Fostering sensitivity analysis for genome-scale inference

"Software into ideas"

Vince Carey, Ph.D.
Channing Division of Network Medicine
Harvard Medical School
PSB 2013/NSF BIGDATA Add-on

Road map of the talk

* Brief discussion of generalized linear models
- Examples of genome-scale inference
  - eQTL enumeration [modest volume]
  - dsQTL enumeration [high volume]
- Sensitivities and greedy tuning
- Holistic workflows: the burden of the past
- The MAMS principles (Multiply-Agnostic, Multiply-Scalable) for statistical algorithm deployments

Obituary

Robert William Maclagan Wedderburn, 1947–1975

Robert Wedderburn, a Fellow of the Society since 1969, died suddenly and unexpectedly in June 1975 while on holiday. He was born in Edinburgh. He attended Fettes College from 1960 to 1965 and took his degree and the Diploma in Statistics at Cambridge. He joined the Statistics Department at Rothamsted Experimental Station immediately afterwards and worked there until his death. During this short period he established himself within the Department as someone with a quite unusual width of knowledge. Not only was he familiar with a great range of statistical theory, much of it far outside his immediate interests, but he had also a substantial command of modern mathematics. Further, this theoretical ability was combined with a sharp eye for the patterns in experimental data, giving him the ideal equipment for a statistician. A voracious reader, he was becoming much in demand as a referee because he could spot a false step in a proof unerringly and, just as usefully, see how two pages of algebra could be condensed to a few lines.

J. R. Statist. Soc. A,
(1972), **135**, *Part* 3, p. 370

Generalized Linear Models

By J. A. NELDER and R. W. M. WEDDERBURN

Rothamsted Experimental Station, Harpenden, Herts

SUMMARY

The technique of iterative weighted linear regression can be used to obtain maximum likelihood estimates of the parameters with observations distributed according to some exponential family and systematic effects that can be made linear by a suitable transformation. A generalization of the analysis of variance is given for these models using log-likelihoods. These generalized linear models are illustrated by examples relating to four distributions; the Normal, Binomial (probit analysis, etc.), Poisson (contingency tables) and gamma (variance components).

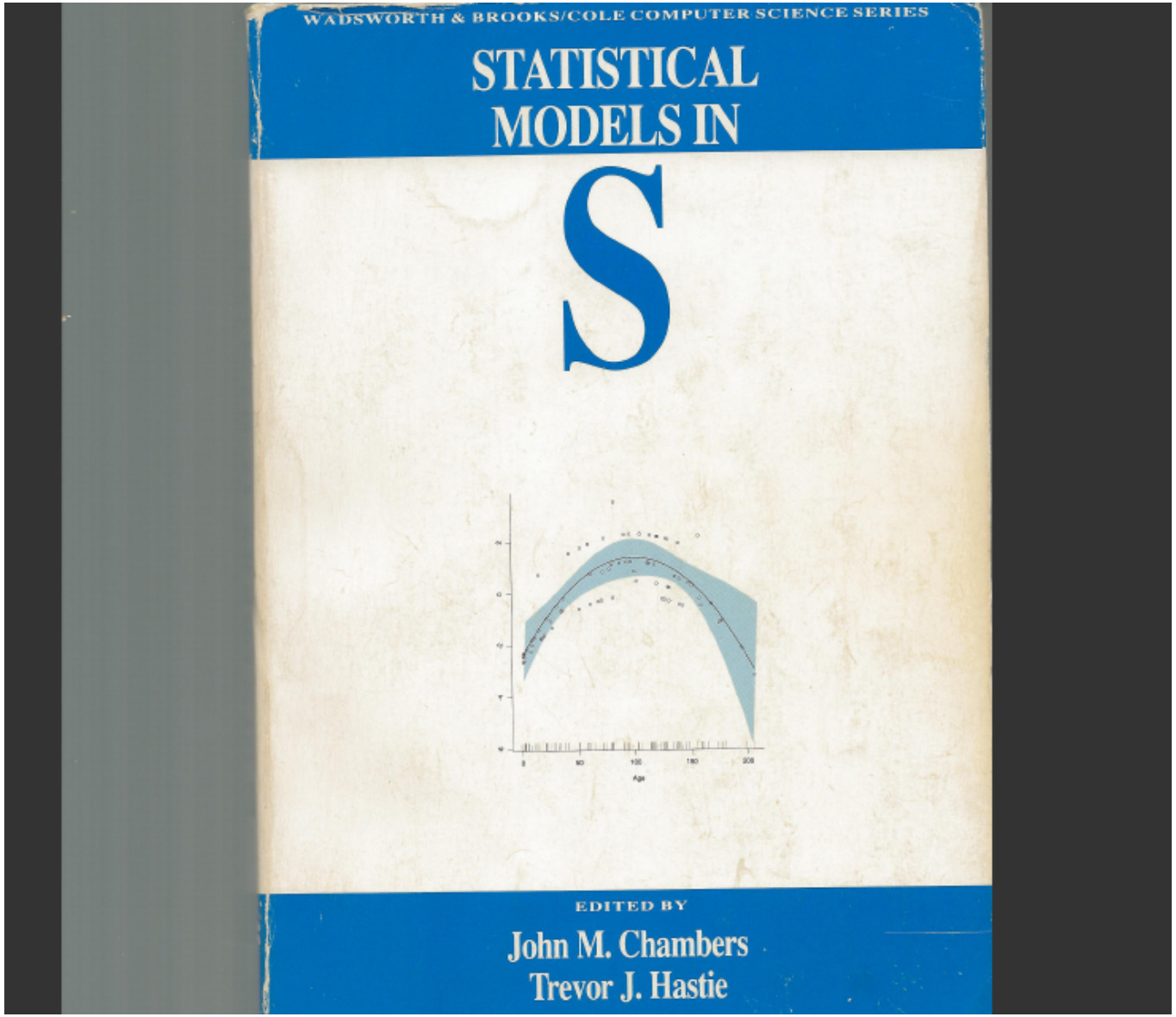
The implications of the approach in designing statistics courses are discussed.

GLM: A productive unification of statistical models, 1972

- Scalar outcome variable Y has mean value μ
- The mean is **linked** to a linear predictor

  $g(\mu) = \alpha + x_1\beta_1 + \ldots + x_p\beta_p$
- The **variance** is a **function** of the mean
  - $\mathrm{Var}(Y) = \phi V(\mu)$
- Choices of g() and V() correspond to Gaussian, Logistic, Poisson, Gamma regression procedures
- Iteratively reweighted least squares can be used for estimation; asymptotically statistically efficient under mild assumptions
- Reprinted in “Breakthroughs in statistics”, along with works of Fisher, Student, Pearson, Wald, ….

1992: deployment as `glm()`



```
> dimnames(glm.variances)
[[1]]:
[1] "name"     "variance" "deviance"

[[2]]:
[1] "constant" "mu(1-mu)" "mu"       "mu^2"     "mu^3"
```

We see that each column of `glm.links` is a link subfamily with five elements, and each column of `glm.variances` is a variance subfamily with three elements. The family generator functions, such as `binomial()` and `poisson()`, protect the user against bad choices; for example, only `logit`, `probit`, and `cloglog` are permissible links when constructing a binomial family.

There are several ways to modify the families and construct private ones:

- The `quasi()` function can be used to build a family from the supplied links and variances whose names appear in the two lists above.
- Users can build their own link or variance subfamilies (by mimicking any of the supplied ones). These can then be used to construct a family, either using `quasi()` or the function `make.family()`.

GLM: 40 years of theory, extension, deployment

- GENSTAT, GLIM: Numerical Algorithms Group
- S, Splus – `glm` infrastructure includes `robust()` family
- R – `stats::glm` and `biglm::bigglm` address "standard" and high-volume fitting requirements (the latter with incremental QR)
- Additional tailored deployments in Bioconductor snpStats, limma, DESeq, edgeR confront genetic and genomic requirements

Why so much time on GLM?

- Illustrates an aspect of algorithmic “holism”: a single interface, focused infrastructure solves all of a class of problems formerly treated piecemeal
- Illustrates the idea of an algorithm template that can receive user-coded functions to modify operations
- Has been re-implemented too often, and examining causes for this can help define requirements for enduring deployments

Questions

- If statisticians had discovered GLM only today, what would be a reasonable approach to implementation? How to sidestep common assumptions
  - "all data in memory"
  - scalar execution of algorithm steps
  - inputs are (mostly) floating point numbers and integers
- What languages and environments will support streamlined implementations, maximizing efficient use of available hardware/software?
- How will interactive data analysis capabilities be achieved with high data volume and environment complexity?

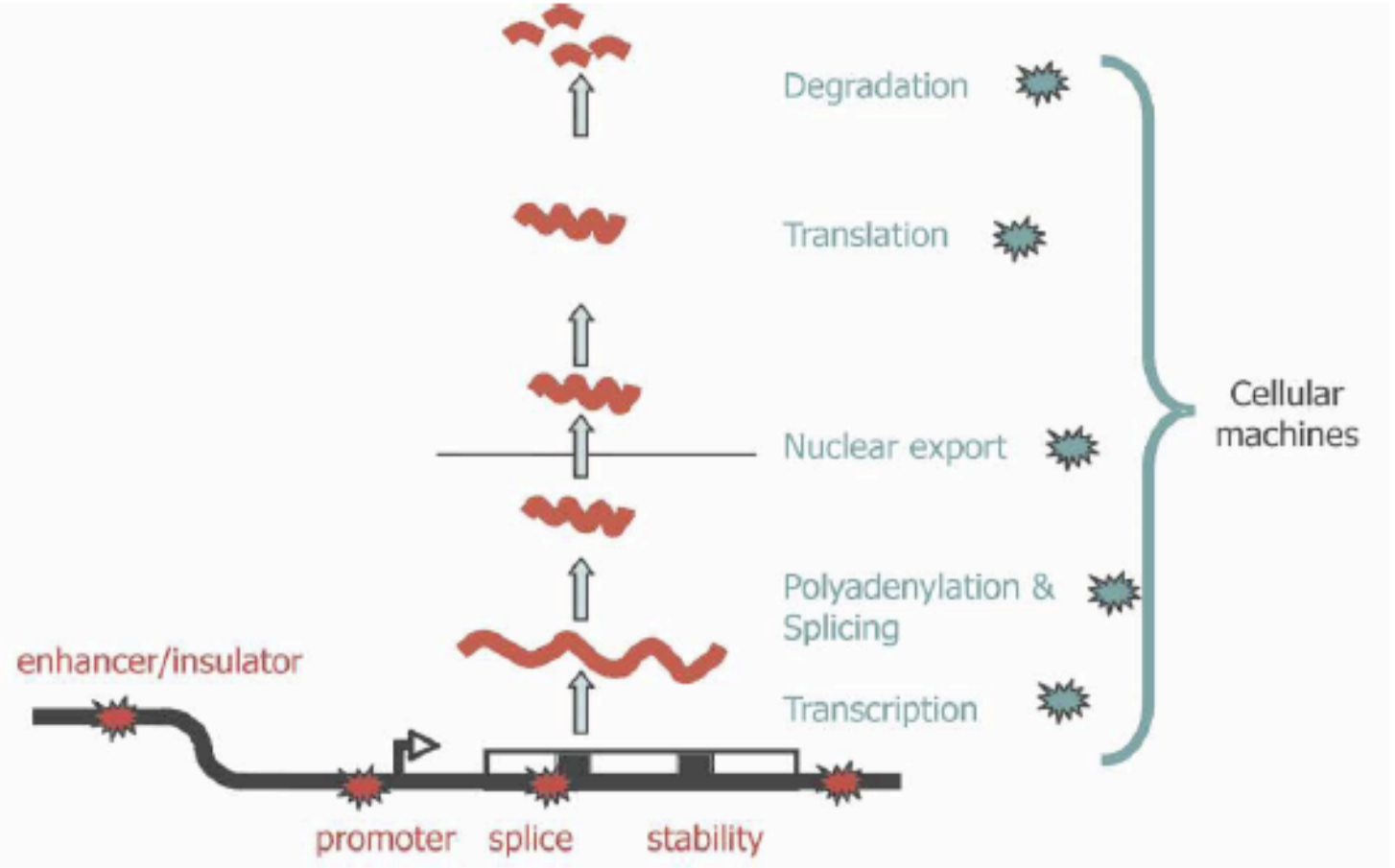
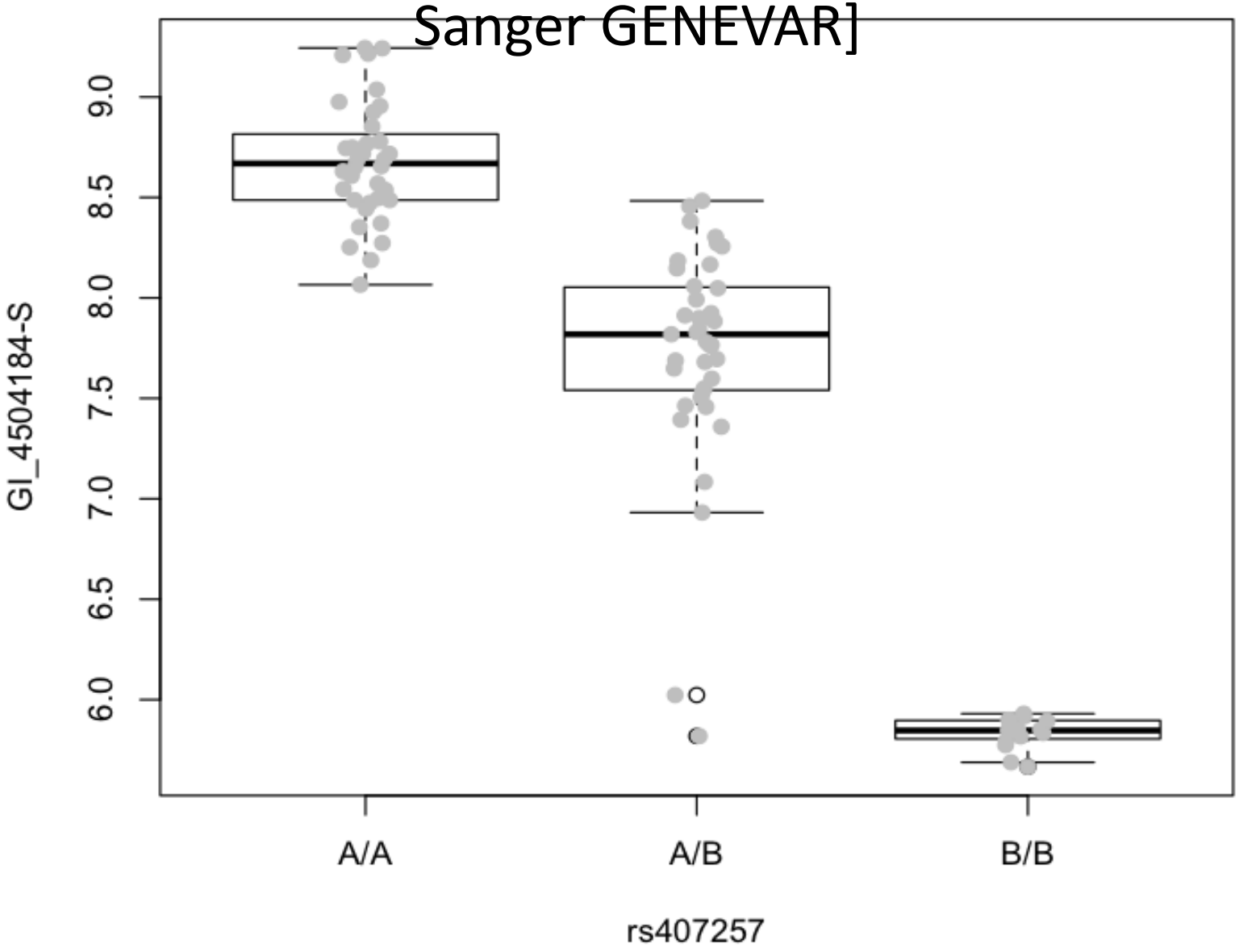


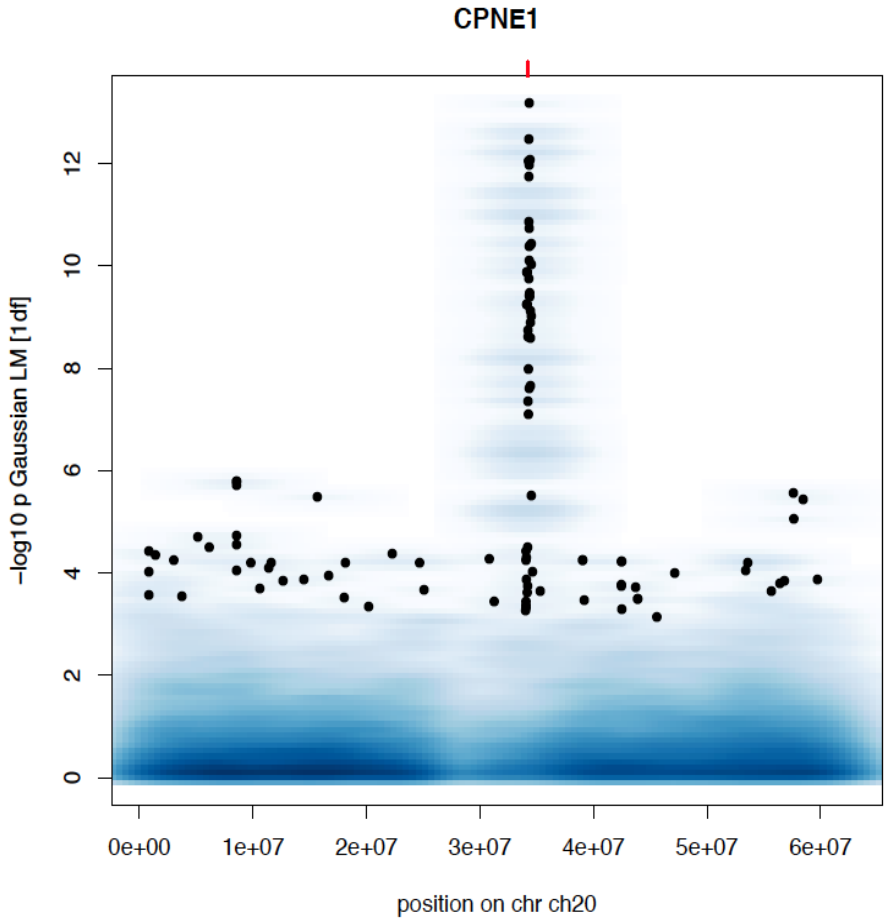
Figure 1. Plausible sites of action for genetic determinants of mRNA levels. Genetic variations influencing gene expression may reside within the regulatory sequences, promoters, enhancers, splice sites, and secondary structure motifs of the target gene and so be genetically in *cis* (red stars), or there may be variations in the molecular machinery that interact with *cis*-regulatory sequences and so act genetically in *trans* (blue stars).

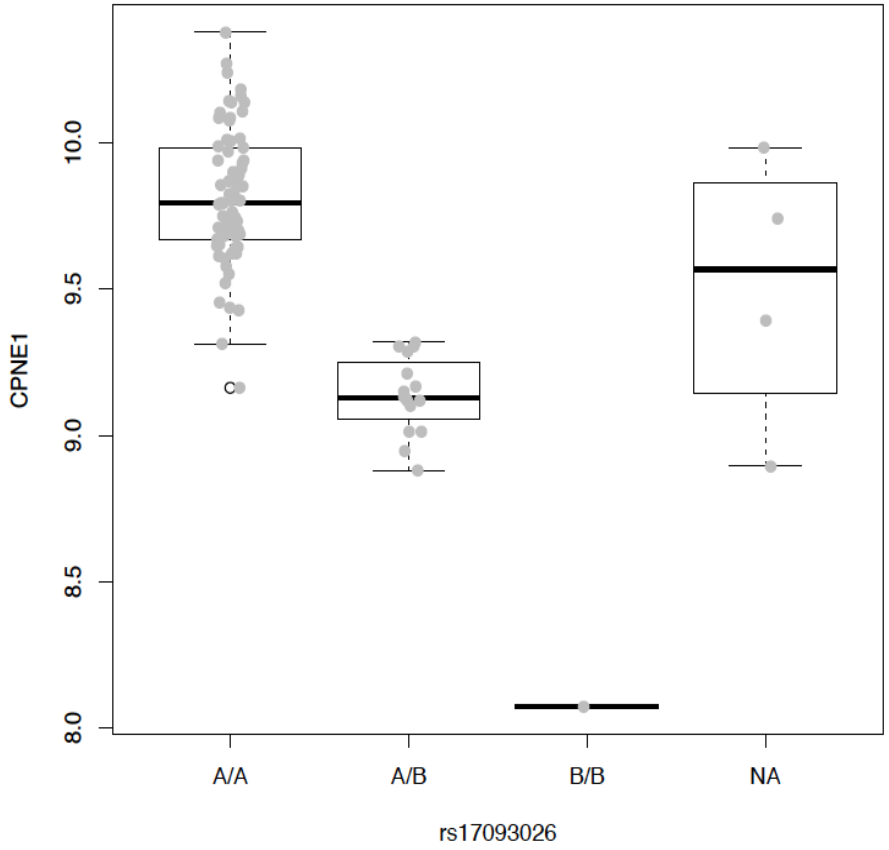
Williams R et al. Genome Research 2007 vol. 17 (12) pp. 1707-1716

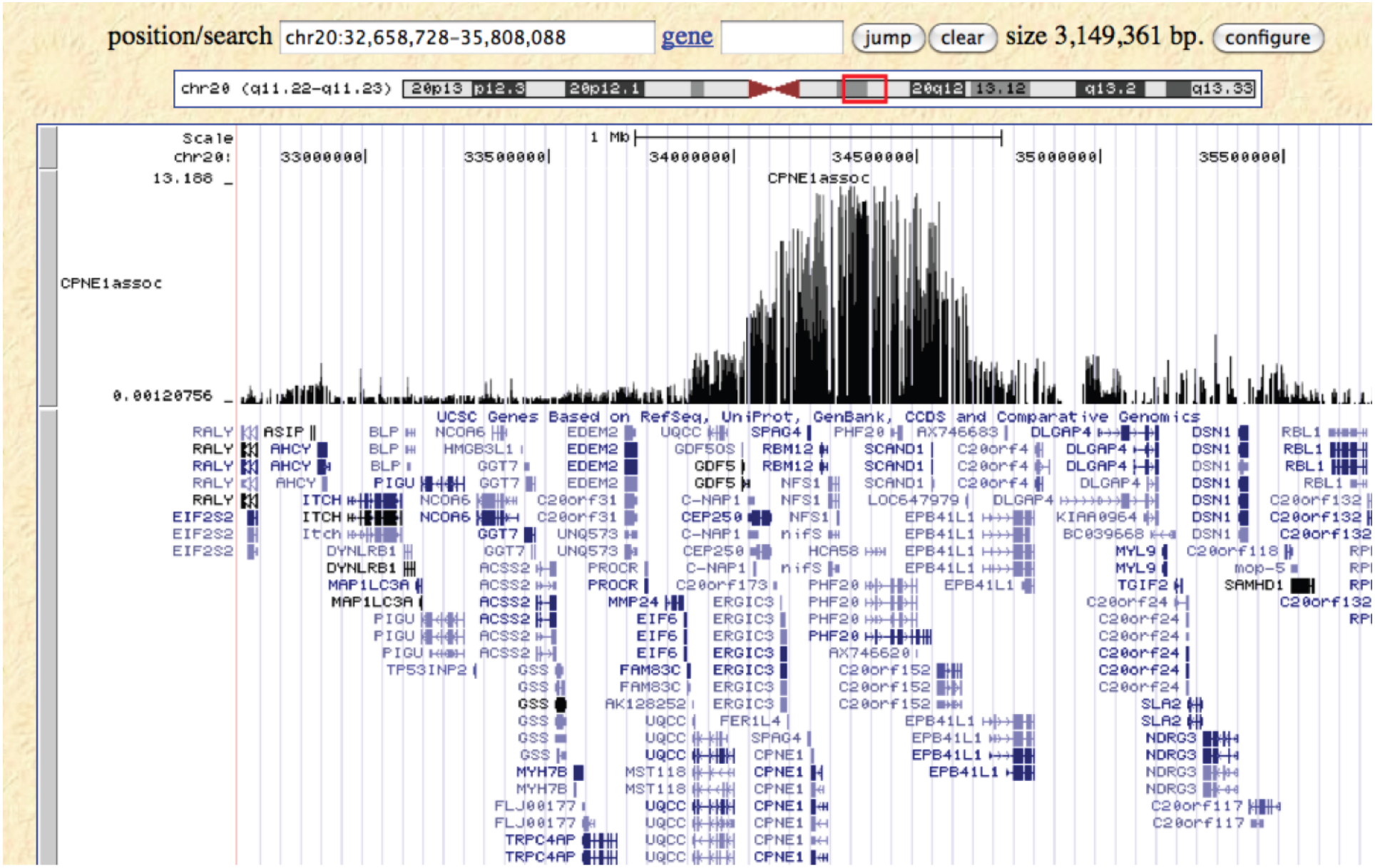
GSTT1 eQTL: Average expression varies by genotype at nearby SNPs – why? [N=90 CEU HM phase 2; Sanger GENEVAR]



Full chromosome scan for CPNE1 and view of the the peak






position/search
chr20:32,658,728-35,808,088
gene
jump
clear
size 3,149,361 bp.
configure
chr20 (q11.22-q11.23)
20p13
p12.3
20p12.1
20q12
13.12
q13.2
q13.33
Scale
chr20:
33000000
33500000
34000000
34500000
35000000
35500000
1 Mb
13.188
CPNE1assoc
0.00120756
UCSC Genes Based on RefSeq, UniProt, GenBank, CCDS and Comparative Genomics

Summary

- Transcriptome and SNP-ome are jointly measured on a number of individuals
  - ~20000 transcripts, ~10 million SNP, …
- Models for additive genetic effects on transcript levels are fit for all gene:snp pairs in cis
- Humps and peaks in the series of association statistics are found along the genome
- Reliability of the procedure, interpretation of results?

LETTER

doi:10.1038/nature10808

DNase I sensitivity QTLs are a major determinant of human expression variation

Jacob F. Degner[1,2]*, Athma A. Pai[1]*, Roger Pique-Regi[1]*, Jean-Baptiste Veyrieras[1,3], Daniel J. Gaffney[1,4], Joseph K. Pickrell[1], Sherryl De Leon[4], Katelyn Michelini[4], Noah Lewellen[4], Gregory E. Crawford[5,6], Matthew Stephens[1,7], Yoav Gilad[1] & Jonathan K. Pritchard[1,4]

The mapping of expression quantitative trait loci (eQTLs) has emerged as an important tool for linking genetic variation to changes in gene regulation[1–5]. However, it remains difficult to identify the causal variants underlying eQTLs, and little is known about the regulatory mechanisms by which they act. Here we show that genetic variants that modify chromatin accessibility and transcription factor binding are a major mechanism through which genetic variation leads to gene expression differences among humans. We used DNase I sequencing to measure chromatin accessibility in 70 Yoruba lymphoblastoid cell lines, for which genome-wide genotypes and estimates of gene expression levels are also available[6–8]. We obtained a total of 2.7 billion uniquely

and enhancer-associated histone marks. Furthermore, bound transcription factors protect the DNA sequence within a binding site from DNase I cleavage, often producing recognizable 'footprints' of decreased DNase I sensitivity[13,15–17].

We collected DNase-seq data for 70 HapMap Yoruba lymphoblastoid cell lines for which gene expression data and genome-wide genotypes were already available[6–8]. We obtained an average of 39 million uniquely mapped DNase-seq reads per sample, providing individual maps of chromatin accessibility for each cell line (see Supplementary Information for all analysis details). Our data allowed us to characterize the distribution of DNase I cuts within individual hypersensitive sites at extremely high resolution. As expected, the DHSs coincided to a great

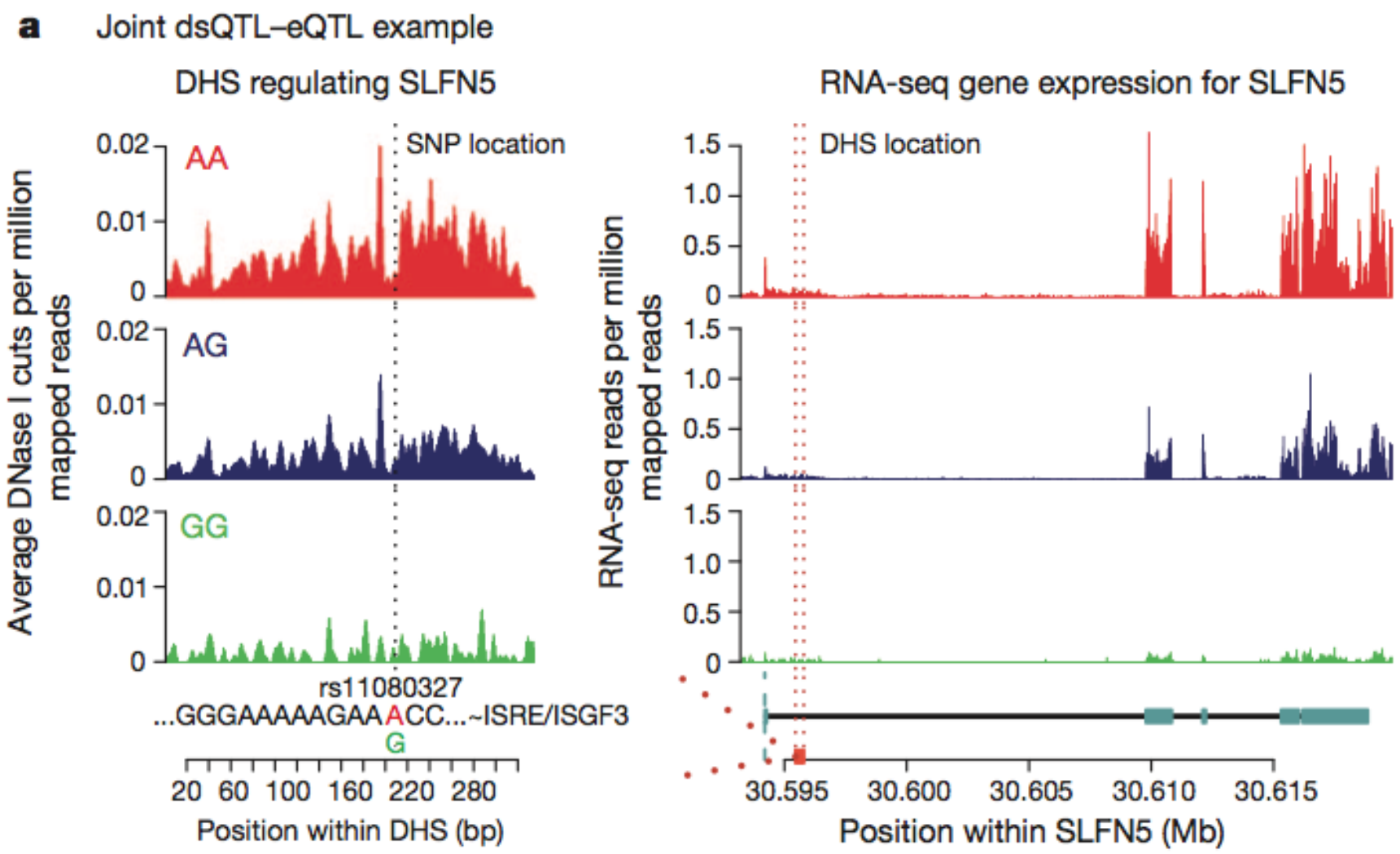
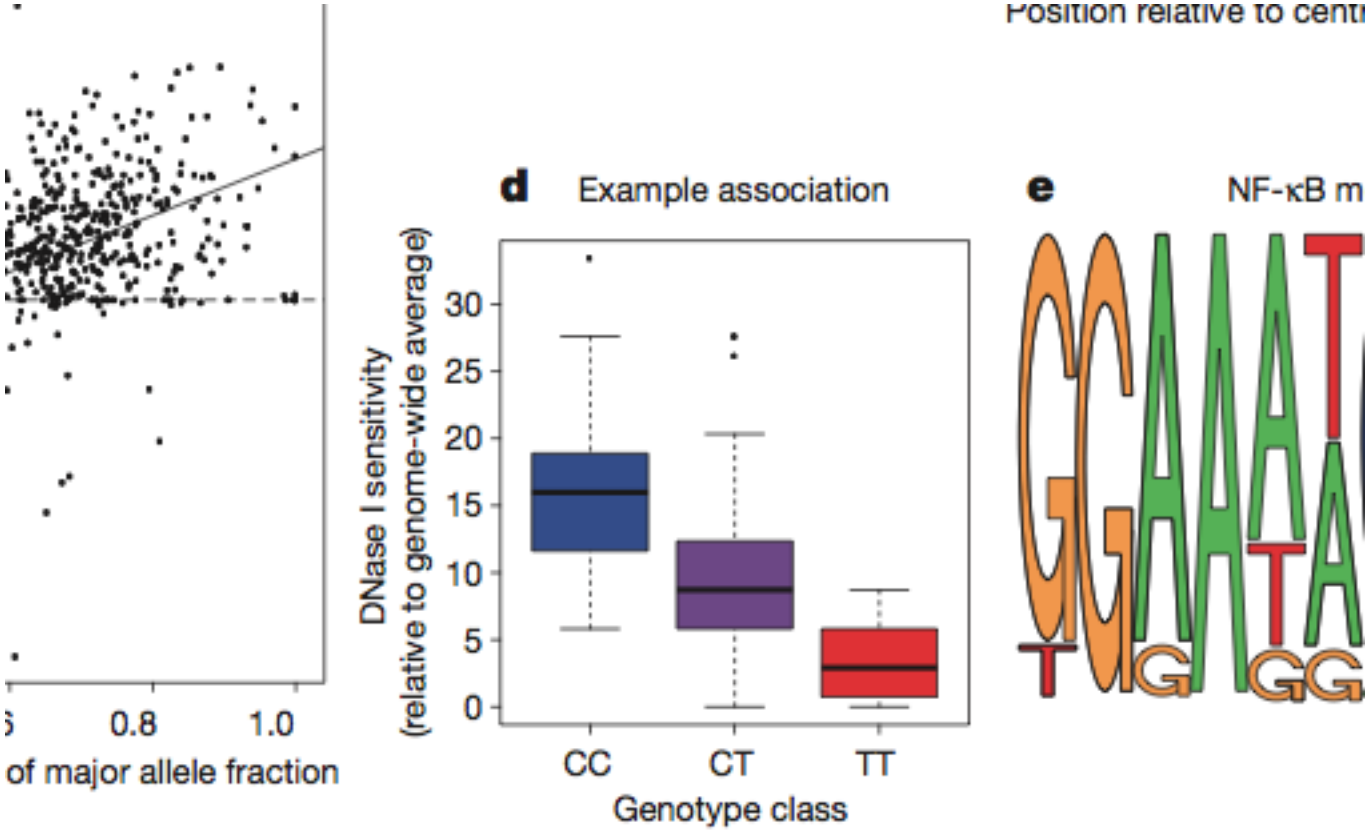


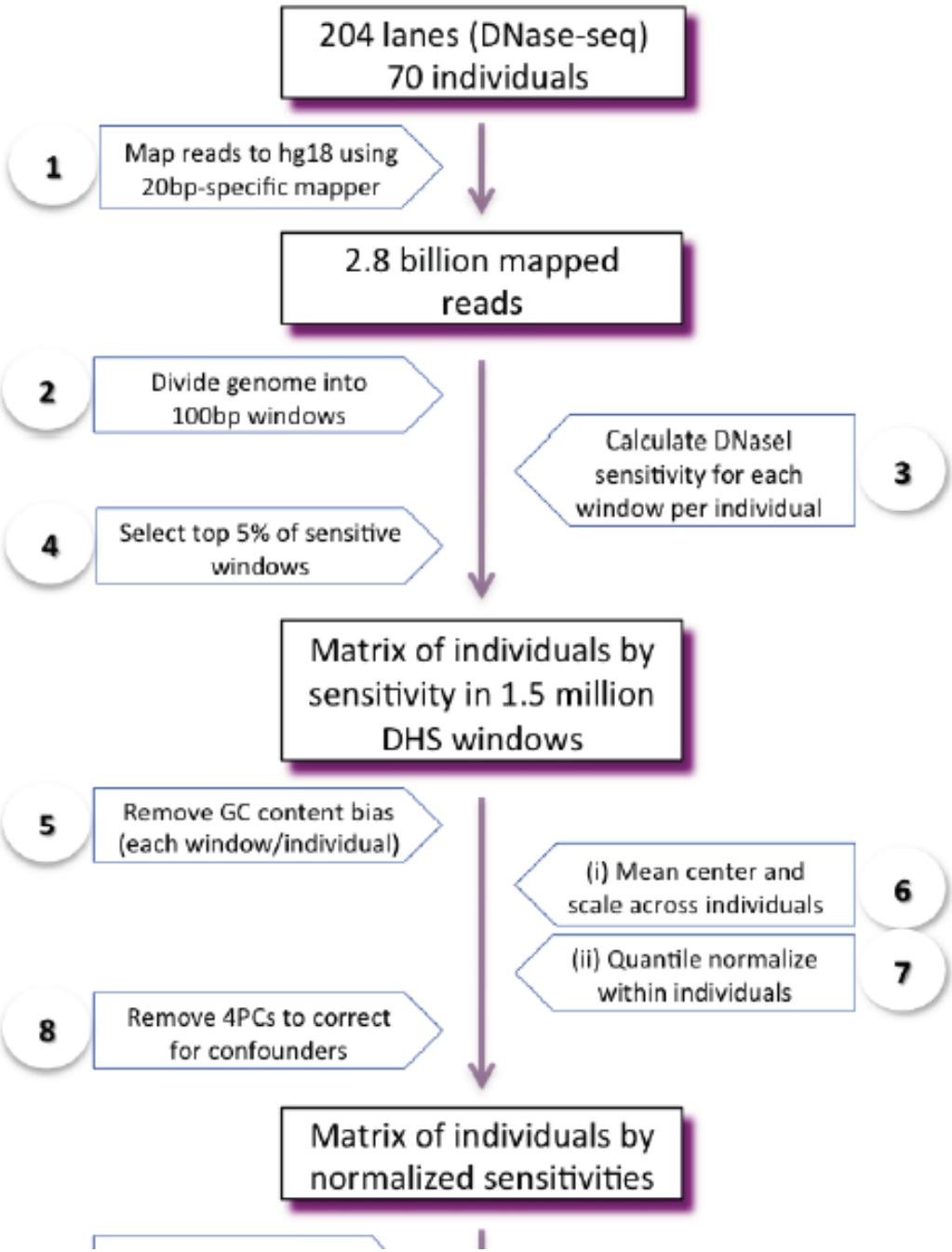
Figure 3 | Relationship between dsQTLs and eQTLs. a, Example of a dsQTL SNP that is also an eQTL for the gene *SLFN5*. The SNP disrupts an interferon- (right) measurer genotype at the p

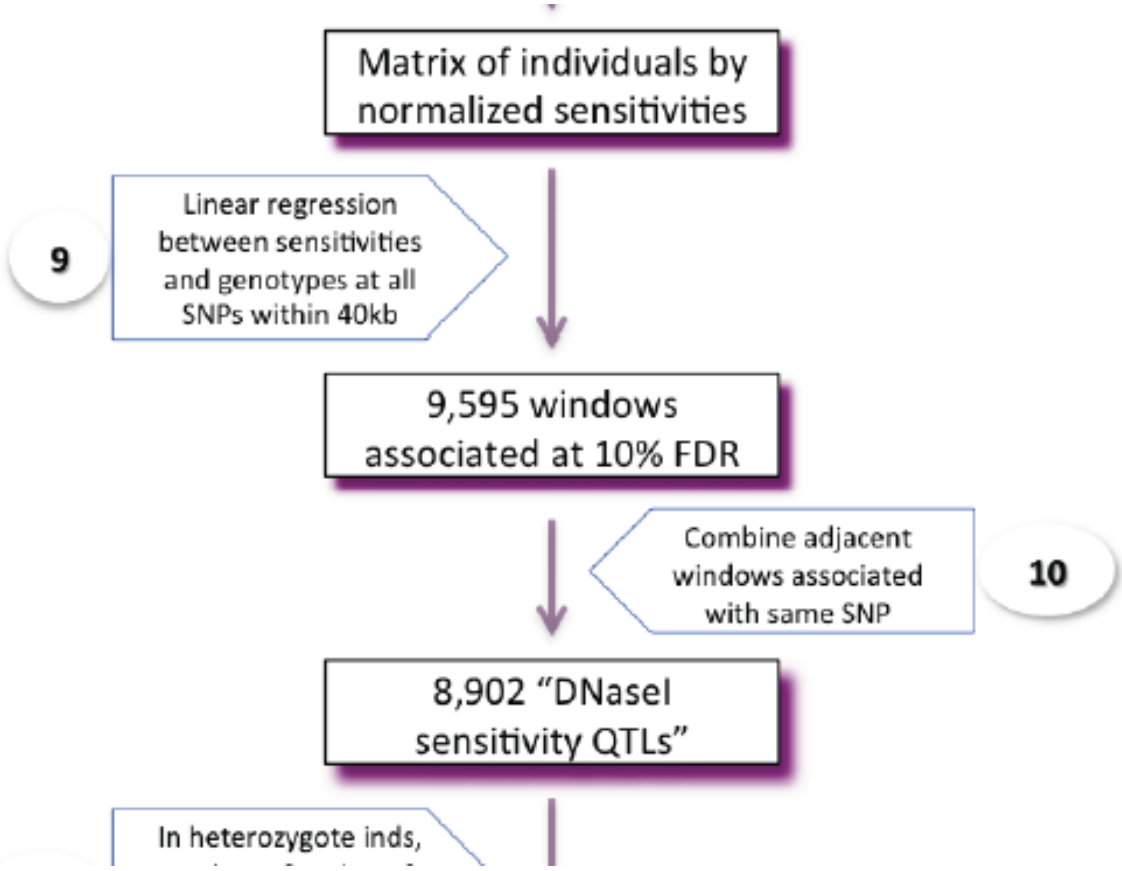


ation of dsQTLs and a typical example.
tion between DNase I cut rates in 100-bp
(green) and 40-kb (black) regions centred
lele-specific analysis of dsQTLs in

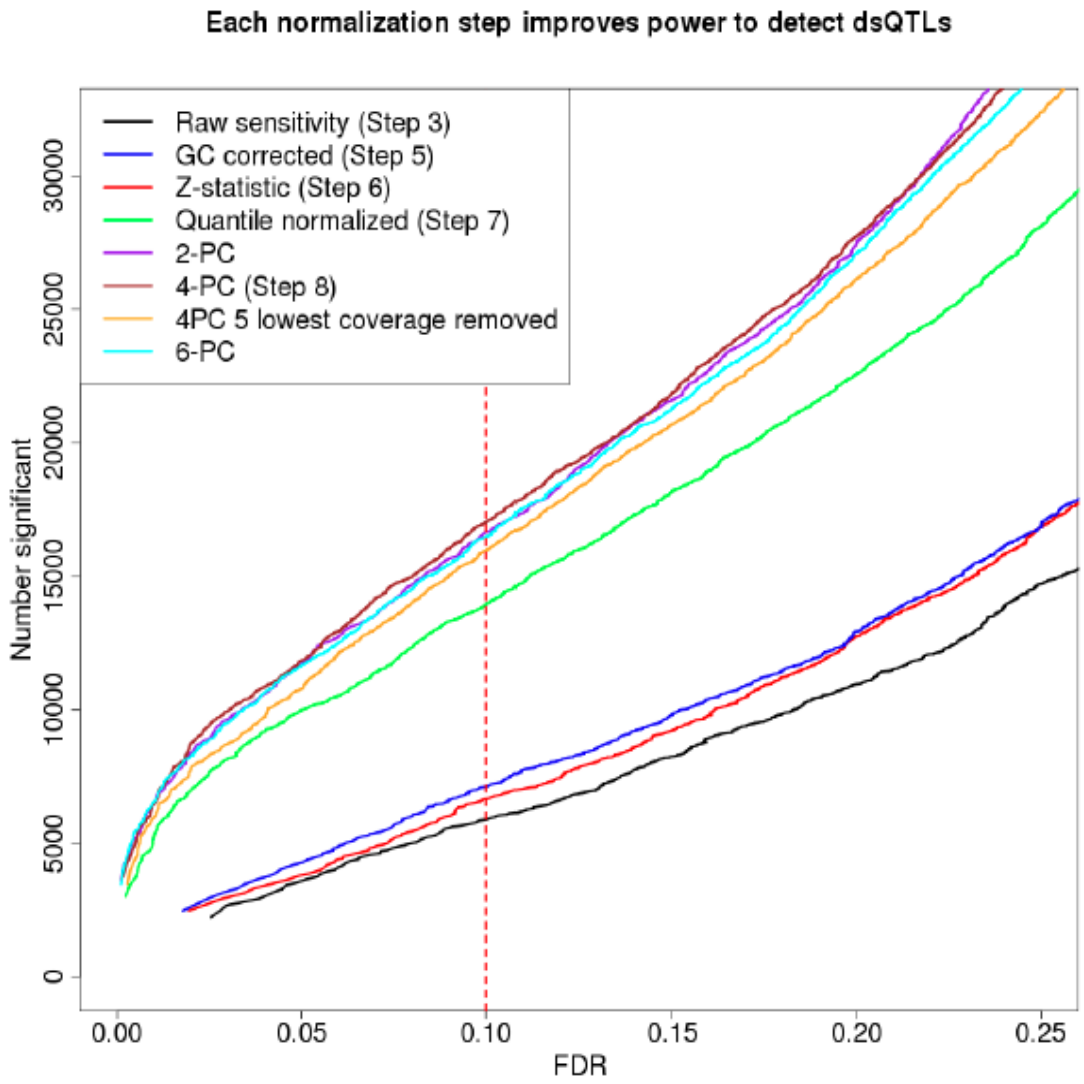
dsQTL (rs4953223). The bla
d, Box plot showing that rs4
accessibility ($P = 3 \times 10^{-13}$
DNase I sensitivity, disrupts

Tuned with
100bp window
top 5% sensitivity
4 PC removal




Matrix of individuals by normalized sensitivities
9
Linear regression between sensitivities and genotypes at all SNPs within 40kb
9,595 windows associated at 10% FDR
Combine adjacent windows associated with same SNP
10
8,902 "DNaseI sensitivity QTLs"
In heterozygote inds,

Greedy tuning for higher yield

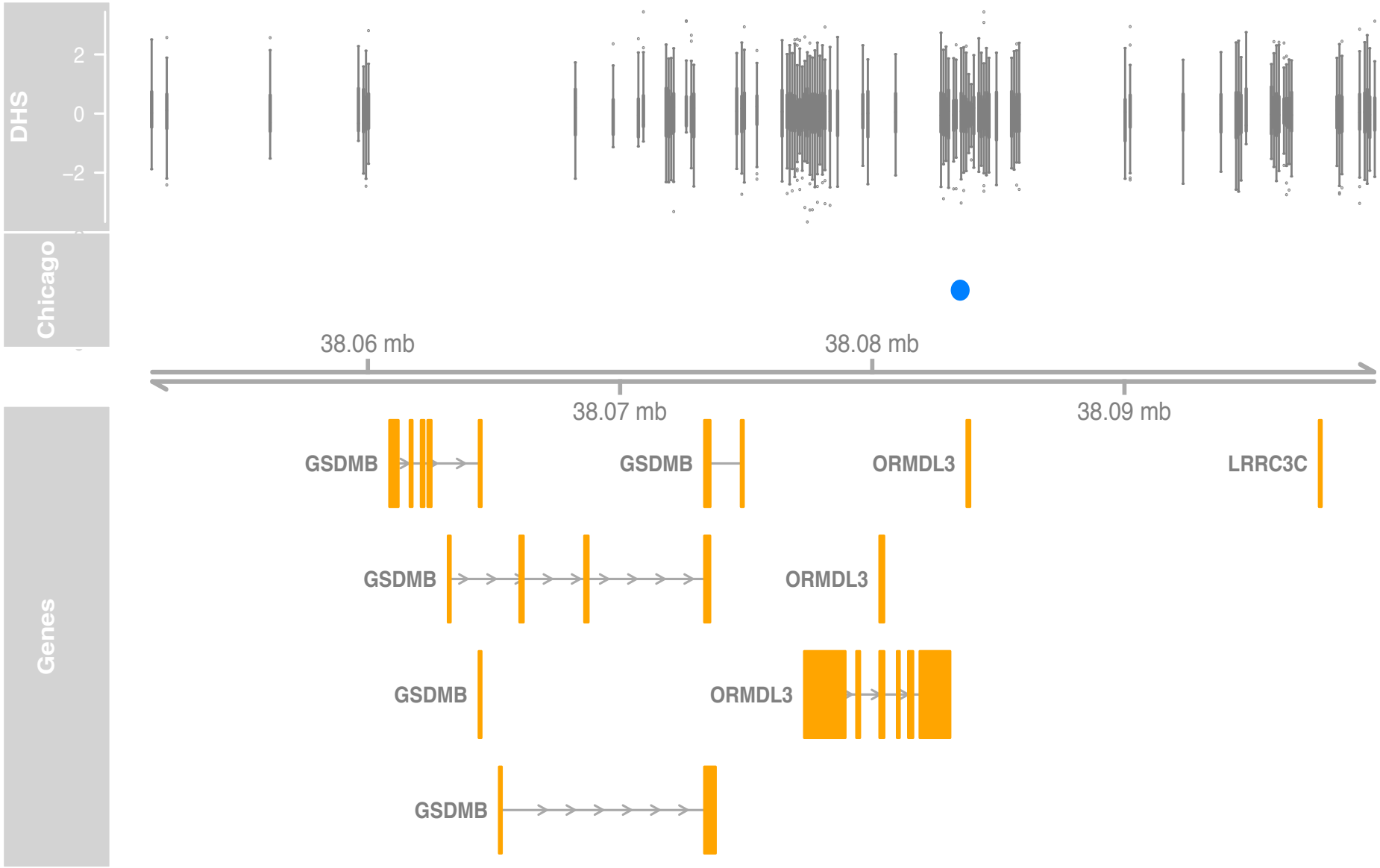


: **Power to detect dsQTLs improves with each normalization step performed.** Here, we

Summary

- Feature space now a continuously scored tiling of the genome
  - Filtered to 1.5 million features but could be many more, could consider as many as 37 million 1KG SNP
- Scope of genetic regulation seems more limited: dropping cis search region from 40kb to 2kb does not drastically affect yield of dsQTL
- A number of ad hoc filtering steps might have more important impacts

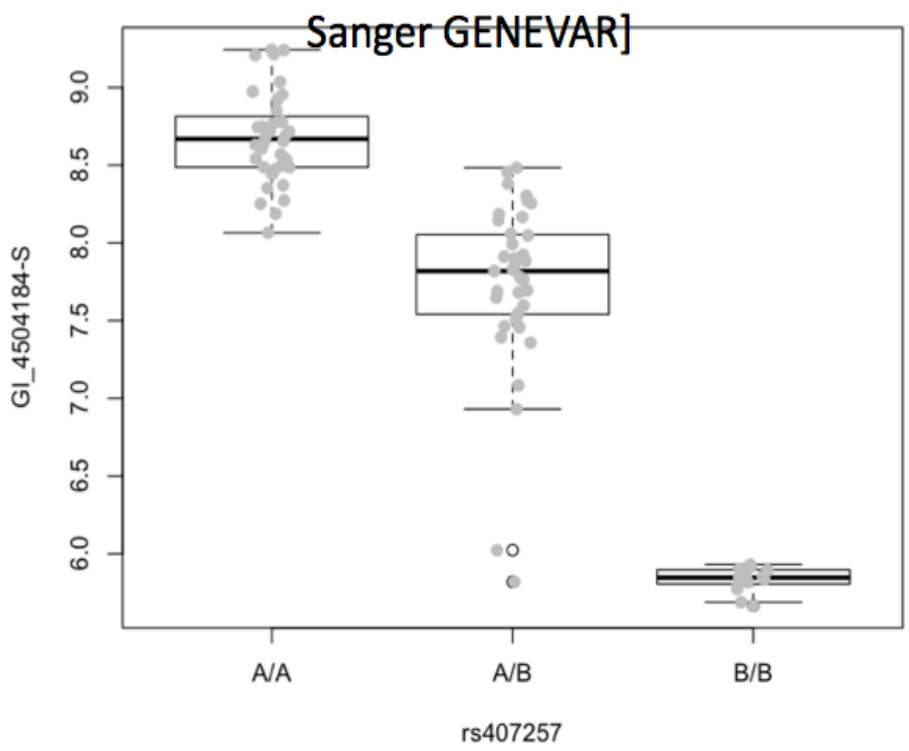
Distributions of norm. DHS over 70 individuals at most sensitive windows in vicinity of ORMDL3

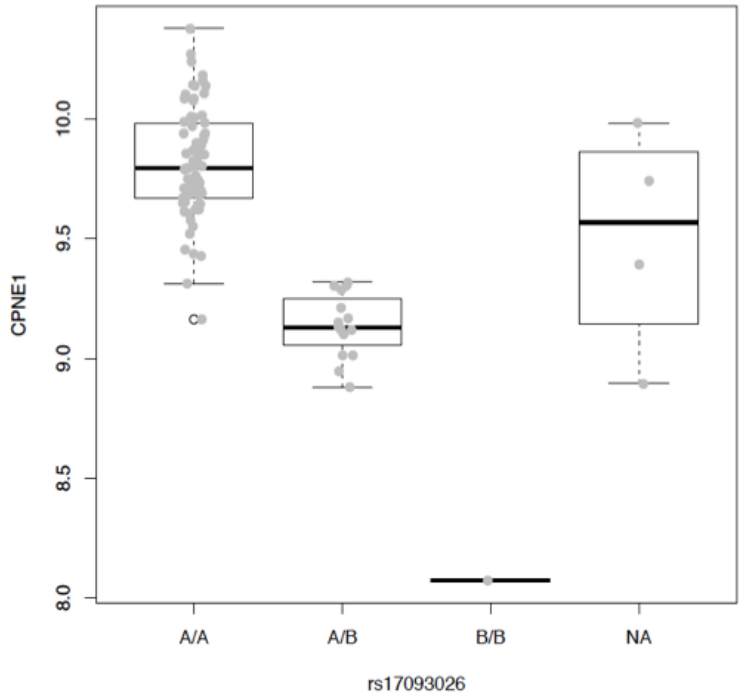


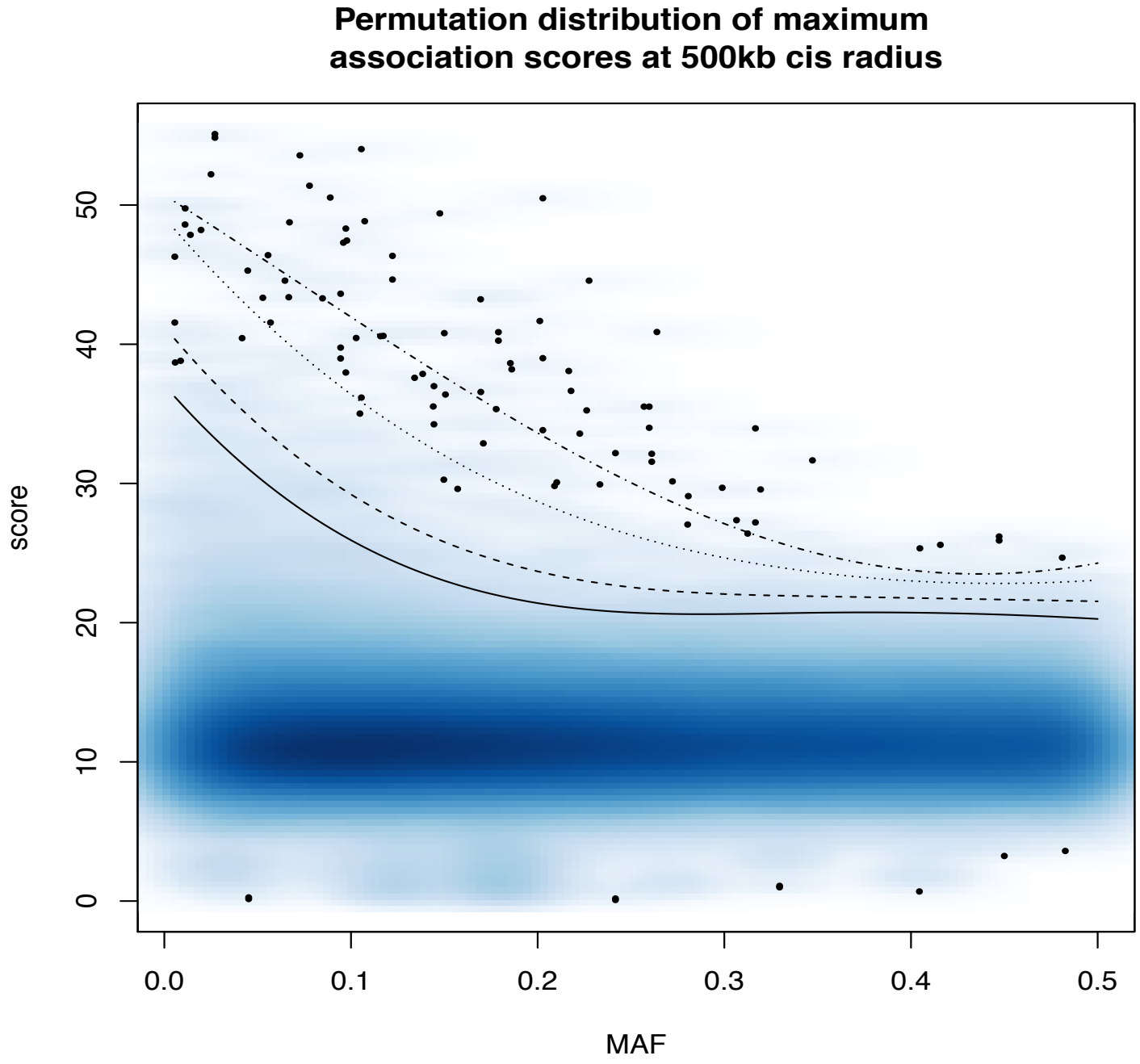
Greedy tuning of eQTL searches

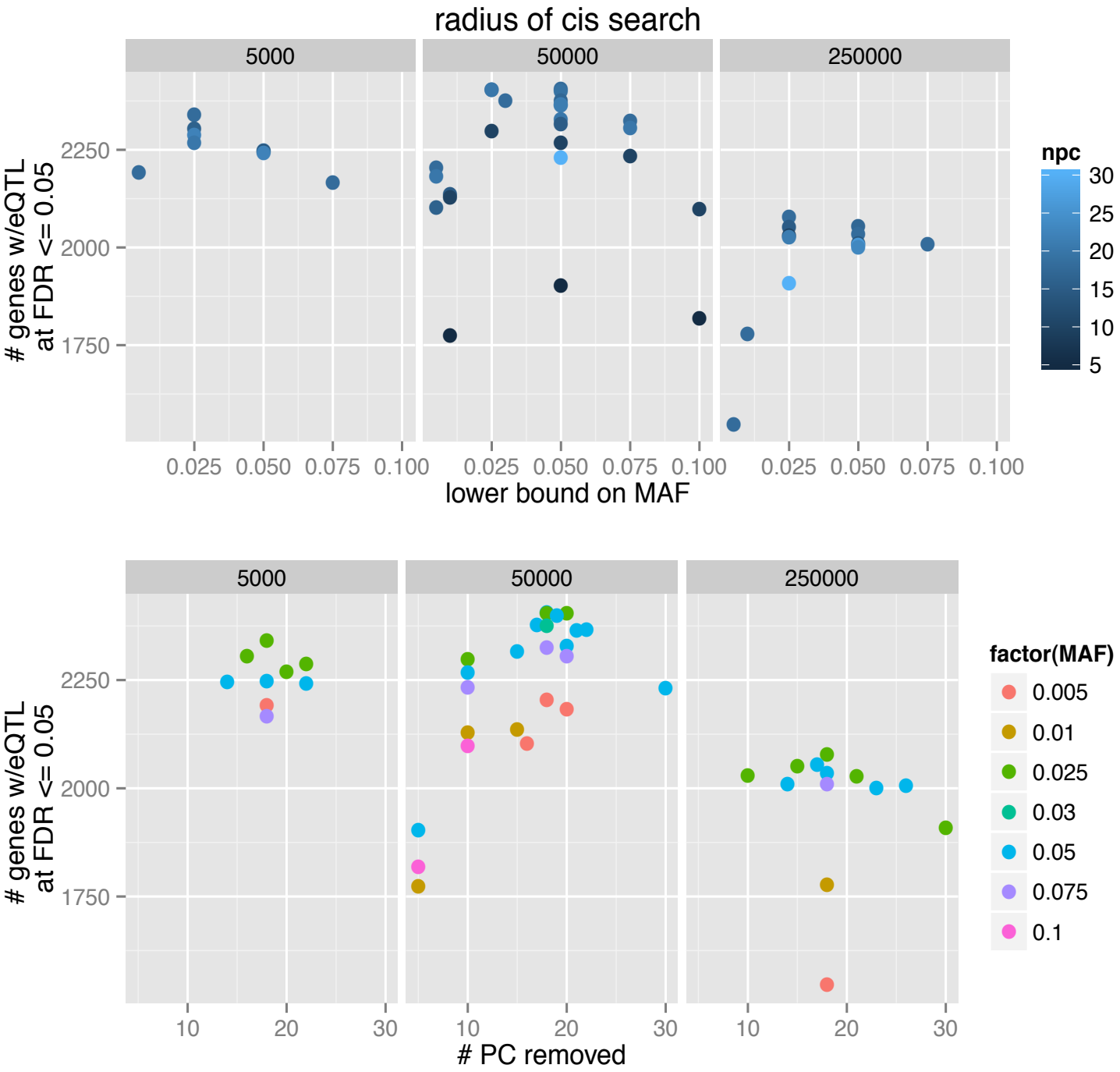
- Yield can be affected by
  - Choice of cis-interval size
  - Depth of search into rare variants (lower bound on minor allele frequency)
  - Approach to removing non-biologic variation from expression assay results (Stegle, Durbin, RECOMB 2008)
- Management of a single search is difficult, but multiple searches or extensive metadata need to be retained so that various calling policies can be compared
- We'll consider combined analysis of CEU and YRI founders (N=120)

Minor allele frequency determines reliability of association inference






Permutation distribution of maximum
association scores at 500kb cis radius
score
0
10
20
30
40
50
0.0
0.1
0.2
0.3
0.4
0.5
MAF


radius of cis search
5000
50000
250000
genes w/eQTL at FDR <= 0.05
2250
2000
1750
0.025
0.050
0.075
0.100
lower bound on MAF
npc
30
25
20
15
10
5
PC removed
10
20
30
factor(MAF)
0.005
0.01
0.025
0.03
0.05
0.075
0.1

Upshots for eQTL

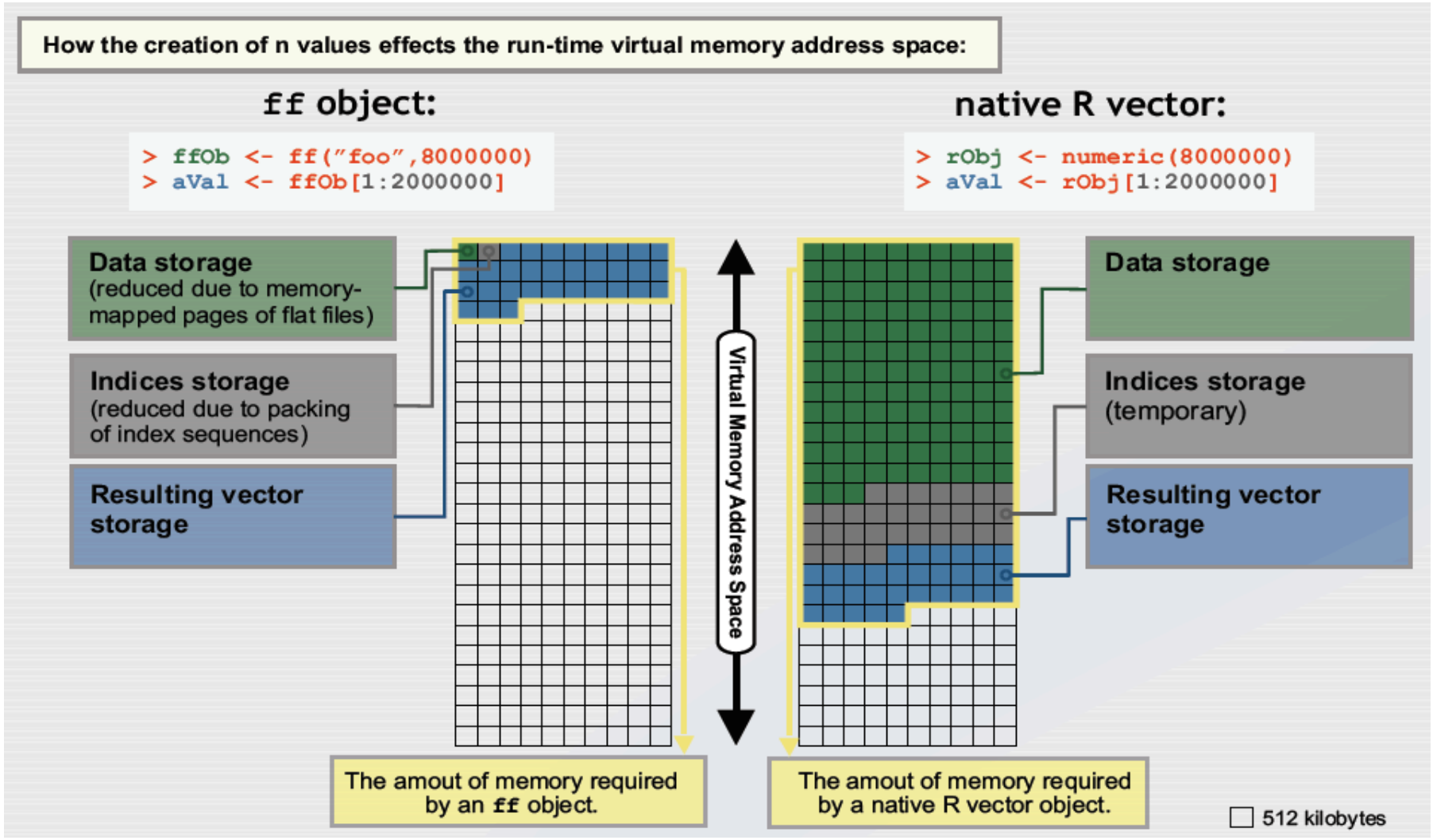
- Very large number of tests
- Evident sensitivity of yield to a number of tuning parameters
- Thorough investigations require exploration of the parameter space
- With GGtools R 2.15 the full 500kb radius,

MAF > 0.05 search took 3h on 88 commodity cores

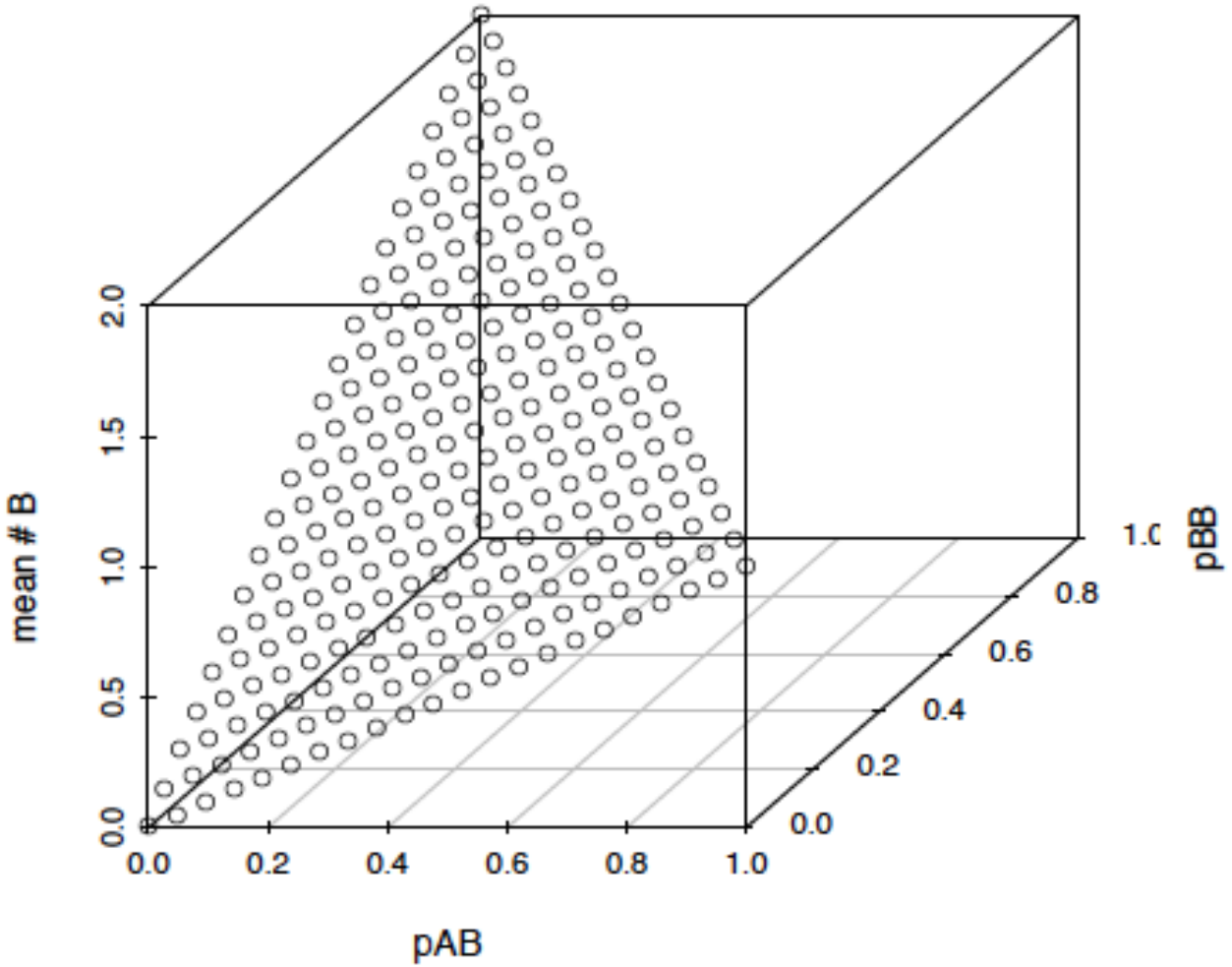
A holistic workflow?

- Computing plug-in FDR for the gene-centric null hypotheses “mean expression of Gene g is not associated with B allele frequency for any SNP within R kb” involves
  - Testing all cis associations for all genes, retaining gene-specific maximum
  - Developing multiple realizations of the permutation distribution of the maximized association
- Four innovations made this feasible 2-3 years ago: serialization, multicore, ff, snpStats

ff to reduce memory consumption



Representing (uncertain) SNP genotypes: David Clayton's byte-sized encoding



Comments

- ca. 2004 expression and genotype data could all reside in main memory – burden of the past
- "Programming around" R with big data?
  - Disk used as buffer for voluminous testing process, written to by processes on multiple cores
  - GLM rewrite for the byte representation of uncertain genotypes?
- We can achieve data compactness, speed, and ready access to statistics, visualization, genomic annotation, and "everything is an object" (with optional validity conditions)

Inputs

- Expression data and MACH-imputed genotype archives are managed in R packages
- Self-describing eSet variants combine expr/snp/sample data

```
SnpMatrix-based genotype set:
number of samples:  90
number of chromosomes present:  1
annotation: illuminaHumanv1.db
Expression data dims: 47293 x 90
Total number of SNP: 305929
Phenodata: An object of class 'AnnotatedDataFrame'
  sampleNames: NA18500 NA18501 ... NA19240 (90
  total)
  varLabels: fam samp ... isFounder (8 total)
  varMetadata: labelDescription
```

Output

```
GGtools mcwBestCis instance.  The call was:
GGtools:::combine2(mcw1 = fullrun, mcw2 = get(allob[i]))
Best loci for 21534 probes are recorded.
Top  4 probe:SNP combinations:
GRanges with 4 ranges and 5 metadata columns:
                seqnames                ranges strand |     score       snpid
                   <Rle>             <IRanges>  <Rle> | <numeric> <character>
  GI_28872735-A       10 [102246027, 103247272]      * |     89.78   rs2863095
  GI_28872737-I       10 [102246027, 103247272]      * |     85.37   rs2863095
  GI_20070185-S       10 [ 15978942,  17055744]      * |     78.91   rs1055340
  GI_28872733-I       10 [102246027, 103247272]      * |     76.51   rs2863095
                  snploc radiusUsed       fdr
                <integer>  <numeric> <numeric>
  GI_28872735-A 102746503      5e+05         0
  GI_28872737-I 102746503      5e+05         0
  GI_20070185-S  16555528      5e+05         0
  GI_28872733-I 102746503      5e+05         0
```

The search intervals are bound to the gene-centric results with the IRanges infrastructure supporting convenient comparison with other data linked to genomic coordinates

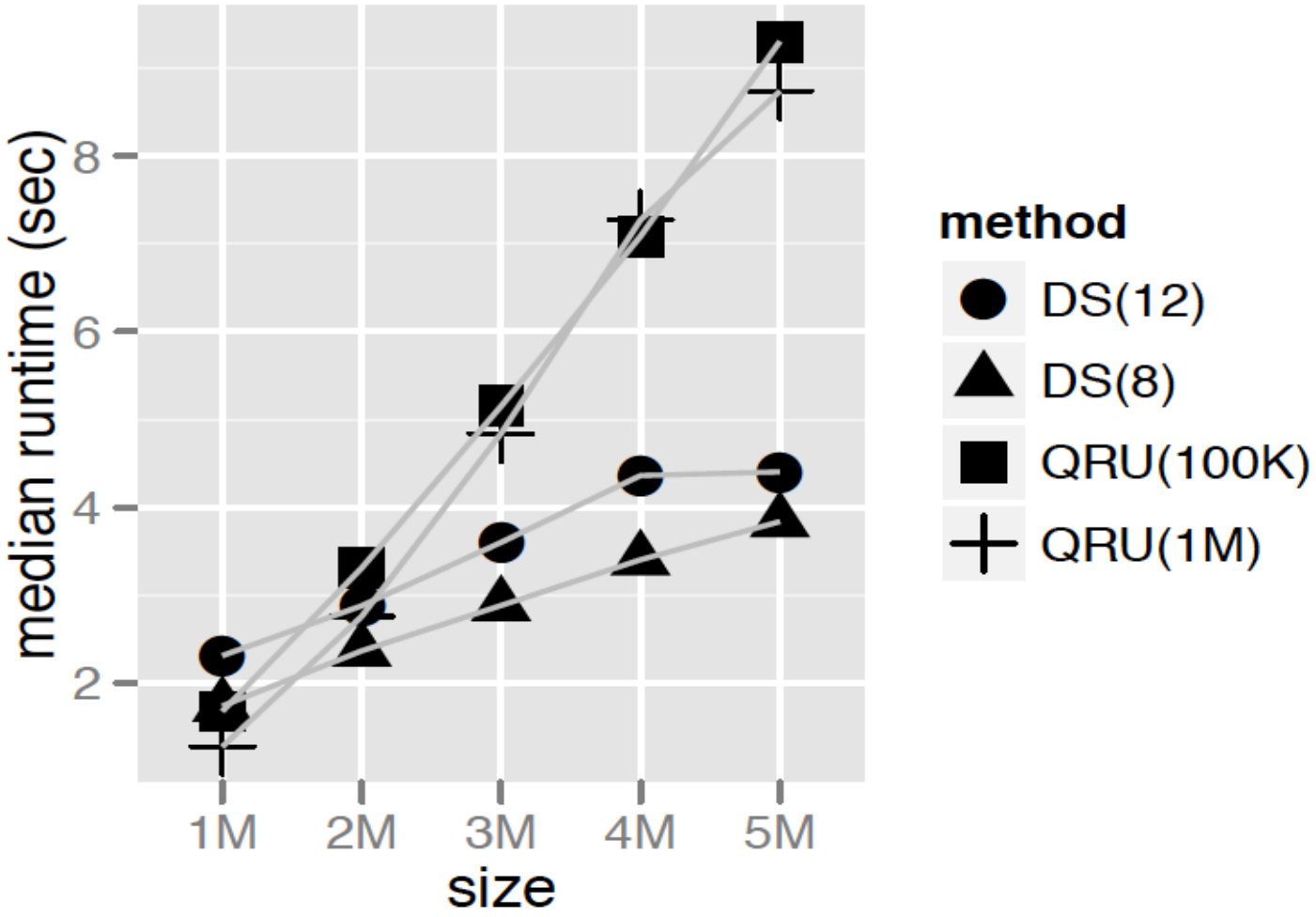
Comments

- The task may be broken up arbitrarily
  - RAM consumption is controllable
  - Dispatch of tasks to slaves
- The outputs may be combined ad libitum, when available
- Managerial tasks may be programmed (BatchJobs package and extensions…)
- Repurposing to other genomic assays is straightforward (dsQTL, meQTL, …)

Multiply agnostic, multiply scalable

- Agnosticism in a statistical algorithm's implementation: Don't know or care about
  - **data source**, except that a data chunk can be acquired on request
  - **data format**, except that certain arithmetic operations on the data tokens are well-defined
  - **execution environment**
- Never command more RAM than is needed to handle a chunk
- Permit as much asynchronous/independent computation as possible

Small proof of concept (5 runs per point)
DS = ‘doubly scalable Fisher scoring’,
QRU is updated QR in biglm



Choice of numerical constituents makes a difference: RcppEigen

Method	Relative	Elapsed	User	Sys
LDLt	1.00	1.18	1.17	0.00
LLt	1.01	1.19	1.17	0.00
SymmEig	2.76	3.25	2.70	0.52
QR	6.35	7.47	6.93	0.53
arma	6.60	7.76	25.69	4.47
PivQR	7.15	8.41	7.78	0.62
lm.fit	11.68	13.74	21.56	16.79
GESDD	12.58	14.79	44.01	10.96
SVD	44.48	52.30	51.38	0.80
GSL	150.46	176.95	210.52	149.86

Table 2: `lmBenchmark` results on a desktop computer for the default size, $100,000 \times 40$, full-rank model matrix running 20 repetitions for each method. Times (Elapsed, User and Sys) are in seconds. The BLAS in use is a locally-rebuilt version of the OpenBLAS library included with Ubuntu 11.10.

What might a MAMS implementation of `lm()` look like?

```
# step 1: sufficient quantities from data frame
  chunk
lm.suffclo = function(fmla) function(df) {
 #
 # closure that obtains X'X and X'y implied by
  #the bound fmla
 # on all the relevant data in df; FIXME:
  #dealing with NA
 #
  mm = model.matrix(fmla, df)
  mr = model.response(model.frame(fmla, df))
  list(xtx=crossprod(mm,mm), xty=crossprod
  (mm,mr))
 }
```

Step 2

- We'll use new facilities in Martin Morgan's Streamer package in Bioconductor
- The Stream class defines an ordered collection of Producer and Consumer components
- A Team class defines a scheme for executing (potentially concurrently) tasks on outputs of the Producer immediately upstream
- A yield() method on a Stream propagates yield requests to the constituents

A mortal sin, but chunk-parallelized QR not ready to hand

```
lm.suff = lm.suffclo( formula )
ssteam = Team( lm.suff, param=param ) #
  #sufficient quantities
accum = Reducer( function(x,y) list
  (sxtx=x$sxtx+y$xtx, sxty=x$sxty+y
  $xty),
     init = list(sxtx=0, sxty=0) )
ss = yield( Stream( data, ssteam,
  accum ) )
beta = solve( ss[[1]] ) %*% ss[[2]]
```

Upshots

- This addresses two aspects of MAMS: scalable data acquisition (RAM usage control) and parameterized (possibly concurrent) execution
- Agnosticism about data format could be addressed via templating (see Runnalls CXXR project for examples)
- To do: improve numerical method, exception handling

Conclusions

- Users appreciate holistic workflows, and performant versions of these are achievable for eQTL; dsQTL addressable without conceptual changes, but higher volume will compel more work on performance to allow sensitivity analysis
- Innovative data representations/volume demands should not compel duplicative algorithm rewrites, but glm has been duplicated a number of times; language design should reduce the need to do this

Conclusions

- Reflection in R has been extremely useful in the developments noted here
  - Data/outputs can be self-descriptive at high level of detail, including metadata on provenance
  - Packages are not objects but can be interrogated in detail, and can manage data decomposition
- Facilities for interrogating computing environments so that execution strategies are well-chosen (and can be programatically chosen) seem less well-developed
- These two concepts have been key to fostering sensitivity analysis in genome-scale inference