CEPH CEU samples via *GGtools*

- Provenance:
  - Expression: GENEVAR project distributed expression data on 90 LCL (lymphoblastoid cell lines) collected on the illumina WG6 v1 platform
  - Genotype: as distributed by HapMap phase II: approximately 4 million loci obtained via Sanger sequencing
  - Familial structure: 30 trios with some extended relationships

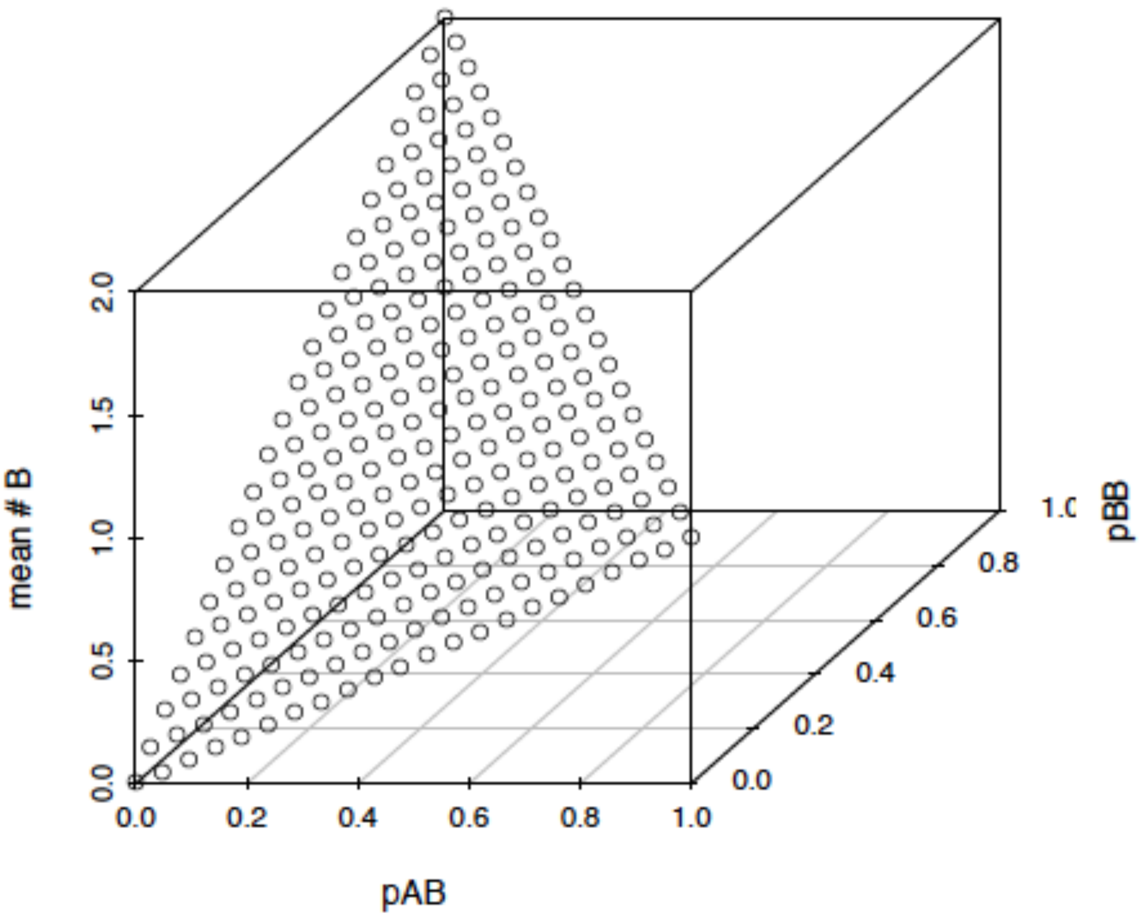
`getSS` draws a selection of chromosomes of SNP data from a package

```
> suppressPackageStartupMessages(library(GGtools))
> c17 = getSS("GGdata", "17", renameChrs="chr17")
Loading required package: GGdata
Loading required package: illuminaHumanv1.db

To get a tailored smlSet, use getSS("GGdata", [chrvec])
available chromosomes are named  1 10 ... X Y
> c17
SnpMatrix-based genotype set:
number of samples:  90
number of chromosomes present:  1
annotation: illuminaHumanv1.db
Expression data dims: 47293 x 90
Phenodata: An object of class "AnnotatedDataFrame"
  sampleNames: NA06985 NA06991 ... NA12892 (90 total)
  varLabels: famid persid ... male (7 total)
  varMetadata: labelDescription
```

Expression and genotype components; count number of risk alleles (+1 ; 0=missing)

```
> dim(exprs(c17))
[1] 47293    90
> dim(smList(c17)[["chr17"]])
[1]    90 89701
> all(sampleNames(c17) == rownames(smList(c17)
[["chr17"]]))
[1] TRUE
> c17g = smList(c17)[["chr17"]]
> as(c17g[1:5,1:5], "matrix")  # raw representation
        rs6565733 rs1106175 rs17054921 rs8064924 rs8070440
NA06985        03        02         03        03        03
NA06991        03        01         03        03        03
NA06993        03        01         03        03        03
NA06994        02        02         03        02        03
NA07000        03        01         03        03        03
```

snpStats byte mapping (253 points) accommodating uncertain genotypes



featureNames and phenoData manipulations

```
> featureNames(c17)[1:3]
[1] "GI_10047089-S" "GI_10047091-S"
"GI_10047093-S"
> phenoData(c17)
An object of class "AnnotatedDataFrame"
  sampleNames: NA06985 NA06991 ... NA12892
(90 total)
  varLabels: famid persid ... male (7 total)
  varMetadata: labelDescription
> table(table(c17$famid))

 3  6 
10 10 
> table(c17$male)

FALSE  TRUE 
   46    44 
```

X[G,S] idiom applies as expected; no simple approach to subsetting loci

```
> dim(c17[1:10,])
Features  Samples
      10       90
> dim(c17[1:10,1:10])
Features  Samples
      10       10
> dim(smList(c17[1:10,1:10])[["chr17"]])
[1]    10 89701
> dim(permEx(c17))
Features  Samples
   47293       90
```

Summary of smlSet strategy

- `ExpressionSet` has substantial infrastructure
  - `X[G,S]` idiom to simplify filtering
  - `X$p` to acquire phenoData variables
- `SnpMatrix` class has many attractions
  - Compact representation of sufficient information for eQTL discovery
  - High-performance implementation of various testing algorithms of interest
- *GGBase*: Bind these together using S4

To create an smlSet

- Acquire the expression data as appropriate
- Transform genotyping results to `SnpMatrix` instance – typically one `SnpMatrix` per chromosome
- *GGBase* `make_smlSet` will take care of details
- Worked OK for up to 4 million loci (early deployment of *GGdata*)
- With more loci need to support divide-and-conquer approach

Reducing expression plus genotype footprint

- Given an `smlSet` instance with C chromosomes represented in the `smList` component, GGtools `externalize` function will create a package
- `getSS` operates on the package name and a chromosome selection to generate a scaled-down `smlSet`
- Get rid of the original holistic object – you can load and unload genotype data at the chromosome level

Some basic genetics computations

```
> col.summary(smList(c17)[["chr17"]])[1:3,]
           Calls Call.rate Certain.calls   RAF    MAF
rs6565733     90         1             1 0.872 0.1278
rs1106175     90         1             1 0.183 0.1833
rs17054921    90         1             1 0.989 0.0111
              P.AA   P.AB   P.BB  z.HWE
rs6565733  0.0111 0.2333 0.7556  0.444
rs1106175  0.6778 0.2778 0.0444 -0.686
rs17054921 0.0000 0.0222 0.9778  0.107
> row.summary(smList(c17)[["chr17"]])[1:3,]
        Call.rate Certain.calls Heterozygosity
NA06985     0.979             1          0.214
NA06991     0.982             1          0.208
NA06993     0.987             1          0.223
```

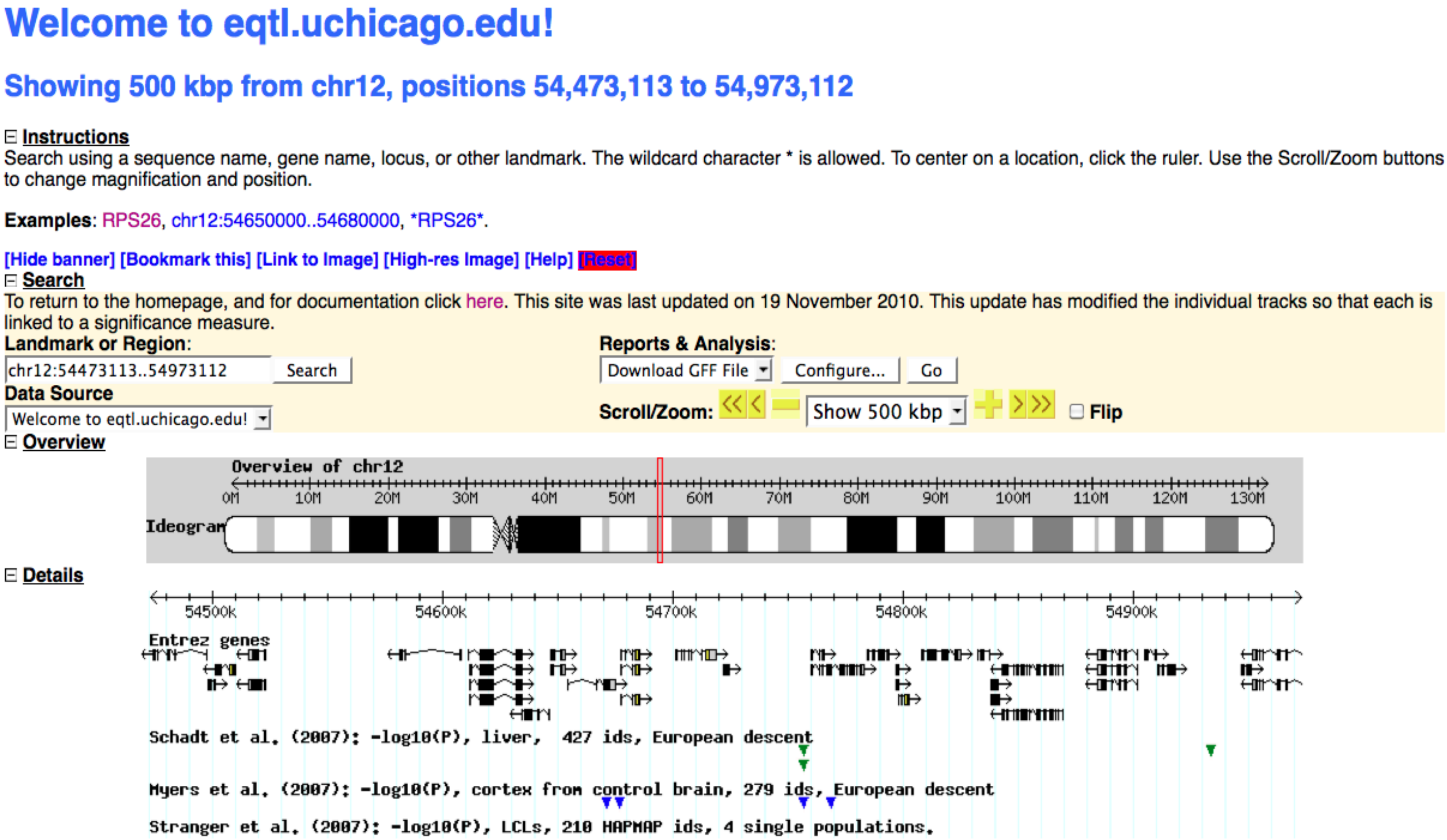
Various LD stats

```
> ld(s17of, depth=20, stats="D.prime")[1:5,]
5 x 59999 sparse Matrix of class "dgCMatrix"
   [[ suppressing 34 column names 'rs4424950', 'rs7503116', 'rs1136388' ... ]]

rs4424950 . 1 1 1 1 1 0.9463 1 1 1 1 0.01100 1 1 0.6678 0.5440 0.3895 0.47612
rs7503116 . . 1 1 1 1 0.9483 1 1 1 1 0.01711 1 1 0.6908 0.5417 0.3785 0.64985
rs1136388 . . . 1 1 1 1.0000 1 1 1 1 0.03866 1 1 0.6947 0.6165 0.4037 0.65025
rs1609550 . . . . 1 1 1.0000 1 1 1 1 0.87029 1 1 0.6044 0.2611 0.1457 0.05194
rs7212865 . . . . . 1 1.0000 1 1 1 1 1.00000 1 1 1.0000 1.0000 1.0000 1.00000

rs4424950 0.5047 0.5034 0.3333 .      .       .      . . . . . . . . . . ......
rs7503116 0.5256 0.5245 0.3371 0.5722 .       .      . . . . . . . . . . ......
rs1136388 0.4950 0.5245 0.3034 0.6290 0.03608 .      . . . . . . . . . . ......
rs1609550 0.1766 0.4719 0.6667 0.5620 0.61449 0.4136 . . . . . . . . . . ......
rs7212865 1.0000 1.0000 1.0000 1.0000 1.00000 1.0000 1 . . . . . . . . . ......
```

Functions in snpStats

```
> objects("package:snpStats")
 [1] "can.impute"            "chi.squared"           "col.summary"
 [4] "convert.snpMatrix"     "convert.snpMatrix.dir" "deg.freedom"
 [7] "effect.sign"           "effective.sample.size" "filter.rules"
[10] "Fst"                   "glm.test.control"      "ibsCount"
[13] "ibsDist"               "imputation.maf"        "imputation.nsnp"
[16] "imputation.r2"         "impute.snps"           "ld"
[19] "misinherits"           "mvtests"               "p.value"
[22] "plot"                  "plotUncertainty"       "pool"
[25] "pool2"                 "qq.chisq"              "read.beagle"
[28] "read.impute"           "read.long"             "read.mach"
[31] "read.pedfile"          "read.plink"            "read.snps.long"
[34] "row.summary"           "sample.size"           "single.snp.tests"
[37] "sm.compare"            "snp.cbind"             "snp.cor"
[40] "snp.imputation"        "snp.lhs.estimates"     "snp.lhs.tests"
[43] "snp.post.multiply"     "snp.pre.multiply"      "snp.rbind"
[46] "snp.rhs.estimates"     "snp.rhs.tests"         "summary"
[49] "switch.alleles"        "tdt.snp"               "test.allele.switch"
[52] "write.plink"           "write.SnpMatrix"       "xxt"
```

Some basic objectives



Welcome to eqtl.uchicago.edu!

Showing 500 kbp from chr12, positions 54,473,113 to 54,973,112

⊟ **Instructions**

Search using a sequence name, gene name, locus, or other landmark. The wildcard character * is allowed. To center on a location, click the ruler. Use the Scroll/Zoom buttons to change magnification and position.

Examples: RPS26, chr12:54650000..54680000, *RPS26*.

[Hide banner] [Bookmark this] [Link to Image] [High-res Image] [Help] [Reset]

⊟ **Search**

To return to the homepage, and for documentation click here. This site was last updated on 19 November 2010. This update has modified the individual tracks so that each is linked to a significance measure.

Landmark or Region: chr12:54473113..54973112 Search

Data Source: Welcome to eqtl.uchicago.edu!

Reports & Analysis: Download GFF File Configure... Go

Scroll/Zoom: Show 500 kbp □ **Flip**

⊟ **Overview**

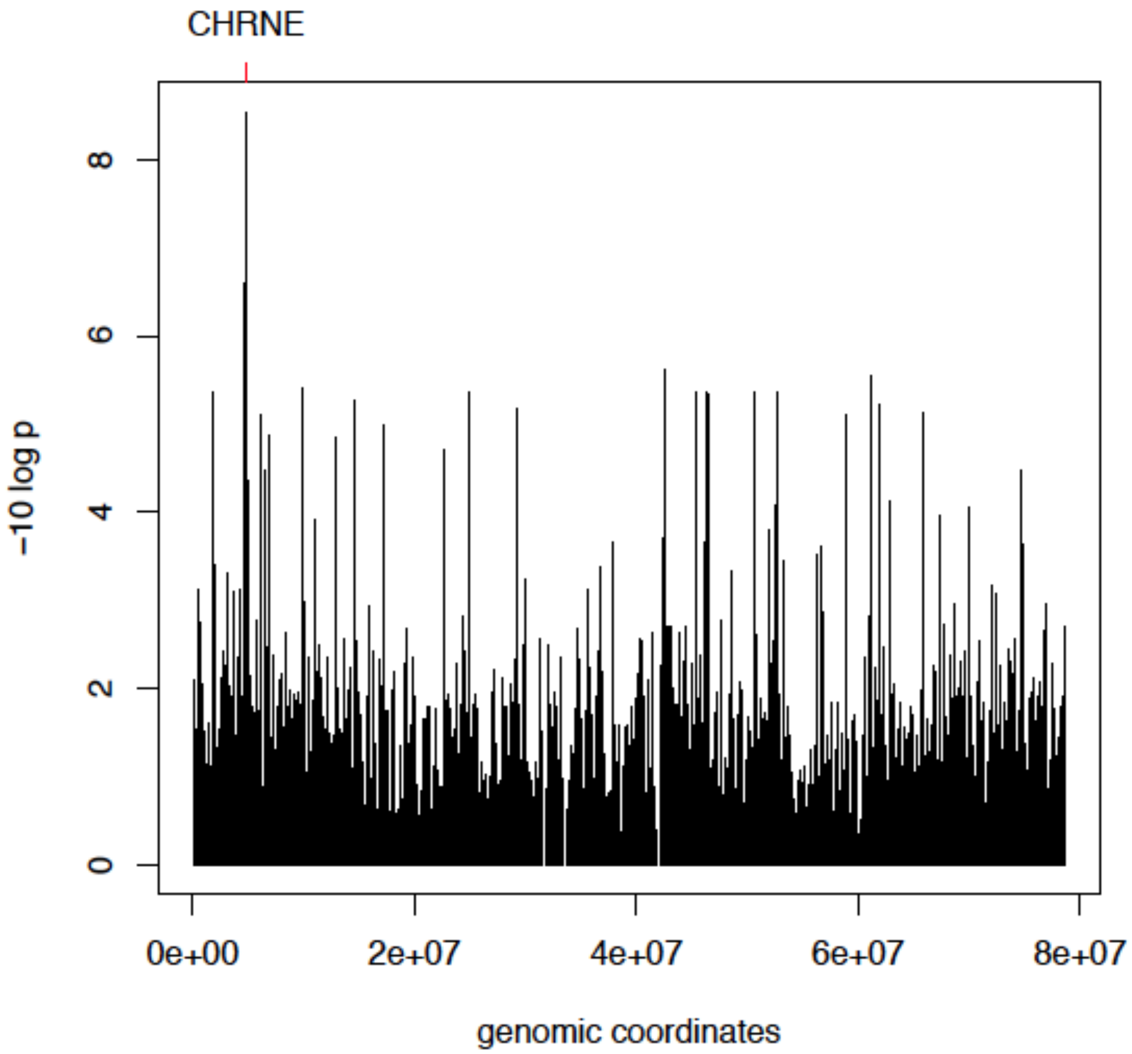
⊟ **Details**

Overview of chr12
0M
10M
20M
30M
40M
50M
60M
70M
80M
90M
100M
110M
120M
130M
Ideogram
54500k
54600k
54700k
54800k
54900k
Entrez genes
Schadt et al. (2007): -log10(P), liver, 427 ids, European descent
Myers et al. (2007): -log10(P), cortex from control brain, 279 ids, European descent
Stranger et al. (2007): -log10(P), LCLs, 210 HAPMAP ids, 4 single populations.

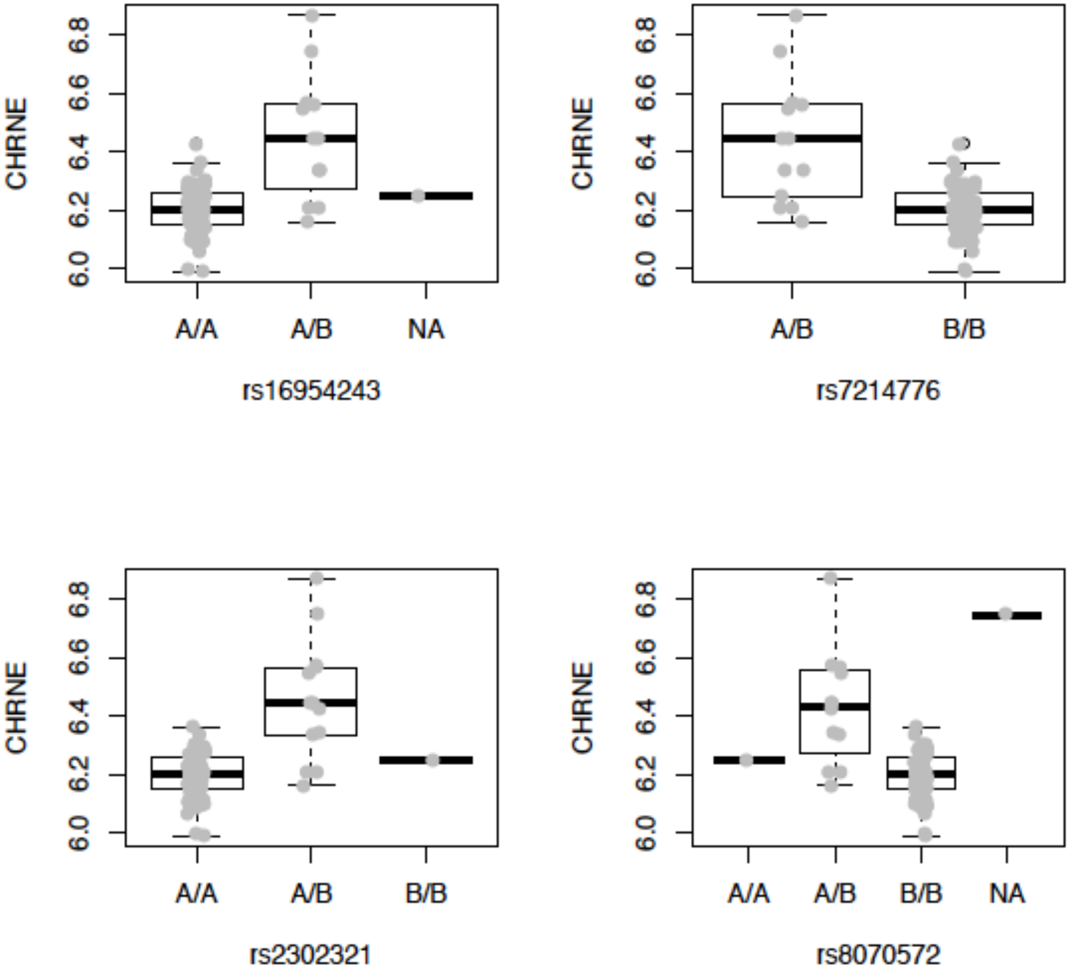
eqtl browser

- Where have eQTL been identified?
- By what studies?
- In what populations?
- In what cell types?
- What is the nature of the association?
- What is the functional context?
- *We want to do this with our own data*

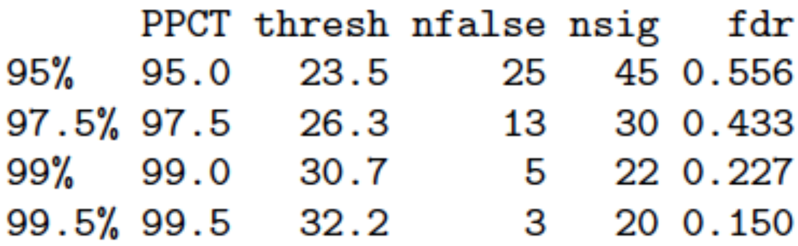
A view of the search



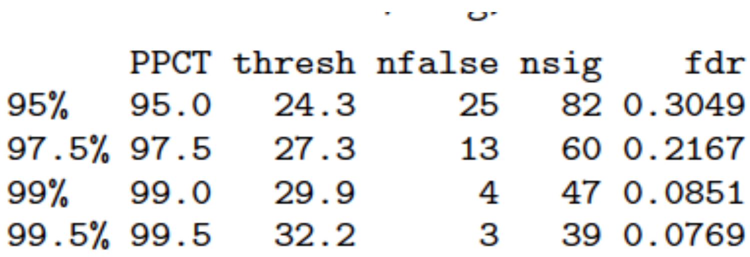
How many loci? Are the tests efficient?



Flexible and powerful inference methods? Each based on 30 million tests. Left: basic preprocessing; Right: PCA-based expression heterogeneity adjustment. (Code to create these tables is in extras.Rnw)



```
      PPCT thresh nfalse nsig   fdr
95%   95.0   23.5     25   45 0.556
97.5% 97.5   26.3     13   30 0.433
99%   99.0   30.7      5   22 0.227
99.5% 99.5   32.2      3   20 0.150
```



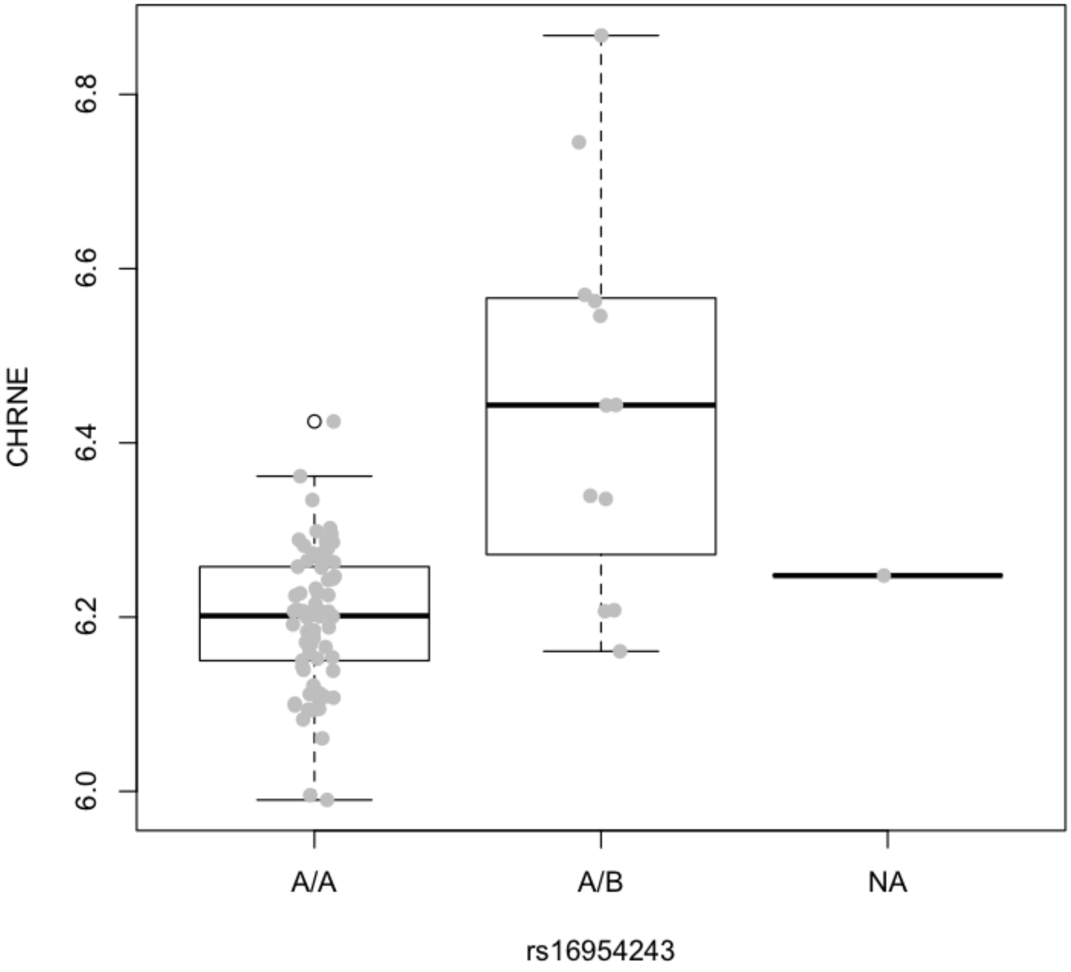
```
      PPCT thresh nfalse nsig    fdr
95%   95.0   24.3     25   82 0.3049
97.5% 97.5   27.3     13   60 0.2167
99%   99.0   29.9      4   47 0.0851
99.5% 99.5   32.2      3   39 0.0769
```

A focused test sequence that we can do together, motivated by Stranger et al.

```
> t1 = gwSnpTests(genesym("CHRNE")~male, c17, chrnum("chr17"))
> t1
gwSnpScreenResult for gene  CHRNE  [probe  GI_38327653-S ]
> topSnps(t1)
              p.val
rs16954243 2.93e-09
rs7214776  7.56e-09
rs8081611  7.56e-09
rs2302321  4.84e-08
rs8070572  2.51e-07
rs7225684  4.09e-07
```

```
> class(t1)
[1] "cwSnpScreenResult"
attr(,"package")
[1] "GGBase"
> class(t1@.Data)
[1] "list"
> class(t1@.Data[[1]])
[1] "GlmTests"
attr(,"package")
[1] "snpStats"
> getClass("GlmTests")
Class "GlmTests" [package "snpStats"]

Slots:

Name:  snp.names var.names     chisq
df         N
Class:       ANY character   numeric
integer   integer

Known Subclasses: "GlmTestsScore”
> length(p.value(t1@.Data[[1]]))
[1] 89701
```

```
plot_EvG(genesym("CHRNE"), rsid("rs16954243"), c17)
```



Summary on focused eQTL testing

- Given an `smlSet` instance, `gwSnpTests` can perform cis (or trans) tests for a given expression probe
- Covariates and strata/cluster identifiers are retrieved from `phenoData` to bind symbols in the formula
- It will use `snp.rhs.tests` from *snpStats* and will wrap the result in an S4 container
- The primary product is a collection of Chi-squared statistics and associated p-values

Exercise: check consistency of findings between two populations

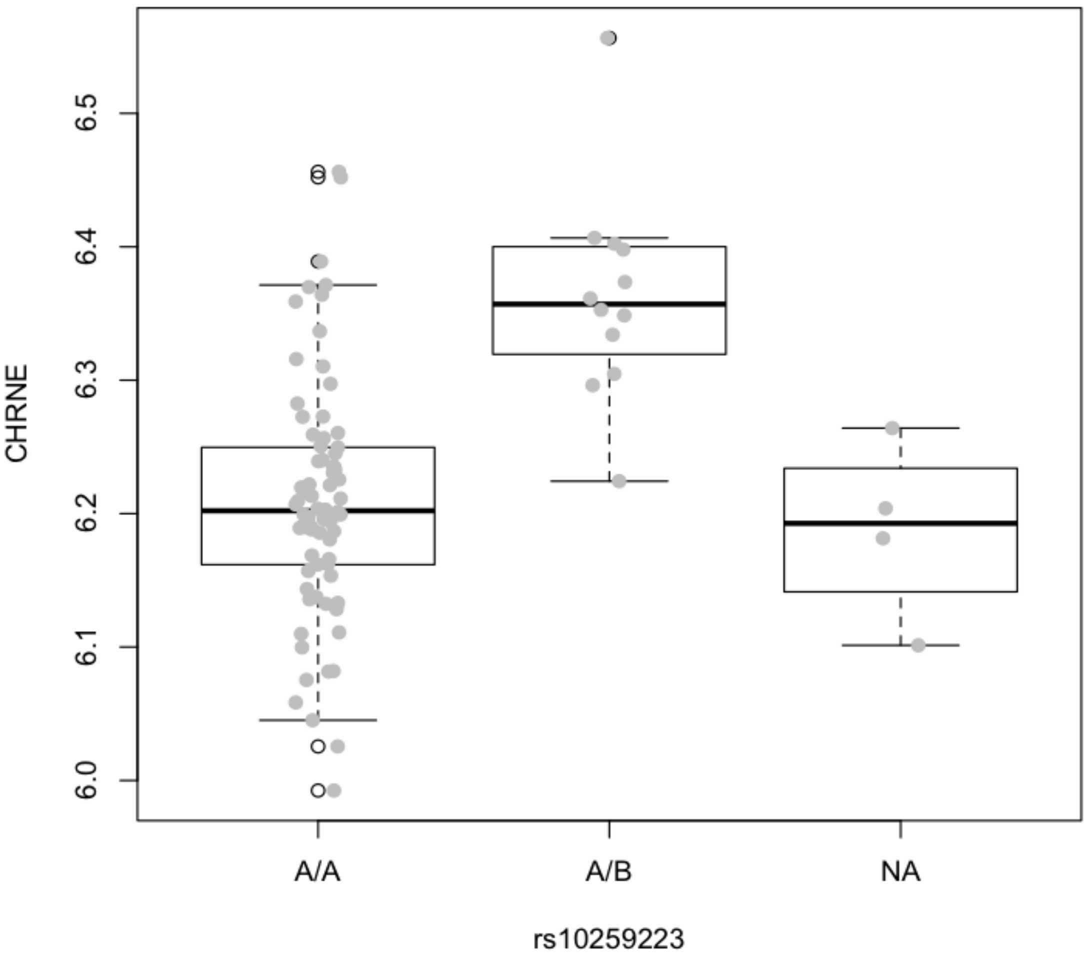
- GGdata delivers smlSets based on CEU
- hmyriB36 delivers smlSets based on YRI
- How do we check our CEU-based result on CHRNE in the YRI cohort?

```
> y17 = getSS("hmyriB36", "17", renameChrs="chr17")
Loading required package: hmyriB36
To get a tailored smlSet, use getSS("hmyriB36", [chrvec])
available chromosomes are named  1 10 ... X Y
> y1 = gwSnpTests(sym = genesym("CHRNE") ~ male, sms = y17,
chrnum("chr17"))
> topSnps(y1)
              p.val
rs9889685  0.000124
rs7212518  0.000135
rs9907560  0.000141
rs6501801  0.000141
rs2024498  0.000143
rs8069187  0.000143
```

A full genome-wide search in YRI (save your work before trying this)

```
> yfull = getSS("hmyriB36", as.character(1:22))
> y2 = gwSnpTests(genesym("CHRNE")~male, sms=yfull)
> sum(sapply(y2@.Data, function(x)length(p.value(x))))
[1] 3762311
> unlist(topSnps(y2,n=1))
 1.p.val  2.p.val  3.p.val  4.p.val  5.p.val  6.p.val  7.p.val  8.p.val 
1.24e-05 1.42e-05 2.53e-05 2.58e-05 8.19e-05 3.15e-05 1.83e-06 1.68e-05 
 9.p.val 10.p.val 11.p.val 12.p.val 13.p.val 14.p.val 15.p.val 16.p.val 
2.52e-05 2.83e-05 1.15e-04 4.84e-06 4.80e-05 2.21e-05 2.34e-05 5.37e-05 
17.p.val 18.p.val 19.p.val 20.p.val 21.p.val 22.p.val 
1.24e-04 7.49e-05 3.36e-04 6.08e-05 3.01e-04 3.29e-05 
> which.min(.Last.value)
7.p.val 
      7 
> topSnps(y2,n=1)[[7]]
              p.val
rs10259223 1.83e-06
```

The best hit



Using permutation to informally assess significance: `permEx` permutes expression against genotype

```
> set.seed(1234)
> y2p = gwSnpTests(genesym("CHRNE")~male, sms=permEx(yfull))
> sort(unlist(topSnps(y2p,n=1)))
20.p.val  7.p.val  5.p.val 14.p.val  6.p.val  9.p.val 11.p.val 22.p.val
4.27e-06 1.02e-05 1.28e-05 1.53e-05 2.26e-05 2.41e-05 2.48e-05 4.21e-05
13.p.val 12.p.val  3.p.val  1.p.val  8.p.val 21.p.val 10.p.val 15.p.val
4.27e-05 4.41e-05 4.60e-05 4.85e-05 5.49e-05 5.70e-05 8.64e-05 8.95e-05
 4.p.val  2.p.val 16.p.val 18.p.val 17.p.val 19.p.val
9.46e-05 1.07e-04 1.11e-04 1.32e-04 1.76e-04 1.83e-04
> y2p = gwSnpTests(genesym("CHRNE")~male, sms=permEx(yfull))
> sort(unlist(topSnps(y2p,n=1)))
16.p.val 17.p.val  5.p.val  3.p.val 11.p.val  1.p.val 20.p.val  2.p.val
1.85e-06 2.72e-06 3.59e-06 4.98e-06 5.59e-06 6.43e-06 8.51e-06 9.70e-06
 6.p.val  7.p.val 14.p.val 13.p.val 15.p.val  9.p.val  8.p.val 22.p.val
1.37e-05 1.42e-05 1.96e-05 2.03e-05 3.12e-05 3.39e-05 4.35e-05 4.79e-05
12.p.val 21.p.val  4.p.val 10.p.val 19.p.val 18.p.val
5.37e-05 5.52e-05 6.58e-05 7.74e-05 1.13e-04 1.82e-04
```

Summary to this point

- *snpStats* package provides infrastructure for representing and testing with large numbers of genotypes (laptop can easily handle 4 million SNP x 90 samples)
- *GGtools* adds tools facilitating exploration of eQTL landscapes for specified genes
  - `smlSet` class design
  - `gwSnpTests, topSnps, plot_EvG`
  - `permEx`
- Next step: transcriptome x SNPome search

Comprehensive survey of cis-associated eQTL

- Expand what has been shown to 20000-40000 expression probes
- Accommodate concerns that real effects on gene G are more likely within 100kb of coding region of G
- Counter-concern: many location assertions are uncertain, but the data themselves are less so
- Approach adopted: measure associations between DNA and mRNA abundance and deal with location-related filtering and interpretation later

After filtering genes (mainly for tractability) we manage 30million tests on one chromosome

```
> suppressPackageStartupMessages(library(ggtut))
> o17 = observed17ceu()
> o17
eqtlTools results manager, computed Fri May  6 16:05:50 2011
gene annotation: illuminaHumanv1.db
There are 1 chromosomes analyzed.
some genes (out of 498): GI_10190685-S GI_10835020-S ...
hmm23927-S hmm5188-S
some snps (out of 60967): rs6565733 rs1106175 ... rs7502145
rs4986109
> 498*60967
[1] 30361566
> o17@call
eqtlTests(smlSet = c17, rhs = ~male, targdir = "c17c", geneApply
= mclapply,
    genegran = 1)
```

Using the 30 million tests to reason about expression regulation

```
> p17 = probesManaged(o17,1)
> unix.time(topOn17 <- lapply(p17, function(x)topFeats
(probeId(x),mgr=o17,ffind=1)))
   user  system elapsed
  13.81    5.21   21.79
Warning message:
In `[.ff_array`(fflist(mgrOrCTD)[[ffind]], , probeid) :
  opening ff /Users/stvjc/ISMB11_MACLIB/ggtut/ffarchives/
c17c/foo_chrchr17.ff
> maxOn17 = sapply(topOn17, "[", 1)
> maxOn17[1:5]
 rs4794214  rs9916609  rs2685524 rs11869731  rs2584597
      18.0       17.6       14.0       15.2       21.4
>
```

These are the maximum per-gene scores obtained in 498x80K tests

Using a permutation to reason about significance of the maximal scores

```
> perm17 = onePerm17ceu()
> unix.time(topOn17_p <- lapply(p17, function(x)topFeats(probeId
(x),mgr=perm17,ffind=1)))
   user  system elapsed 
  13.78    5.21   21.20 
Warning message:
In `[.ff_array`(fflist(mgrOrCTD)[[ffind]], , probeid) :
  opening ff /Users/stvjc/ISMB11_MACLIB/ggtut/ffarchives/c17c_perm/
foo_chrchr17.ff
> maxOn17_p = sapply(topOn17_p, "[", 1)
> targs = c(.95, .975, .995)
> pthresh = quantile(maxOn17_p, targs)
> nfalse = sapply(pthresh, function(x)sum(maxOn17_p>x))
> nsig = sapply(pthresh, function(x)sum(maxOn17>x))
> fdr = nfalse/nsig
> cbind(pthresh, nfalse, nsig, fdr)
      pthresh nfalse nsig   fdr
95%      23.5     25   45 0.556
97.5%    26.3     13   30 0.433
99.5%    32.2      3   20 0.150
```

Three approaches to improving power

- Filter the SNP: ignore loci with very low MAF (depends on sample size)
- Alter the question: focus the tests to physical intervals about the coding region, instead of surveying all same-chromosome tests
- Reduce effects of measurement variation: adjust for components of `expression heterogeneity’
- Of course these can be combined, but each has a tuning element.

A helper function

```
fdrtab =
function(obs, perm, pct=c(.95, .975, .99)){
   thresh = quantile(perm,pct)
   nfalse = sapply(thresh, function(x)sum(perm>x))
   nsig = sapply(thresh, function(x)sum(obs>x))
   fdr=nfalse/nsig
   cbind(pct,thresh,nfalse,nsig,fdr)
}
```

Confining to MAF > .1 leads to 36 calls at FDR=.14, vs 20 calls at FDR=.15

```
> unix.time(topobsAt.1 <- lapply(p17,
   function(x)topFeats(probeId(x), mgr=o17, ffind=1, n=150,
minMAF=.1)))
   user  system elapsed
   23.8    12.8    36.7
> toppermAt.1 = lapply(p17, function(x)topFeats(probeId
(x),mgr=perm17,ffind=1,n=150,minMAF=.1))
> s1 = function(x)sapply(x,"[",1)
> fdrtab(s1(topobsAt.1), s1(toppermAt.1))
        pct thresh nfalse nsig   fdr
95%   0.950   19.2     25   78 0.321
97.5% 0.975   20.5     13   53 0.245
99%   0.990   22.0      5   36 0.139
```

Using proximity information to filter tests

- Two elements are involved: location metadata, and the `cisProxScores` method that filters the `eqtlTestsManager` data
- We use GRanges instances for metadata
- The user can supply desired gene and SNP locations, and the radii of intervals of interest

Pre-computed location-delimited score sets are provided, self-describing; the scoresByGenes method will harvest

```
> data(CPS17)
> data(PERMCPS17)
> CPS17
GGtools cisProxScores instance.
The call was: cisProxScores(dradset = c(50000, 2e+06), direc = df1,
        snpGRL = list(obs17 = snpgr17),
        geneGRL = list(obs17 = g17rngsnr), ffind = 1)
intervals examined: FL0e+00.5e+04 FL5e+04.2e+06
> args(scoresByGenes)
function (cps, intvind = 1, as.GRanges = TRUE, dups2max = TRUE,
    snpGR = NULL, scoreConverter = function(x) x)
NULL
```

When limiting to SNP within 50kb, we have 70 calls of genes with eQTL at FDR 7%

```
> topIn50k = sapply(scoresByGenes(CPS17,1,FALSE),max)
> topIn50k_perm = sapply(scoresByGenes
(PERMCPS17,1,FALSE),max)
> fdrtab(topIn50k,topIn50k_perm)
        pct thresh nfalse nsig    fdr
95%   0.950   9.95     24  113 0.2124
97.5% 0.975  11.28     12   88 0.1364
99%   0.990  12.89      5   70 0.0714
```

The third way

- The third way involves computing PCs (or allied quantities) from the expression data and using these as adjustments in the formula to eqtlTests
- This is fully worked out in extras.pdf, so we won’t belabor it here.

Checking for structural coincidences

- Given a list of SNP and their locations, we can assess how frequently they occur within regions occupied by other genomic features using IRanges facilities
- We saw above that we can assert that there are 70 genes with eQTL within 50kb of coding region with FDR 7%. What are the associated SNP?

Find the best scoring loci's names, and check

```
> sb1 = scoresByGenes(CPS17, as.GRanges=FALSE)
> sb1max = sapply(sb1, max)
> sum(sb1max>12.89)
[1] 70
> isSig = which(sb1max>12.89)
> bestAt.7 = sapply(sb1[isSig], function(x)names(sort
(x,decreasing=TRUE))[1])
> bestAt.7[1:5]
GI_10190685-S GI_10835020-S GI_11038675-A GI_11345491-S
GI_11496988-S
   "rs489698"   "rs1666263"  "rs12946669"  "rs17761864"
"rs12450199"   # remember X[G,S] ?
> o17[rsid("rs489698"), probeId("GI_10190685-S")]
$chr17
         GI_10190685-S
rs489698           15.5
```

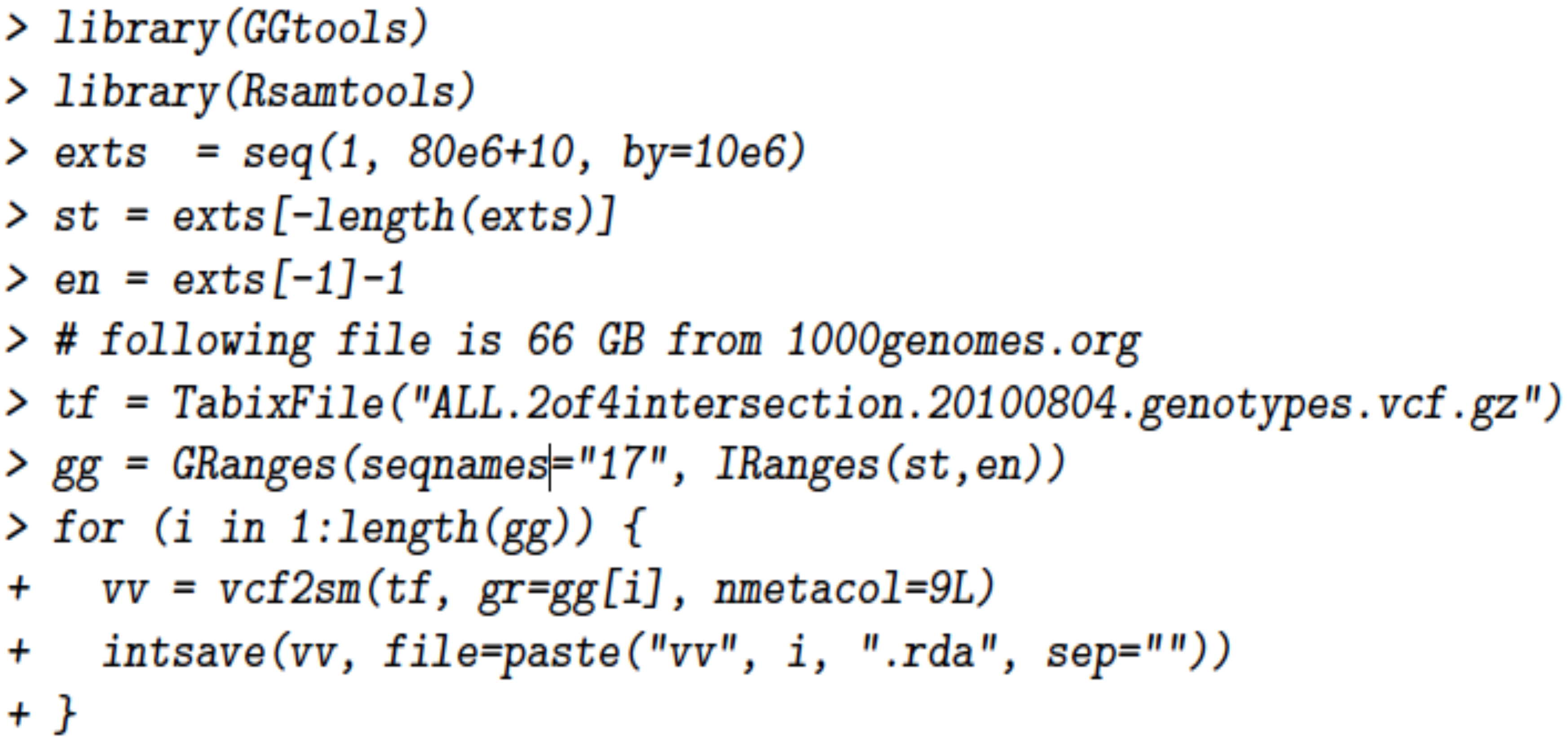
Question: what proportion of the important loci lie in exons?

```
> txdb = hg18tx()
> exloc = exons(txdb, vals=list(exon_chrom="chr17"))
> exloc[1:2]
GRanges with 2 ranges and 1 elementMetadata value
    seqnames           ranges strand |   exon_id
       <Rle>        <IRanges>  <Rle> | <integer>
[1]    chr17 [181049, 182929]      + |    212607
[2]    chr17 [181936, 182046]      + |    212609
> data(snpgr17)
> snpgr17[1:2]
GRanges with 2 ranges and 0 elementMetadata values
          seqnames       ranges strand |
             <Rle>    <IRanges>  <Rle> |
rs1106176    chr17 [6934, 6934]      * |
rs6420494    chr17 [7214, 7214]      * |
> mean(snpgr17[bestAt.7] %in% exloc)
[1] 0.271
```

VariantAnnotation package

- extras.fdr reviews how to use VariantAnnotation to obtain coding predictions for variants; you have sufficient resources

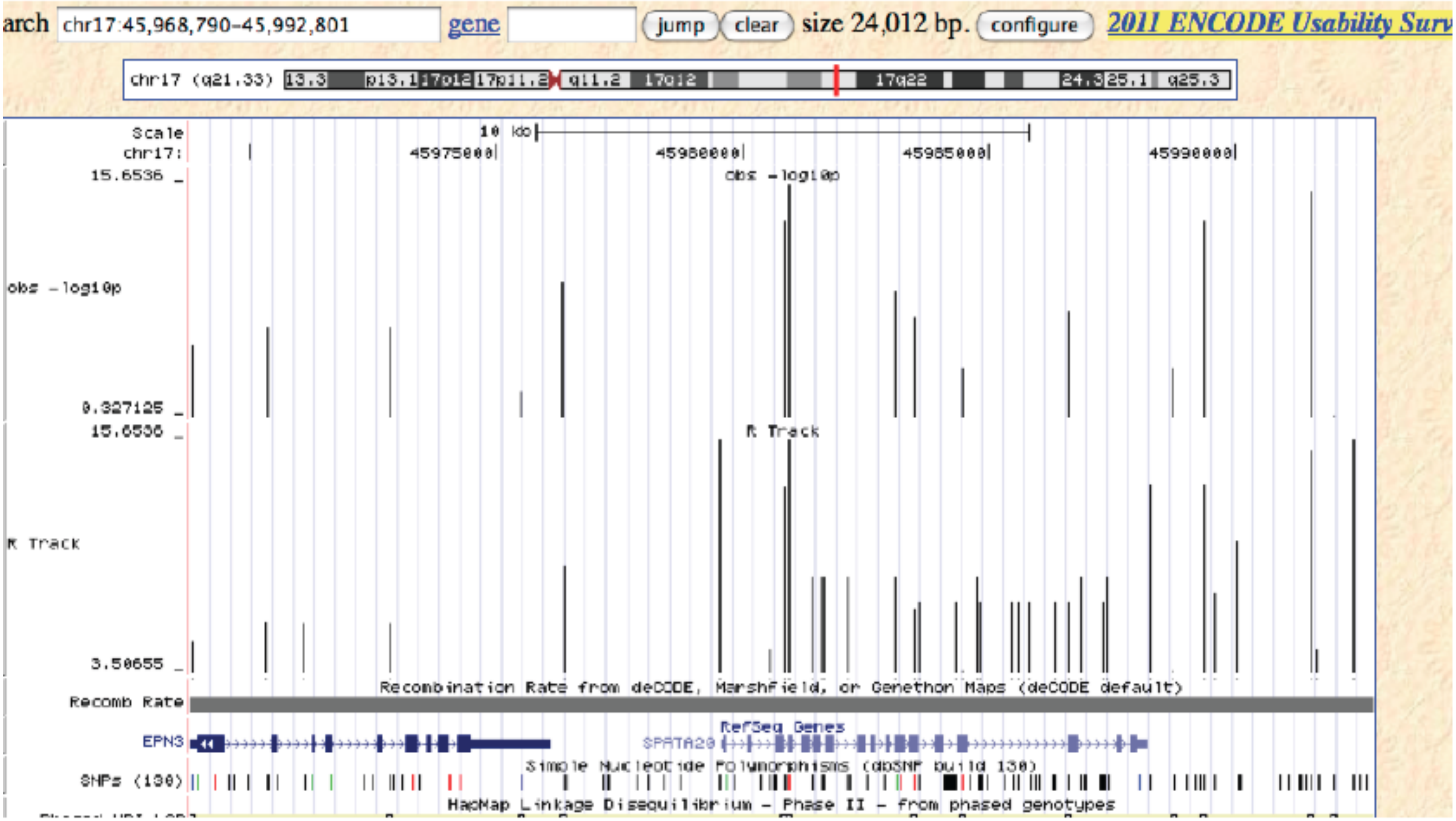
Improving resolution with population genomics models for imputation of unobserved loci: start with 1KG calls



```
> library(GGtools)
> library(Rsamtools)
> exts  = seq(1, 80e6+10, by=10e6)
> st = exts[-length(exts)]
> en = exts[-1]-1
> # following file is 66 GB from 1000genomes.org
> tf = TabixFile("ALL.2of4intersection.20100804.genotypes.vcf.gz")
> gg = GRanges(seqnames="17", IRanges(st,en))
> for (i in 1:length(gg)) {
+   vv = vcf2sm(tf, gr=gg[i], nmetacol=9L)
+   intsave(vv, file=paste("vv", i, ".rda", sep=""))
+ }
```

```
> data(rules.n43)  # identify those unobserved for N=47
> rules.n43[1:3]
rs1106176 ~ No imputation available
rs6420494  ~  rs11654695+rs9789059+rs8073513+rs7225087 (MAF =
0.128, R-squared = 0.901)
rs6420495  ~  rs11654695+rs12449775+rs8078223+rs9907102 (MAF
= 0.163, R-squared = 0.802)
> summary(rules.n43)
             SNPs used
R-squared     1 tags 2 tags 3 tags 4 tags   <NA>
  [0,0.1)       1514   1846    854    868      0
  [0.1,0.2)        6    920   1399   2053      0
  [0.2,0.3)        0    296    656   3327      0
  [0.3,0.4)        0    191    413   3005      0
  [0.4,0.5)        0    127    231   2864      0
  [0.5,0.6)        1    179    247   2722      0
  [0.6,0.7)        3    296    261   2451      0
  [0.7,0.8)       58    586    414   2840      0
  [0.8,0.9)      807   1162    925   4839      0
  [0.9,0.95)    3485   1433   1159   3893      0
  [0.95,0.99)   2473    914    707   1840      0
  [0.99,1]     33534    880   1911   5380      0
  <NA>             0      0      0      0 374836
```

Benefits of imputation



Summary to this point

- Data structures for focused and cis surveys are easy to construct/filter
- Decisionmaking on preprocessing, reduction of expression heterogeneity, filtering loci using MAF, confining to specific cis intervals, and imputing will have an impact on interpretation
- Relating results back to structural metadata made easier with IRanges, VariantAnnotation

Two topics remain

- Working with trans surveys: we provide information relating SNP on chr17 to genes on chr1 and chr9
  - We'll review pp14-15 of extras.pdf
- Working with short read archives to reason about the existence of allelically imbalanced transcription

Short reads with RNA-seq

- The PLoS Genetics paper of Cheung 2010 is accompanied by a GEO submission including MAQ-aligned short read sets for 41 individuals, e.g.:
    - -rw-r--r-- 1 stvjc st 1167812886 Apr 20 20:59 GSM424339_GM12003.map.gz
    - -rw-r--r-- 1 stvjc st 1347675340 Apr 20 17:46 GSM424338_GM11994.map.gz
- These are binary files with no embedded experiment-level metadata that I am aware of

Transforming to BAM

- samtools-0.1.12a/misc/maq2sam-short GSM424321_GM06993.map.gz > gm06993.sam
- 1137 wget http://hgdownload.cse.ucsc.edu/goldenPath/hg18/chromosomes/chr1.fa.gz
- 1138 gunzip chr1.fa.gz
- 1142 samtools faidx chr1.fa
- 1147 grep NC_000001 gm06993.sam > gm06993_1.sam
- 1154 vi chr1.fa
- 1155 samtools faidx chr1.fa
- 1158 samtools view -T chr1.fa -S -b gm06993_1.sam > gm06993_1.bam

41 lightly manually transformed MAQ alignments are in ggtut/inst/bam

```
> bff=PileupFiles(dir(system.file("bam", package="ggtut"),
     patt="bam$", full=TRUE))
> bff
class: PileupFiles
names:  (0 total)
plpFiles: litgm06985.bam, ..., litgm12891_1.bam (41 total)
plpParam: class PileupParam
> plpFiles(bff)
BamFileList of length 41
> plpFiles(bff)[[1]]
class: BamFile
path: /Users/stvjc/ISMB11_MACLIB/ggtut/bam/litgm06985.bam
index: /Users/stvjc/ISMB11_MACLIB/ggtut/bam/litgm06985.bam
isOpen: FALSE
```

New S4 ReferenceClass discipline gives efficiencies for working with BAM

```
> getClass("BamFile")
Reference Class "BamFile":

Class fields:

Name:      .extptr         path       index
Class: externalptr   character   character

 Class Methods: 
    "callSuper", "copy", "export", "field", 
"getClass", "getRefClass", "import", 
"initFields", "trace", "untrace"


 Reference Superclasses: 
    "RsamtoolsFile", "envRefClass"
```

When metadata are not bound early for propagation…

```
> bp = sapply(plpFiles(bff), path)
> no1 = gsub(".*bam.lit..", "", bp)
> no2 = gsub("_1.bam", "", no1)
> no3 = gsub(".bam", "", no2)
> nanames = paste("NA", no3, sep="")
> names(bff) = nanames
> bff
class: PileupFiles
names: NA06985, ..., NA12891 (41 total)
plpFiles: litgm06985.bam, ...,
litgm12891_1.bam (41 total)
plpParam: class PileupParam
```

The plpParam, unpopulated

```
> plpParam(bff)
class: PileupParam
plpFlag: keep0=2047 keep1=2047
plpMinBaseQuality: 13
plpMinMapQuality: 0
plpMinDepth: 0
plpMaxDepth: 250
plpYieldSize: 1
plpYieldBy: range
plpYieldAll: FALSE
plpWhat: 'seq' 'qual'
plpWhich: GRanges (length 0)
```

Bear with me

```
> setGeneric("callFreqs", function(pfl, sn, loc)
standardGeneric("callFreqs"))
[1] "callFreqs"
> setMethod("callFreqs", c("PileupFiles", "character",
"numeric"),
   function(pfl, sn, loc) {
     if (length(loc)>1) stop("requires scalar loc")
     open(pfl)
     on.exit(close(pfl))
     which = GRanges(seqnames=sn, IRanges(loc,width=1))
     param = PileupParam(which=which)
     pinfo = function(x) {
       x[["seq"]][,,1]  # reduce to matrix
     }
     ans = applyPileups(pfl, pinfo, param=param)[[1]]
     colnames(ans) = names(pfl)
     ans
 })
```

The payoff

```
> CF.rs8535 = callFreqs(bff, "chr1", 111587452)
> CF.rs8535[,1:8]
  NA06985 NA06993 NA06994 NA07000 NA07022 NA07034 NA07055 NA07056
A      66       0      86      51       0       1       0     119
C       0     193      30       1      39     251      53       5
G       0       0       1       0       0       0       0       0
T       0       0       0       0       0       0       0       0
N       0       0       0       0       0       0       0       0
```

We have created a tool that obtains the call distribution at a specific genomic Coordinate, on the basis of 41 BAM files.

Quality distributions at the locus

```
> setGeneric("callQualDists", function(pfl, sn, loc)
standardGeneric("callQualDists"))
[1] "callQualDists"
> setMethod("callQualDists", c("PileupFiles",
"character", "numeric"),
+   function(pfl, sn, loc) {
+     if (length(loc)>1) stop("requires scalar loc")
+     open(pfl)
+     on.exit(close(pfl))
+     which = GRanges(seqnames=sn, IRanges(loc,width=1))
+     param = PileupParam(which=which)
+     pinfo = function(x) {
+       x[["qual"]][,,1]  # reduce to matrix
+     }
+     ans = applyPileups(pfl, pinfo, param=param)[[1]]
+     colnames(ans) = names(pfl)
+     ans
+ })
```

```
> CQ.rs8535 = callQualDists(bff, "chr1", 111587452)
> dim(CQ.rs8535)
[1] 94 41
> CQ.rs8535[1:6,1:6]
  NA06985 NA06993 NA06994 NA07000 NA07022 NA07034
!       0       0       0       0       0       0
"       0       0       0       0       0       0
#       0       0       0       0       0       0
$       0       0       0       0       0       0
%       0       0       0       0       0       0
&       0       0       0       0       0       0
```

```
> qdists = lapply(1:41, function(x)rep(rownames
(CQ.rs8535),CQ.rs8535[,x]))
> names(qdists) = colnames(CQ.rs8535)
> lapply(qdists, table)[1:4]
$NA06985
 :  ?  @  <  =  >  2  6  7  8  9  A  B  C 
 1  2  5  3  1  3  1  2  2  1  1 13 23  8 

$NA06993
 ;  :  ?  @  =  >  2  9  A  B  C 
 1  1  8  9  1  6  2  4 27 98 36 

$NA06994
 ;  :  ?  .  @  /  <  =  >  0  3  4  5  6  7  9  A  B  C 
 5  1  8  2 10  1  6  5 10  1  3  1  3  1  3  5 10 35  7 

$NA07000
 ?  >  6  B  C 
 2  1  1 35 13 
```

Final exercise

- Transform the Phred quality tokens derived in the previous slide to numeric quantities facilitating comparison of quality distributions across samples
- Check whether there is an association of average call quality for a sample and the allelic imbalance reported at that sample