Bioconductor Workshop
Using R for Genome-Wide Analyses

Ken Rice

UW Biostatistics

Seattle, July 2009

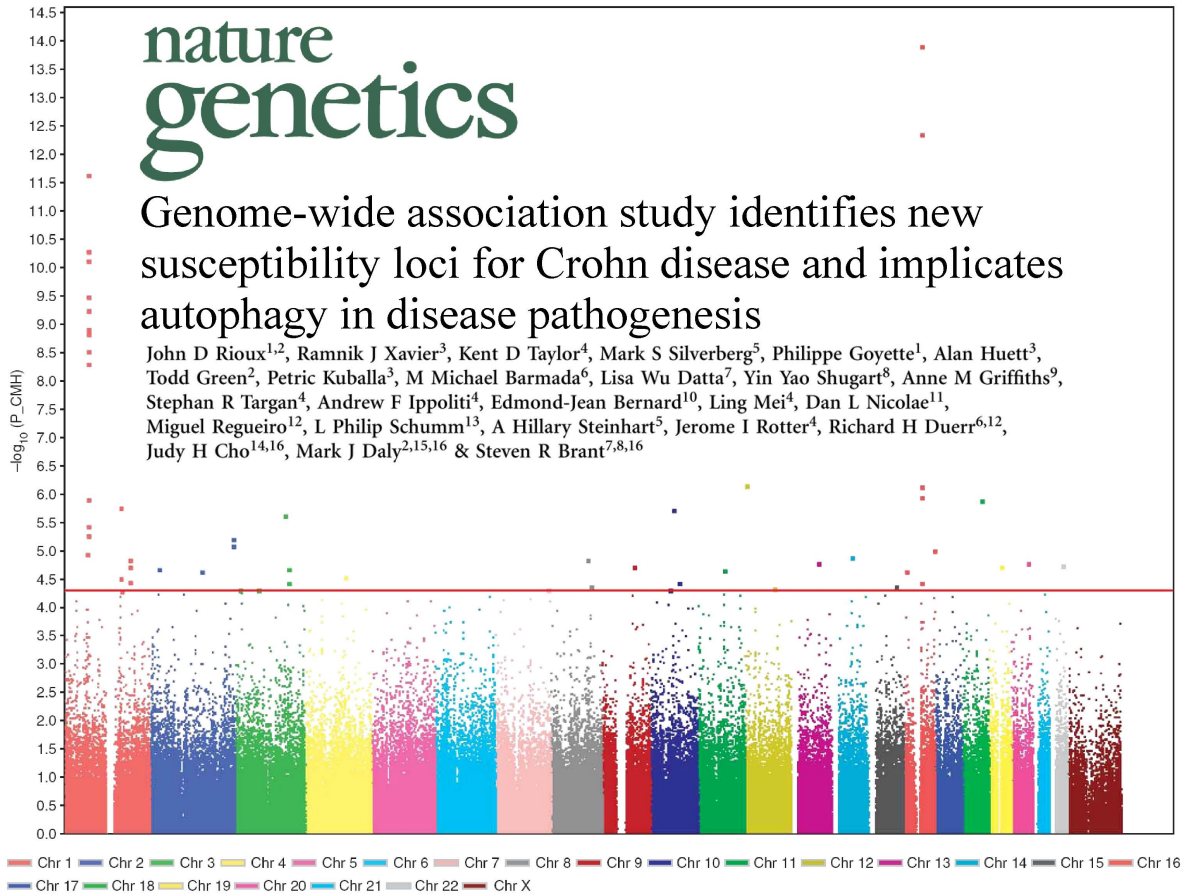
Introduction

- Assistant Prof, UW Biostat
- Currently veRy busy with Genome-Wide Studies
- Chair, Analyis Committee, for the CHARGE Consortium

My experience with R is as a (frequent) user – much of today's material is from a short course I teach with Thomas Lumley.

`http://faculty.washington.edu/kenrice/sisg`

Motivation



- Learning about diseases *via* genomics – the 'first pass' is to do millions of e.g. case-control tests
- How to do this quickly? accurately? for free?

Examples

A competitive field! ‘Findings’ are high impact...

BBC NEWS

OPEN The News in 2 minutes

News Front Page

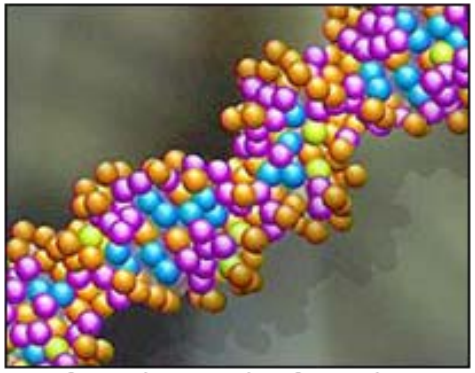
Africa
Americas
Asia-Pacific
Europe
Middle East
South Asia
UK
Business
Health
Medical notes
Science/Nature
Technology
Entertainment
Also in the news

Last Updated: Wednesday, 6 June 2007, 17:00 GMT 18:00 UK

E-mail this to a friend Printable version

Serious diseases genes revealed

A major advance in understanding the genetics behind several of the world's most common diseases has been reported.



DNA from thousands of people was analysed

The landmark Wellcome Trust study analysed DNA from the blood of 17,000 people to find genetic differences.

They found new genetic variants for depression, Crohn's disease, coronary heart disease, hypertension, rheumatoid arthritis and type 1 and 2 diabetes.

The remarkable findings, published in Nature, have been hailed as a new chapter in medical science.

Examples

A competitive field! ‘Findings’ are high impact...

BBC NEWS

OPEN The News in 2 minutes

News Front Page

Africa
Americas
Asia-Pacific
Europe
Middle East
South Asia
UK
Business
Health
Medical notes
Science/Nature
Technology
Entertainment
Also in the news

Last Updated: Thursday, 12 April 2007, 18:10 GMT 19:10 UK

E-mail this to a friend Printable version

Clear obesity gene link 'found'

Scientists say they have identified the clearest genetic link to obesity yet.

Scientists have found a clear genetic link to obesity

They found people with two copies of a "fat" version of a gene had a 70% higher risk of obesity than those with none, and weighed 3kg (6.5lb) more.

The work in Science by the Peninsula Medical School and Oxford University studied data from about 40,000 people.

The findings suggest that although improving lifestyle is key to reducing obesity, some people may find it harder to lose weight because of their genes.

Half of white Europeans carry one copy of the variant and one in six has two copies, experts estimate.

> **“The typical message has been that if you are overweight it is due to sloth and gluttony and it is your fault”**

Examples

A competitive field! ‘Findings’ are high impact...

BBC NEWS

OPEN The News in 2 minutes

News Front Page

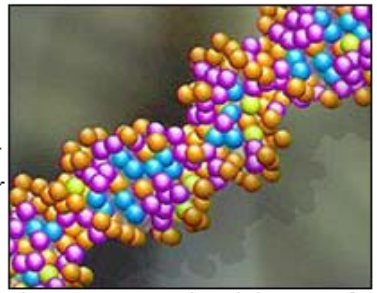
Africa
Americas
Asia-Pacific
Europe
Middle East
South Asia
UK
Business
Health
Medical notes
Science/Nature
Technology
Entertainment
Also in the news

Last Updated: Saturday, 9 June 2007, 00:13 GMT 01:13 UK

E-mail this to a friend Printable version

Scientists find new dementia gene

Scientists say they have discovered a new gene linked with late-onset Alzheimer's disease.



The scientists analysed the DNA of over 1,000 people

People with a damaged copy of the gene, GAB2, may be at four times increased risk of developing dementia, Neuron journal reports.

Experts said the latest findings were some of the most significant to emerge since the discovery of the ApoE4 Alzheimer's gene.

Late-onset Alzheimer's affects one in 10 people over 65 and half of over 85s.

The researchers, from 15 institutions including the Institute of Neurology in London, analysed the DNA of 1,411 people and found GAB2 influenced the risk of dementia among those with APOE4.

> **“The results are some of the most significant finds for genetic risk factors since the discovery of ApoE4”**
>
> Professor Clive Ballard, director research at the Alzheimer's Society

Examples

A competitive field! ‘Findings’ are high impact...

BBC NEWS

OPEN The News in 2 minutes

News Front Page

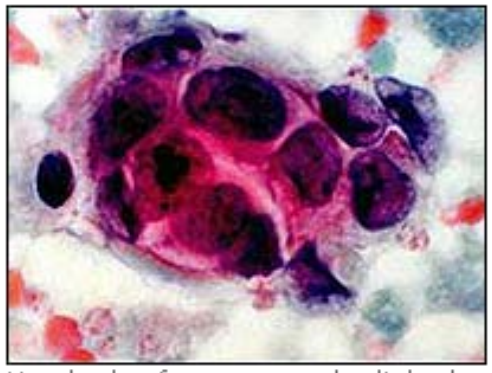
Africa
Americas
Asia-Pacific
Europe
Middle East
South Asia
UK
Business
Health
Medical notes
Science/Nature
Technology
Entertainment
Also in the news

Last Updated: Monday, 28 May 2007, 05:57 GMT 06:57 UK

E-mail this to a friend Printable version

New breast cancer genes discovery

Scientists have developed a new technique to identify genes that increase the chance of women developing breast cancer.



Hundreds of genes may be linked to breast cancer

They hope it will lead to a single blood test which would reveal a woman's risk of getting the disease.

Researchers say the new technique speeds up gene identification and could mean finding all the genes associated with breast cancer.

Cancer Research UK described the development as "hugely significant".

Examples

Still a competitive area...

NEWS

Watch ONE-MINUTE WORLD NEWS

News Front Page

Africa
Americas
Asia-Pacific
Europe
Middle East
South Asia
UK
Business
Health
Medical notes
Science & Environment
Technology
Entertainment
Also in the news

Page last updated at 23:02 GMT, Sunday, 17 May 2009 00:02 UK

E-mail this to a friend Printable version

Women's menstruation genes found

Scientists say they have begun to crack the genetic code that helps determine when a girl becomes a woman.

The genes were found on chromosomes six and nine

A UK-led team located two genes on chromosomes six and nine that appear to strongly influence the age at which menstruation starts.

The Nature Genetics study also provides a clue for why girls who are shorter and fatter tend to get their periods months earlier than classmates.

The genes sit right next to DNA controlling height and weight.

Examples

Still a competitive area...

Hindustan Times

Fri,29 May 2009 hindustantimes.com

Zany Science

NEWS | ELECTIONS '09 new! | VIEWS | BUSINESS | CRICKET | CINEMA | LIFESTYLE | TABLOID | PHOTOS | VIDEO | E

Smart Zone | Kids Zone | Window Seat | Education | Teens | **Zany Science**

Home ↳ HTNext ↳ Zany Science ↳ Story

BP treatments to get better

AIII

London, May 11, 2009

First Published: 13:45 IST(11/5/2009)
Last Updated: 14:10 IST(11/5/2009)

Scientists from Massachusetts General Hospital claim that they have identified eight genetic variants associated with hypertension.

The research team, as a part of Global Blood Pressure Genetics (Global BPgen) study group, analysed the genome of 130,000 individuals from around the world.

Examples

Still a competitive area...

NEWS

Watch ONE-MINUTE WORLD NEWS

News Front Page

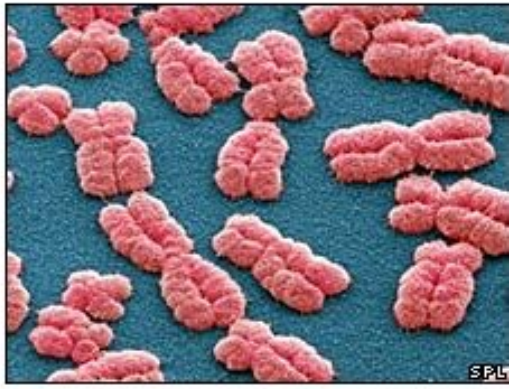
Africa
Americas
Asia-Pacific
Europe
Middle East
South Asia
UK
Business
Health
Medical notes
Science & Environment
Technology
Entertainment

Page last updated at 09:07 GMT, Monday, 15 September 2008 10:07 UK

E-mail this to a friend Printable version

Gene tests 'create undue stress'

Gene tests to predict a person's future risk of life-threatening disease may be damaging to health by causing unnecessary stress, an expert claims.



Chromosomes house our DNA

Professor Nilesh Samani, British Heart Foundation chair of cardiology, says the tests are too inaccurate to help the individual.

Someone deemed high risk for a disease based on their gene test may never go on to develop the condition.

Data Cleaning

Before analysis gets started, the gigabytes of data we have must be 'cleaned'

- Mismatches discovered (Sex, Ancestry)
- Family structure discovered (e.g. Sibs, 'Kinship Coefficient')
- Dumping SNPs with 'high' missing rates (e.g. $\leq$ 99% complete)

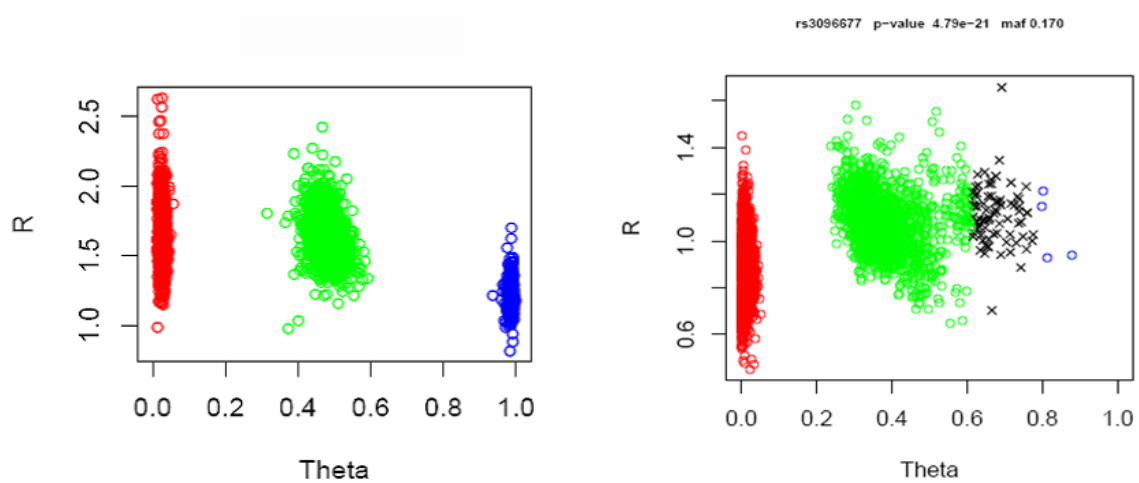
As we require $p < 10^{\text{exciting}}$ in tests, even minor flaws cause headaches, by the 1000. (But we have e.g. 2.5 million tests to do)

Most of the cleaning is straightforward; compute, say the MLE for kinship. But, done carelessly, it can be **slow**.

Data Cleaning: HWE test

Does your SNP data look like this?

Genotype	AA	Aa	aa
Proportion	$(1-p)^2$	$2p(1-p)$	p^2



Yes! Not so much

- We don't *believe* Hardy-Weinberg holds exactly
- But it's v v unlikely we are *miles* from HWE. The HWE test is good at spotting mis-calls, in ancestry-specific groups
- The approximate test is okay. The exact test is preferred...

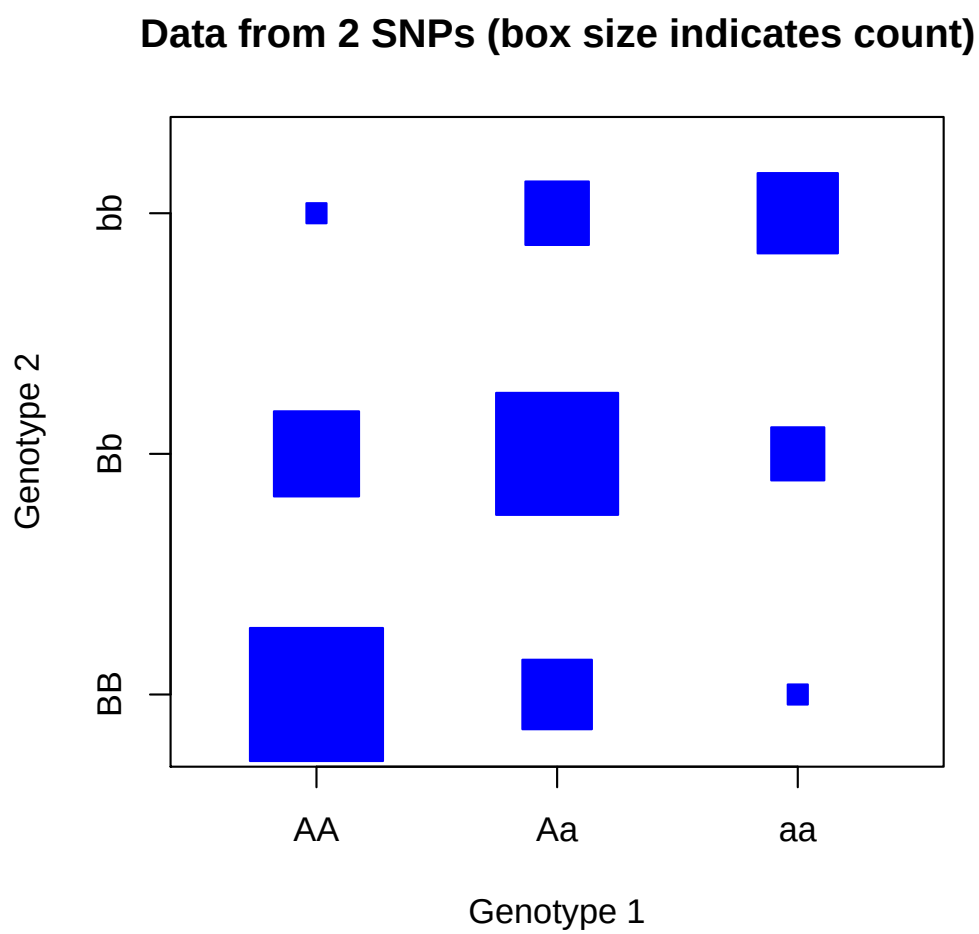
Data Cleaning: HWE test

The `hwde` package has the `hwexact()` function. This is okay (and we use it, basically) but will be slow with large datasets. It uses (smart) ennumeration of all the possible datasets for n subjects. It can be improved by

- Stopping calculating when you're sure that e.g. $p > 0.1$. As we're doing something like 10^6 tests, $p \geq 10^{-4}$ (or so) are not worth getting out of bed for – although you'll have to truncate plots, etc.
- If you're sure of n, construct a lookup table, and use that.
- Doing the (quick) approximate test, and only looking at $\tilde{p} \leq 0.1$ for the full works.
- Coding the hard stuff in `C`, not `R`

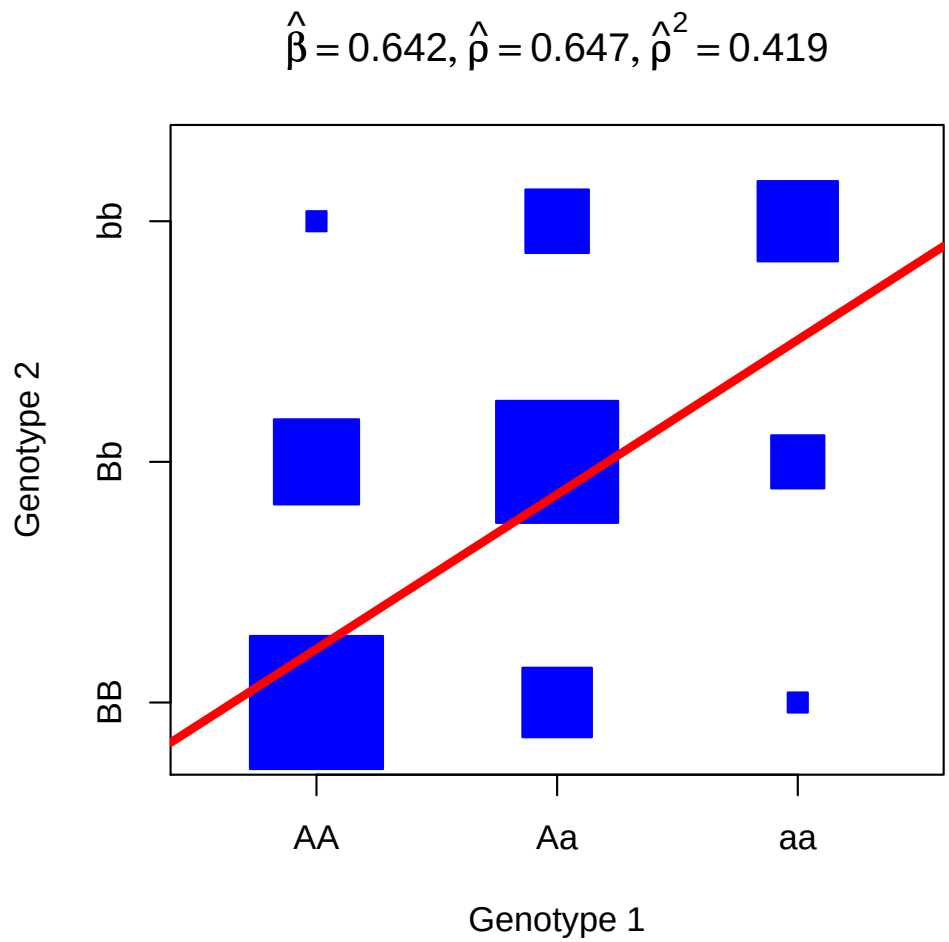
Data Cleaning: r^2 for all SNPs

A brief reminder/introduction:



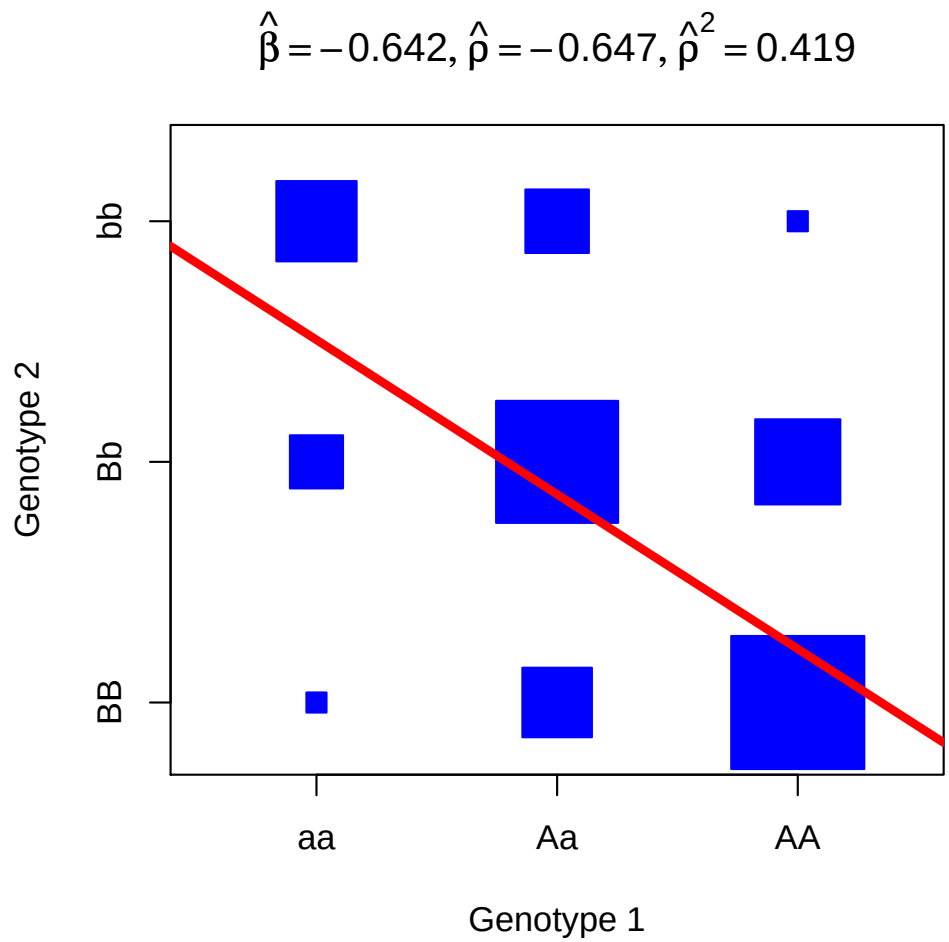
Data Cleaning: r^2 for all SNPs

A brief reminder/introduction:



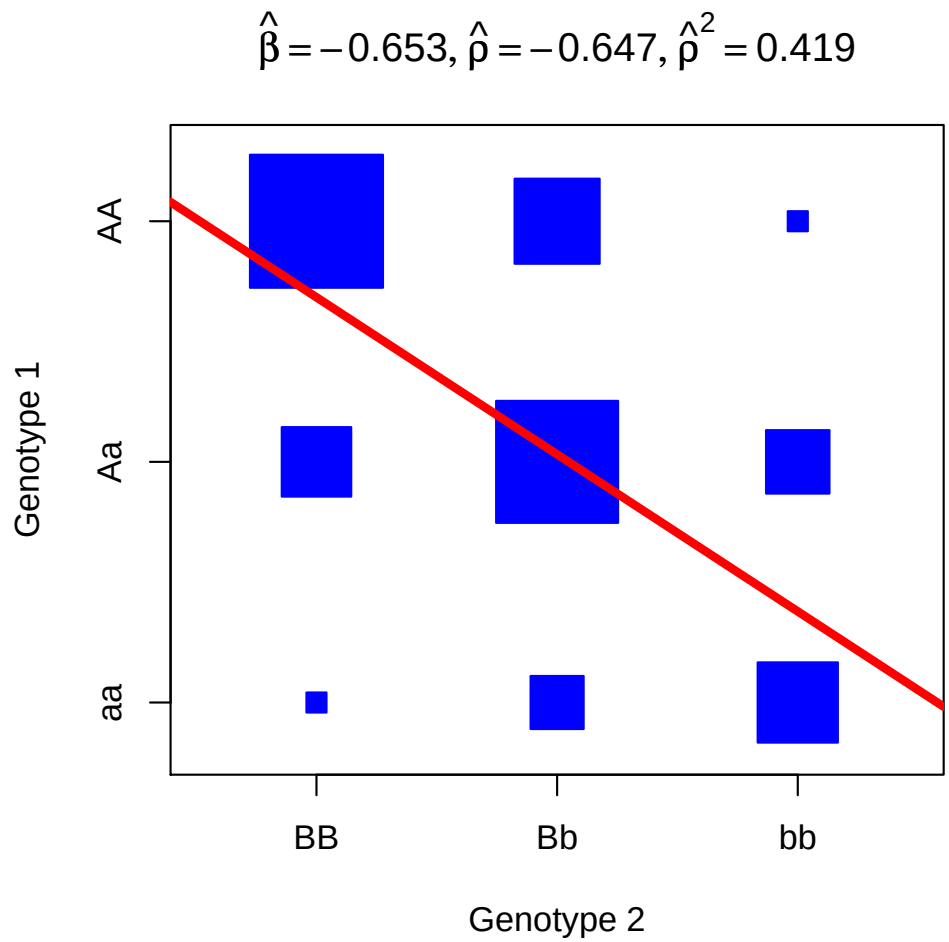
Data Cleaning: r^2 for all SNPs

A brief reminder/introduction:



Data Cleaning: r^2 for all SNPs

A brief reminder/introduction:



Data Cleaning: r^2 for all SNPs

We see that;

- $\widehat{\beta} = \frac{\text{Cov}(G_1,G_2)}{\text{Var}(G_1)}$ but $\rho = \frac{\text{Cov}(G_1,G_2)}{\sqrt{\text{Var}(G_1)\text{Var}(G_2)}}$ ($\widehat{\rho}$, formally)
- $r^2 = \rho^2$ doesn't care about a/A or b/B designation – but **you** probably do
- ρ (and ρ^2) doesn't care about 0/1/2 vs 1/2/3 – but often '0'≡missing, so be careful
- ρ^2 doesn't care if you switch the G_1, G_2 labels

We'd like to check our r^2 match the HapMap (roughly)

Given documentation, computing r^2 for 2 SNPs' data should not be hard. Computing it for many SNPs probably doesn't *look* hard, if you have `R` experience.

Data Cleaning: r^2 for all SNPs

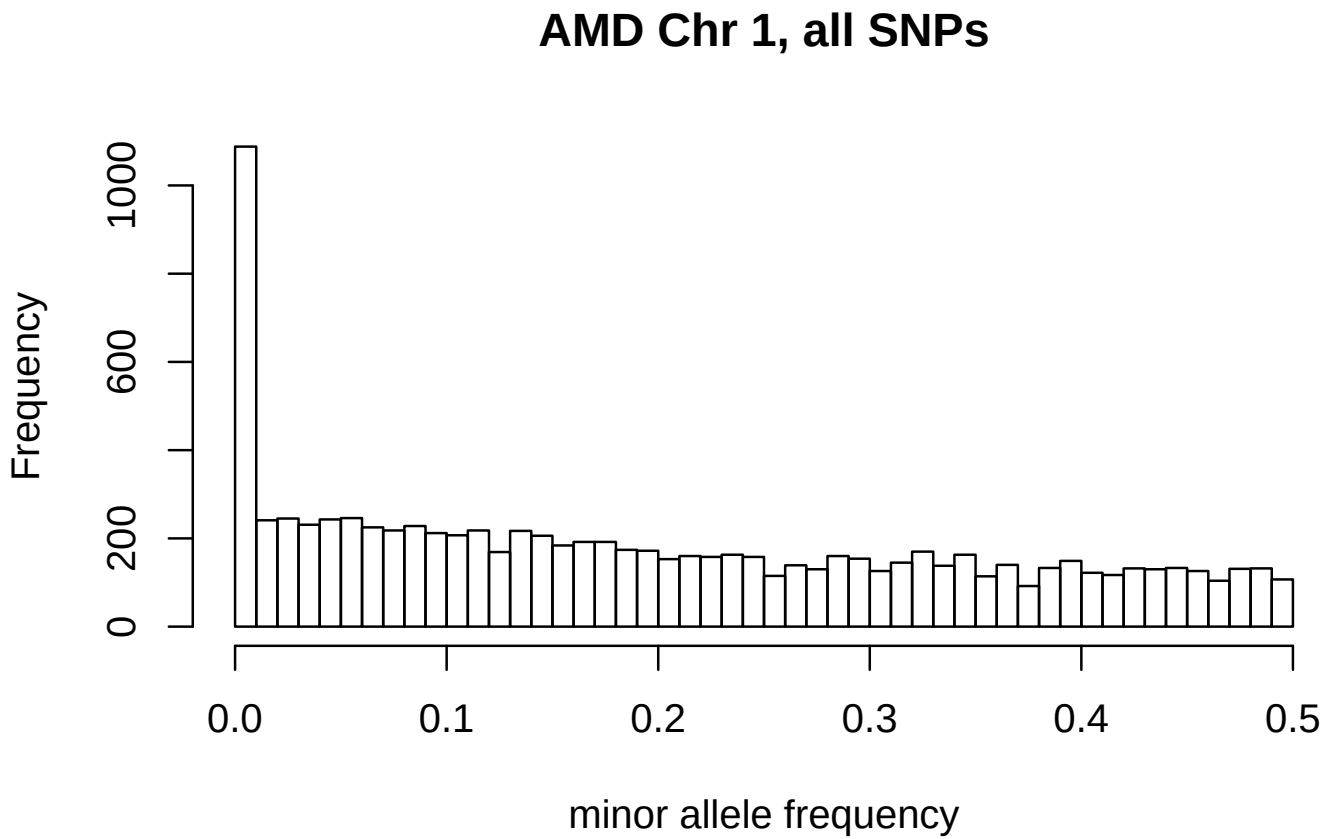
For some example data, consider LD of 9000 Chr 1 SNPs in the AMD dataset (see the site). $\binom{9202}{2} = 42.3$ million pairs (eek!). There are numerous **very bad** ways to do this job!

The challenges are;

1. To do calculations **quickly** (hard)

2. **Not to bother** with unnecessary ones (easier) – we'll drop all SNPs with minor allele frequency ≤ 0.05

Data Cleaning: r^2 for all SNPs



This filters out 2048 SNPs, leaving 7154. $\binom{7154}{2}$=25.6M

Data Cleaning: r^2 for all SNPs

We'll go through some 'traditional' improvements to code; here's a first attempt;

```
r2.out <- matrix(NA, 7154, 7154)

for( i in 1:7154 ){
   for( j in 1:7154 ){
      r2.out[i,j] <- cor(amd[i,], amd[j,])^2
}}
```

... clearly we can be smarter than this.

Data Cleaning: r^2 for all SNPs

Recall that r^2 didn't care if we 'switched the axes' $\Rightarrow$ only compute r^2_{ij} if $i > j$

```
for( i in 1:7154 ){
   for( j in i:7154 ){
      r2.out[i,j] <- cor(amd[i,], amd[j,])^2
}}
```

This saves a factor of two

Data Cleaning: r^2 for all SNPs

'Note' that every SNP has $r^2 = 1$ with itself

$\Rightarrow$ don't compute r^2_{ij} if $i = j$

```
for( i in 1:(7154-1) ){
   for( j in (i+1):7154 ){
      r2.out[i,j] <- cor(amd[i,], amd[j,])^2
}}
```

This is a very minor saving

Data Cleaning: r^2 for all SNPs

At the moment, our code doesn't do anything special with NAs;

```
> cor( c(1,3,5,NA), c(-2,5,0,6) )
[1] NA
```

'Default' use of `cor()` would be a bit wasteful. There are only 6432 AMD SNPs with complete data, and the rest typically have only a few NAs

- $\Rightarrow$ we *can* get some useful estimate of r^2 from the subjects with data from SNP `i` **and** `j`
- ... afterwards, need to watch out for 'weirdness' due to this decision

Data Cleaning: r^2 for all SNPs

`cor()` can do the complete-cases analysis, if we supply option `use="complete.obs"`. (See the help file for details; if **all** missing this gives an error)

```
for( i in 1:(7154-1) ){
 for( j in (i+1):7154 ){
   r2.out[i,j] <- cor(amd[i,], amd[j,], use="complete.obs")^2
}}
```

For more general GWAS work, learn how to use `tryCatch()` – Murphy's Law applies. Also e.g. `system.time()`

Data Cleaning: r^2 for all SNPs

Let's try the code. For an estimate of runtime;

```
system.time({
for( i in 1:(1000-1) ){
   for( j in (i+1):1000 ){
      r2.out[i,j] <- cor(amd[i,], amd[j,], use="complete.obs")
}}
})
```

This does $\binom{1000}{2}$=0.5M pairs, and takes $\sim$ 3 minutes.

Data Cleaning: r^2 for all SNPs

The full works; (took 2.5 hours on my desktop)

```
for( i in 1:(7154-1) ){
   for( j in (i+1):7154 ){
      r2.out[i,j] <- cor(amd[i,], amd[j,], use="complete.obs")
}}
Warning messages:
1: In cor(amd[i, ], amd[j, ], use = "complete.obs") :
  the standard deviation is zero
```

Ooops. This is worrying; is it fatal?

Data Cleaning: r^2 for all SNPs

... is it fatal?

No – it's only a warning. Supplying `cor()` with data where e.g. $G_1 = aa$ for everyone leads to this warning, and `NA` as the output (see the documentation)

- `NA` as output **does** make sense here
- Defaults options are sensible, so don't panic too soon
- Recall we filtered MAF<0.05. The weirdness could happen when the missingness in G_2 leads to effective MAF=0 for G_1.
- *Perhaps* all genotypes=Aa (HWE filters would catch this)
- Catching all potential errors is *really* hard – really robust code is required

Data Cleaning: r^2 for all SNPs

2.5 hours (optimized!) is pretty rubbish. How to do massively better?

- The `cor()` function calls `C`. **If you feed it a matrix**, it calls `C` to give you the correlations of all pairs of columns

- This gets all the data (and `for()` 'administration') into `C`, not `R` (and is therefore faster)

- Doing this in 10^{-5} seconds not 10^{-3} is beneficial – multiply by 10^6 to see this!

Data Cleaning: r^2 for all SNPs

```
r2.matrix.quick <- cor( t(amd), use="pairwise.complete.obs" )^2
```

- 2 minutes on my desktop (!)
- The admin/data reading **was** the bottleneck – and we optimized it
- This holds much more generally in GWAS (where 'vectorized' `C` code is not available for every job)
- Caveats about `NA`s and 'weirdness' still apply
- With more SNPs/people, may need to split Chromosomes into chunks, to get everything in memory

(In a class of genetics-oriented students, none of them spotted this trick. It *is* in the help files, but isn't obvious. In non-GWAS work I'd never mention it to them)

Data Cleaning: r^2 for all SNPs

To finish off, it would be nice to have a plot of r^2 versus inter-SNP distance (`pos[j]-pos[i]` in AMD)

A couple of ideas to help this along;

- Produce the plot in PNG format – with the `png()` command. A PDF would be nice, but would have to keep track of 25.6M points, making it a massive file.

- Add points to the plot in groups. Making a new vector of 25.6M inter-SNP distances needlessly uses up a huge amount of memory in your `R` session

Data Cleaning: r^2 for all SNPs

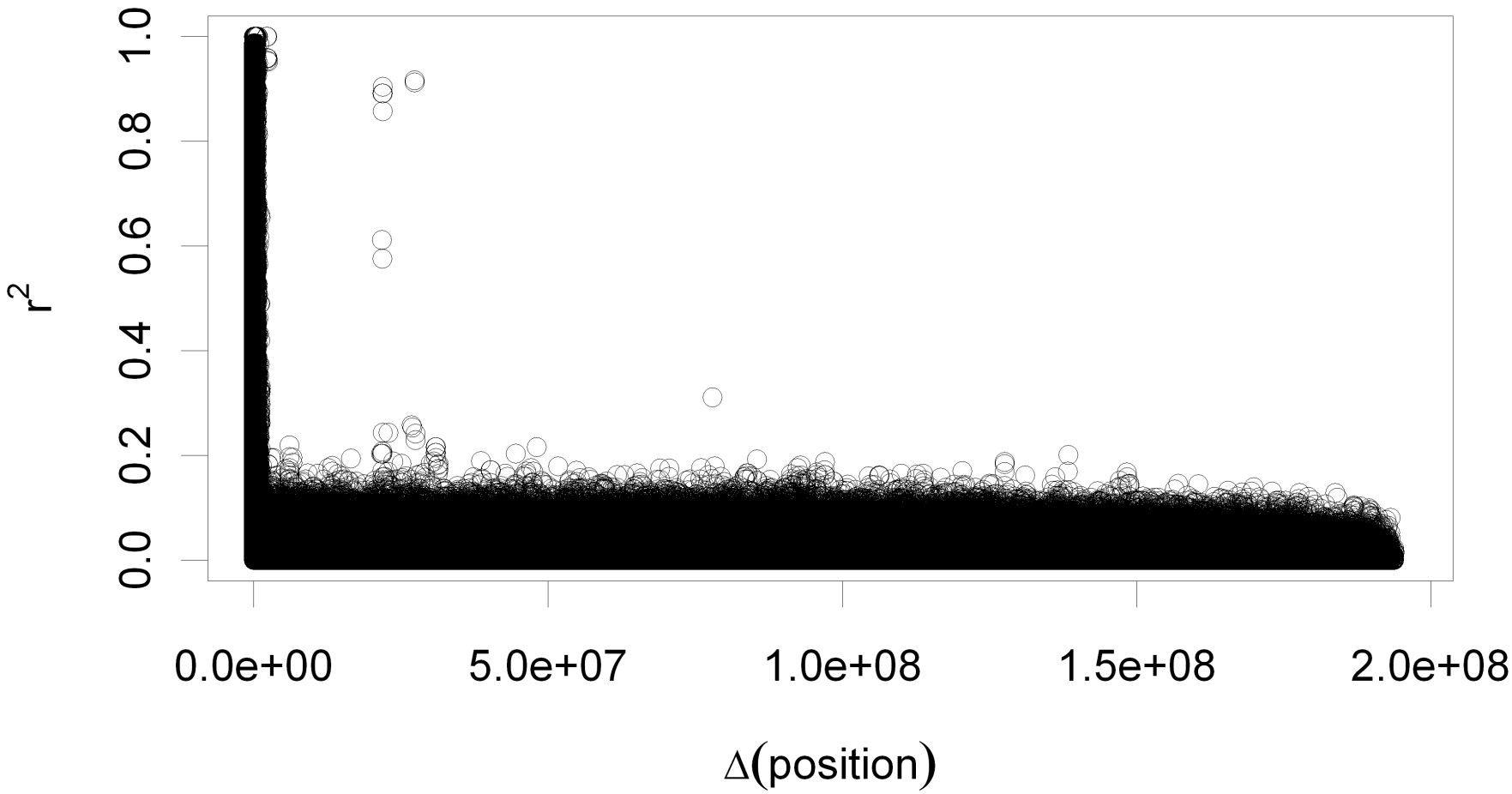
```
png("r2plot.png", w=6*600, h=4*600, pointsize=12*600/72)
#set up the plot, with fancy axis labels;
plot(0, type="n", xlim=c(0,2.5E8), ylim=c(0,1),
  xlab=expression(Delta(plain(position))), ylab=expression(r^2) )

#add the points, one SNP at a time;
for(i in 1:(7154-1)){
  points( amd$pos[(i+1):7154]-amd$pos[i], r2.out[i,(i+1):7154] )
}
dev.off()
```

The output is clunky-but-okay;

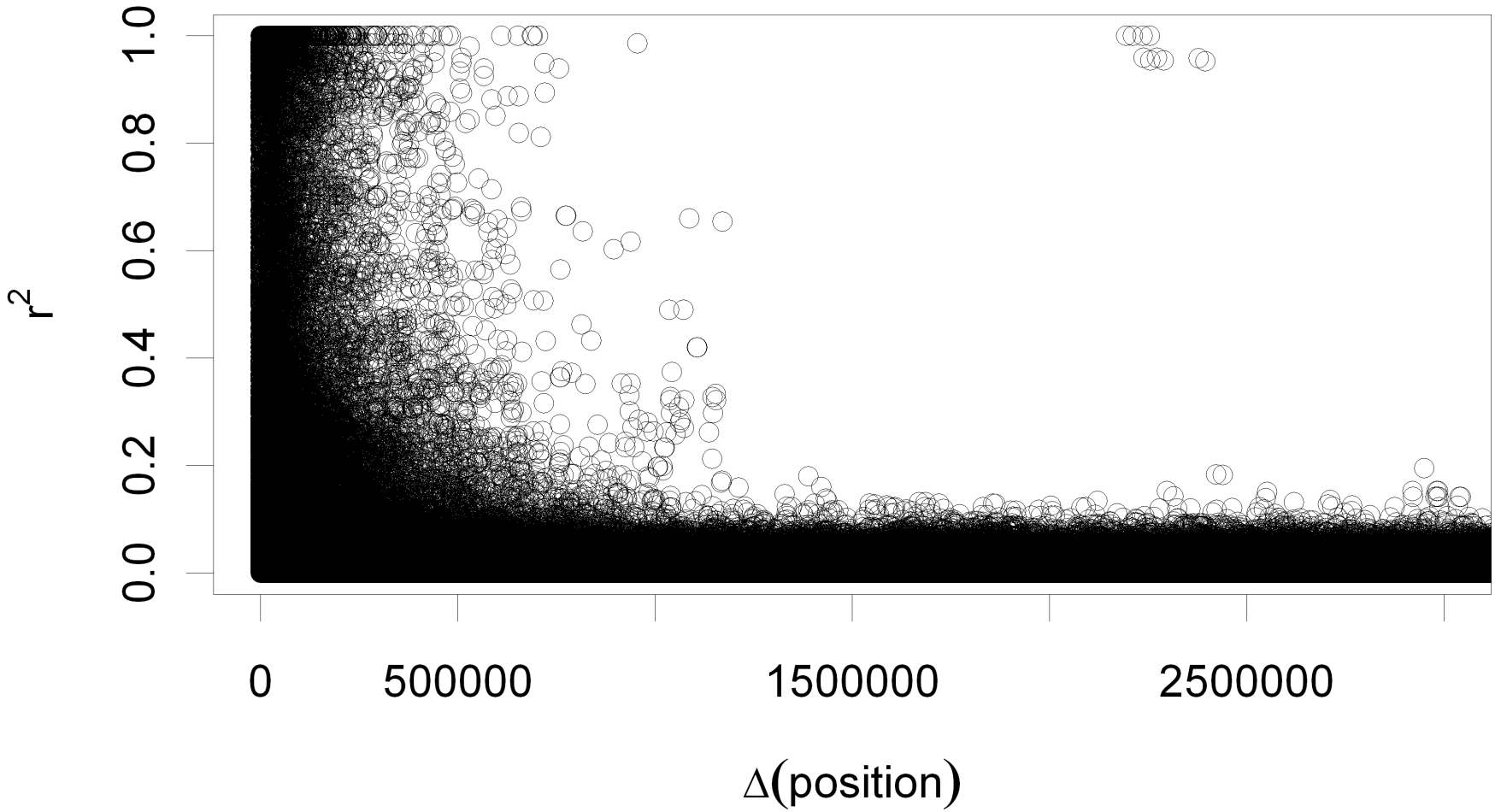
Data Cleaning: r^2 for all SNPs

Plotting r^2 against inter-SNP distance;



Data Cleaning: r^2 for all SNPs

Plotting r^2 against inter-SNP distance; (zoom)



Large data

"R is well known to be unable to handle large data sets."

Solutions:

- Get a bigger computer: Linux computer with 16Gb memory for $< \$2500$
- Don't load all the data at once (methods from the mainframe days).

Large data: storage formats

R has two convenient data formats for large data sets

- For ordinary large data sets, the `RSQLite` package provides storage using the SQLite relational database.

- For very large 'array-structured' data sets such as whole-genome SNP chips, the `ncdf` package provides storage using the netCDF data format.

Large data: netCDF

netCDF was designed by the NSF-funded UCAR consortium, who also manage the National Center for Atmospheric Research.

Atmospheric data are often array-oriented: eg temperature, humidity, wind speed on a regular grid of (x, y, z, t).

Need to be able to select 'rectangles' of data – eg range of (x, y, z) on a particular day t.

Because the data are on a regular grid, the software can work out where to look on disk without reading the whole file: efficient data access.

Large data: how big are GWAS?

Array oriented data (position on genome, sample number) for genotypes, probe intensities.

Potentially very large data sets:

2,000 people $\times$ 300,000 = tens of Gb

16,000 people $\times$ 1,000,000 SNPs = hundreds of Gb.

Even worse after imputation to 2,500,000 SNPs.

R can't handle a matrix with more than $2^{31} - 1 \approx 2$ billion entries even if your computer has memory for it. Even data for one chromosome may be too big.

Large data: using netCDF

With the `ncdf` package:

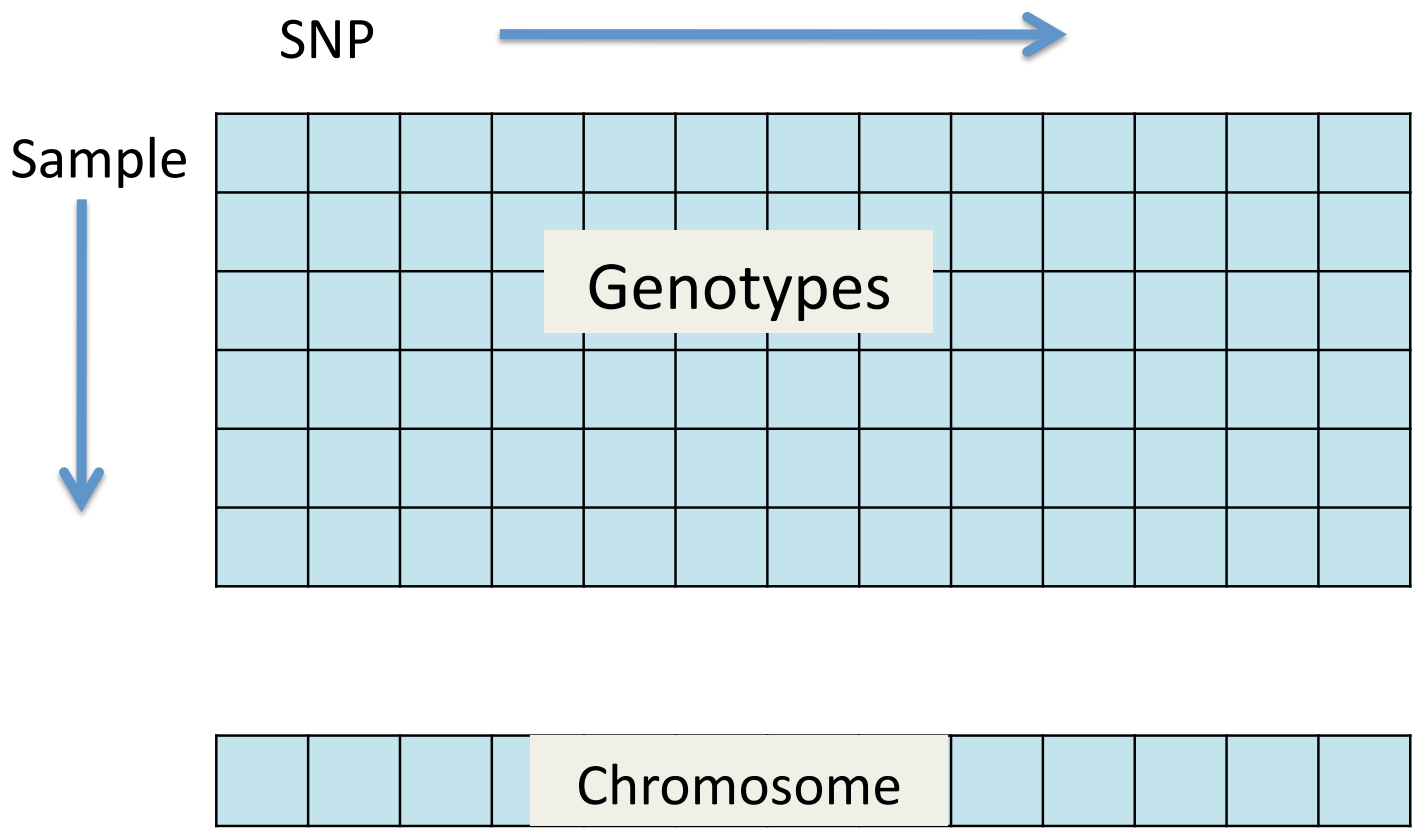
`open.ncdf()` opens a netCDF file and returns a connection to the file (rather than loading the data)

`get.var.ncdf()` retrieves all or part of a variable.

`close.ncdf()` closes the connection to the file.

Large data: using netCDF

Variables can use one or more array dimensions of a file



Large data: example

Finding long homozygous runs (possible deletions)

```
library("ncdf")
nc <- open.ncdf("hapmap.nc")

## read all of chromosome variable
chromosome <- get.var.ncdf(nc, "chr", start=1, count=-1)
## set up list for results
runs<-vector("list", nsamples)

for(i in 1:nsamples}{
   ## read all genotypes for one person
   genotypes <- get.var.ncdf(nc, "geno", start=c(1,i),count=c(-1,1))
   ## zero for htzygous, chrm number for hmzygous
   hmzygous <- genotypes != 1
   hmzygous <- as.vector(hmzygous*chromosome)
```

Large data: example

```
  ## consecutive runs of same value
  r <- rle(hmzygous)
  begin <- cumsum(r$lengths)
  end <- cumsum(c(1, r$lengths))
  long <- which ( r$lengths > 250 & r$values !=0)
  runs[[i]] <- cbind(begin[long], end[long], r$lengths[long])
}

close.ncdf(nc)
```

Notes

- `chr` uses only the 'SNP' dimension, so `start` and `count` are single numbers
- `geno` uses both SNP and sample dimensions, so `start` and `count` have two entries.
- `rle` compresses runs of the same value to a single entry.

Large data: making netCDF files

Creating files is more complicated

- Define dimensions
- Define variables and specify which dimensions they use
- Create an empty file
- Write data to the file.

Large data: netCDF 'dimensions'

Specify the name of the dimension, the units, and the allowed values in the `dim.def.ncdf` function.

One dimension can be 'unlimited', allowing expansion of the file in the future. An unlimited dimension is important, otherwise the maximum variable size is 2Gb.

```
snpdim<-dim.def.ncdf("position","bases", positions)
sampledim<-dim.def.ncdf("seqnum","count",1:10, unlim=TRUE)
```

Large data: netCDF ‘variables’

Variables are defined by name, units, and dimensions

```
varChrm <- var.def.ncdf("chr","count",dim=snpdim,
         missval=-1, prec="byte")
varSNP <- var.def.ncdf("SNP","rs",dim=snpdim,
        missval=-1, prec="integer")
vargeno <- var.def.ncdf("geno","base",dim=list(snpdim, sampledim),
        missval=-1, prec="byte")
vartheta <- var.def.ncdf("theta","deg",dim=list(snpdim, sampledim),
        missval=-1, prec="double")
varr <- var.def.ncdf("r","copies",dim=list(snpdim, sampledim),
       missval=-1, prec="double")
```

Large data: creating files

The file is created by specifying the file name ad a list of variables.

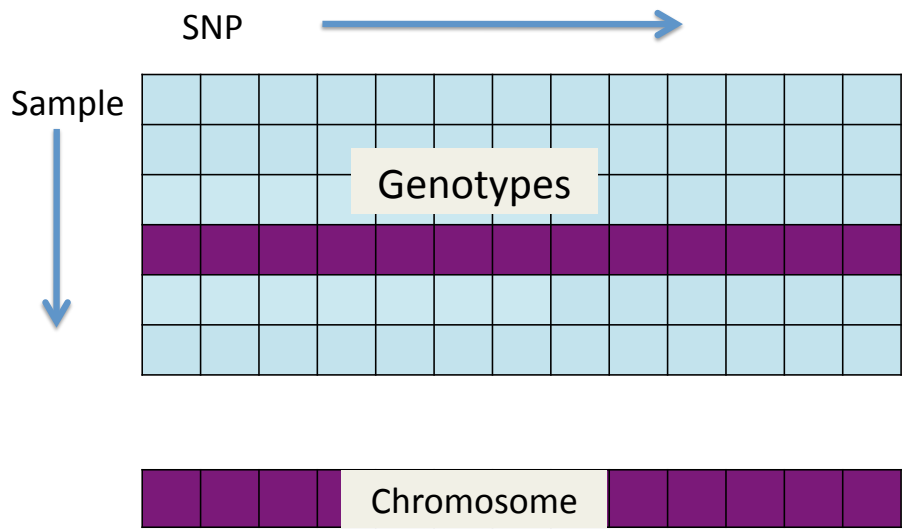
```
genofile<-create.ncdf("hapmap.nc", list(varChrm, varSNP, vargeno,
                 vartheta, varr))
```

The file is empty when it is created. Data can be written using `put.var.ncdf()`. Because the whole data set is too large to read, we might read raw data and save to netCDF for one person at a time.

```
for(i in 1:4000){
    geno<-readRawData(i) ## somehow
    put.var.ncdf(genofile, "geno", genc,
           start=c(1,i), count=c(-1,1))
}
```

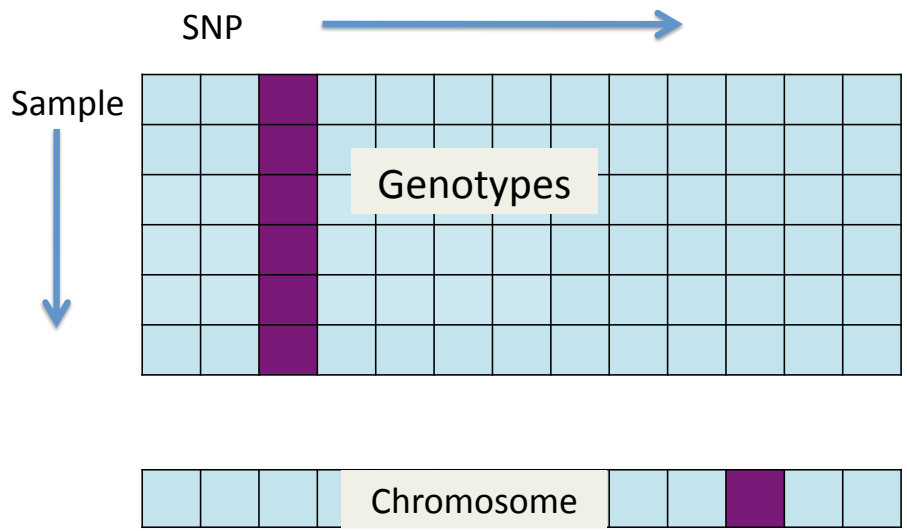
Large data: using netCDF efficiently

Read all SNPs, one sample



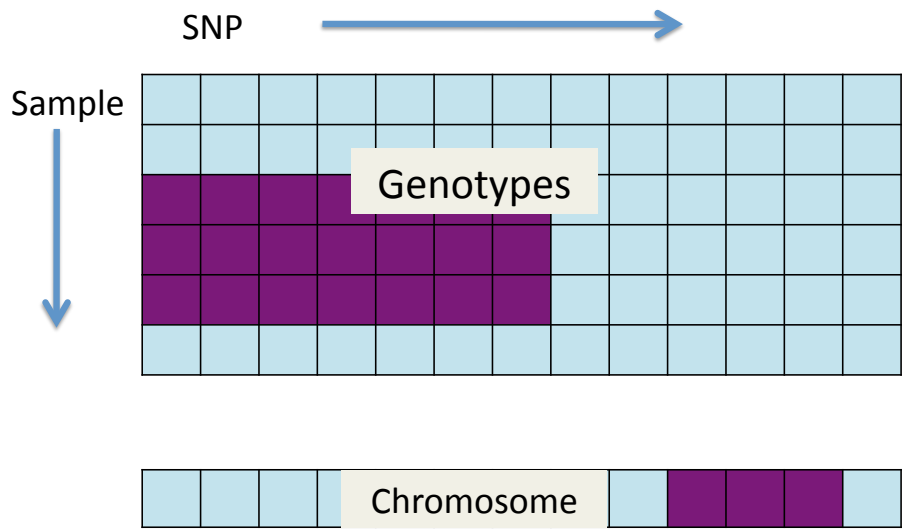
Large data: using netCDF efficiently

Read all samples, one SNP



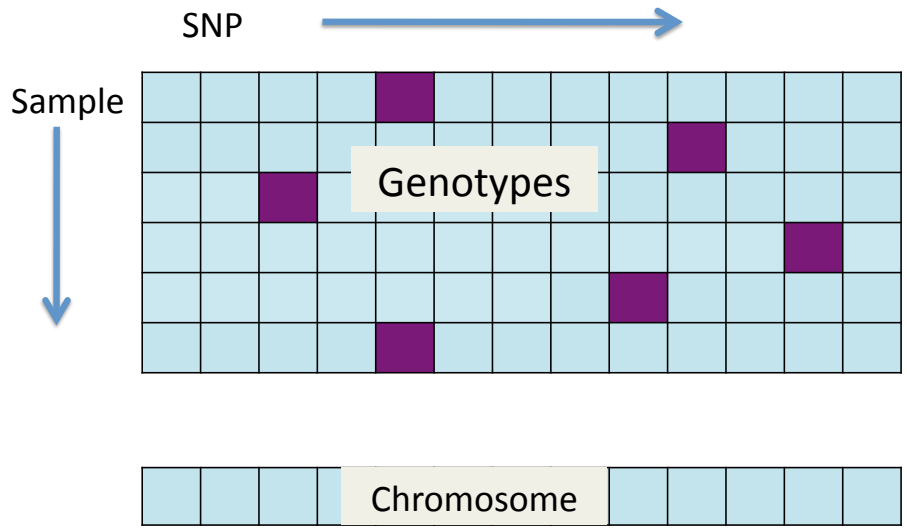
Large data: using netCDF efficiently

Read some samples, some SNPs.



Large data: using netCDF efficiently

Random access is not efficient: eg read probe intensities for all missing genotype calls.



Large data: using netCDF efficiently

- Association testing: read all data for one SNP at a time
- Computing linkage disequilibrium near a SNP: read all data for a contiguous range of SNPs
- QC for aneuploidy: read all data for one individual at a time (and parents or offspring if relevant)
- Population structure and relatedness: read all SNPs for two individuals at a time.

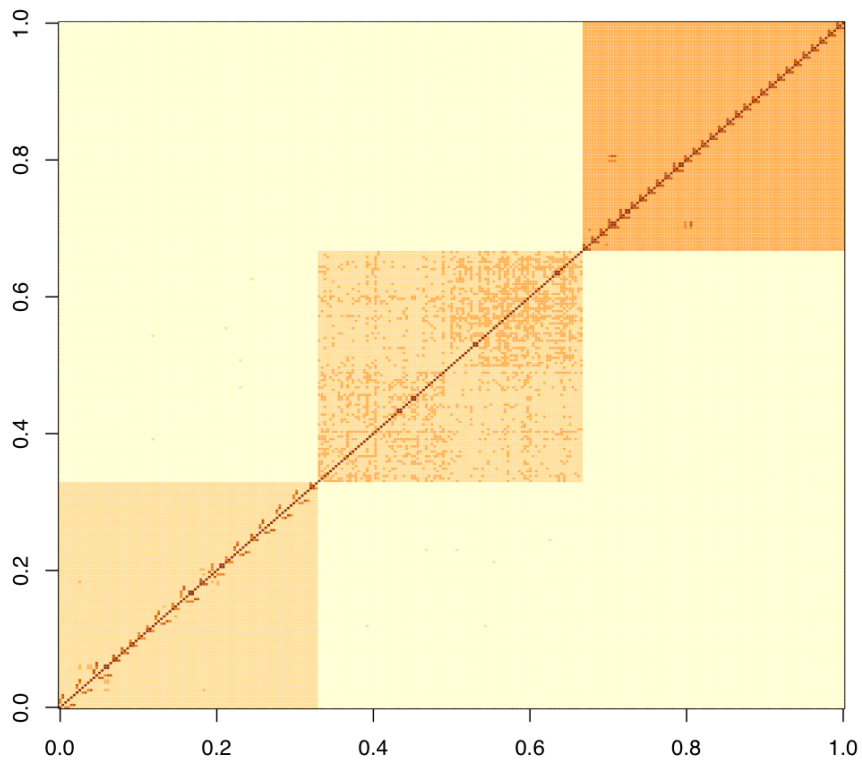
Large data: using netCDF efficiently

Another example; computing IBS for pairs of a `hapmap` dataset (some setup skipped)

```
p<-proc.time()
for(i in 2:nsamples){
  genoi<-get.var.ncdf(hapmap,"genotype",
                      start=c(1,i),count=c(nsnps,1))[autosomes]
  goodi<-genoi>=0
  xymat[i,i]<-sum(genoi[goodi]^2)
  counts[i]<-sum(genoi[goodi])
  ibs[i,i]<-2
  missed[i]<-nauto-sum(goodi)
  for(j in 1:i){
      genoj<-get.var.ncdf(hapmap,"genotype",start=c(1,j),count=c(nsnps,1))[autosom
      goodj<-genoj>=0
      good<-goodi & goodj
      xymat[i,j]<-sum(genoi[good]*genoj[good])
      ibs[i,j]<-sum( (genoi[good]==genoj[good])*2+(genoi[good]==1))/sum(good)
      xymat[j,i]<-xymat[i,j]
      ibs[j,i]<-ibs[i,j]
  }
 if(!(i%%10)) print(c(i,proc.time()-p))
 p<-proc.time()}
```

Large data: using netCDF efficiently

Plotting the results; (for HapMap – use `C` for huge studies)



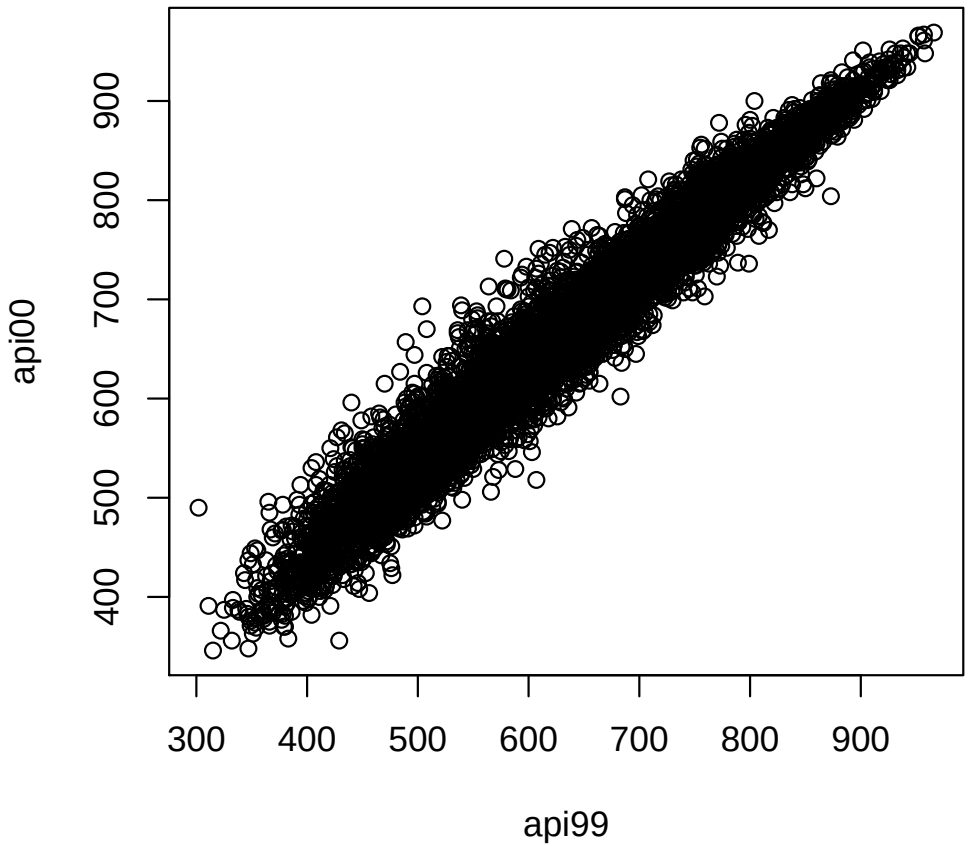
Bioconductor favorites: hexbin

GWAS (and genetics/genomics in general) tends to produce **massive** datasets. On any (standard) plot of e.g. 10,000 points, **many** will overlap

A simple example is the California Academic Performance Index reported from 6194 schools (in the `survey` package)

```
> install.packages("survey")
> library(survey)
> data(api)
> plot(api00~api99,data=apipop) # plain plot
```

Bioconductor favorites: hexbin



Bioconductor favorites: hexbin

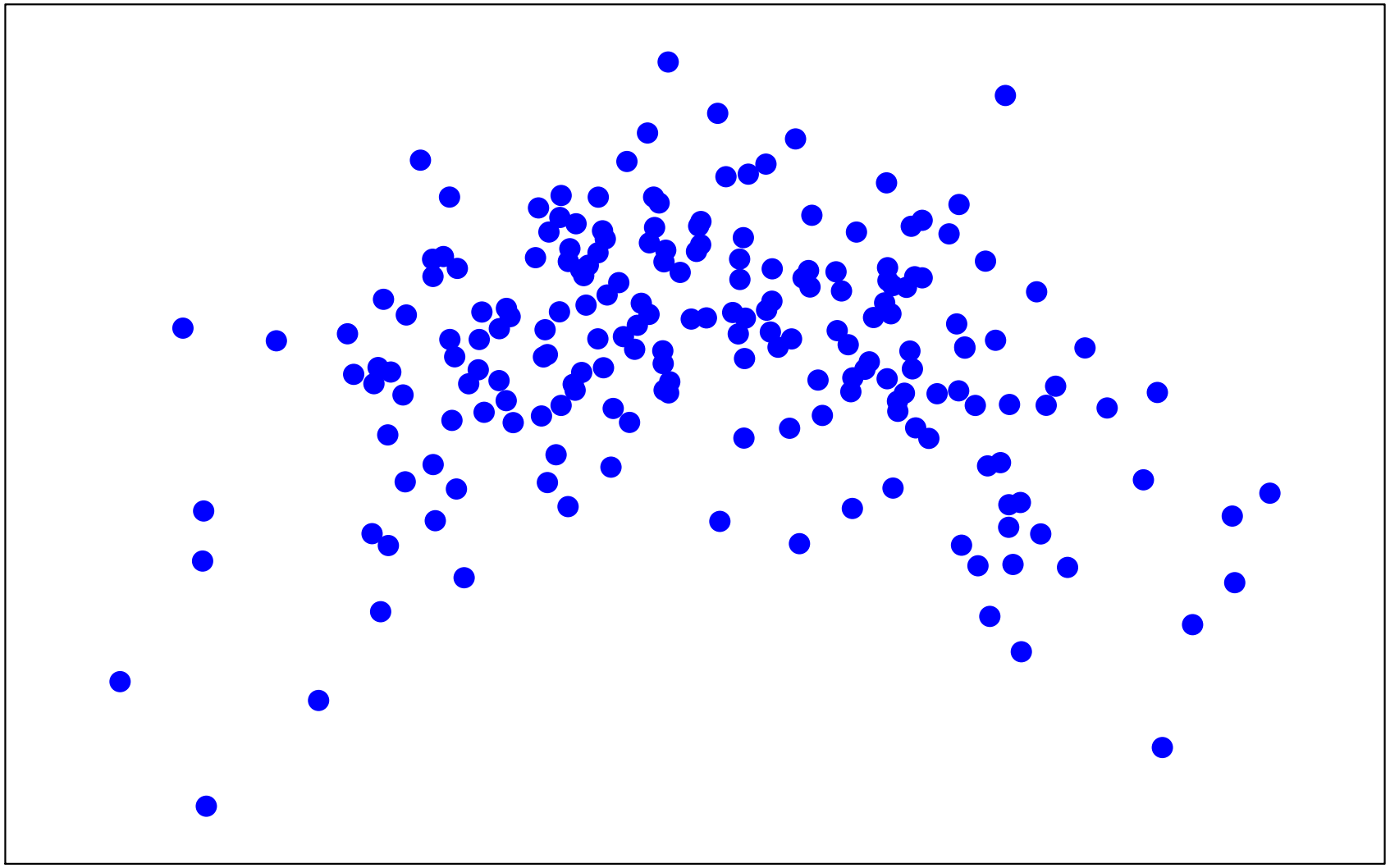
We don't *really* care about the exact location of every single point.

- How **many** points in one 'vicinity' compared to others?
- Any 'outliers' far from all other data points?

In one dimension, histograms answer these questions by **binning** the data

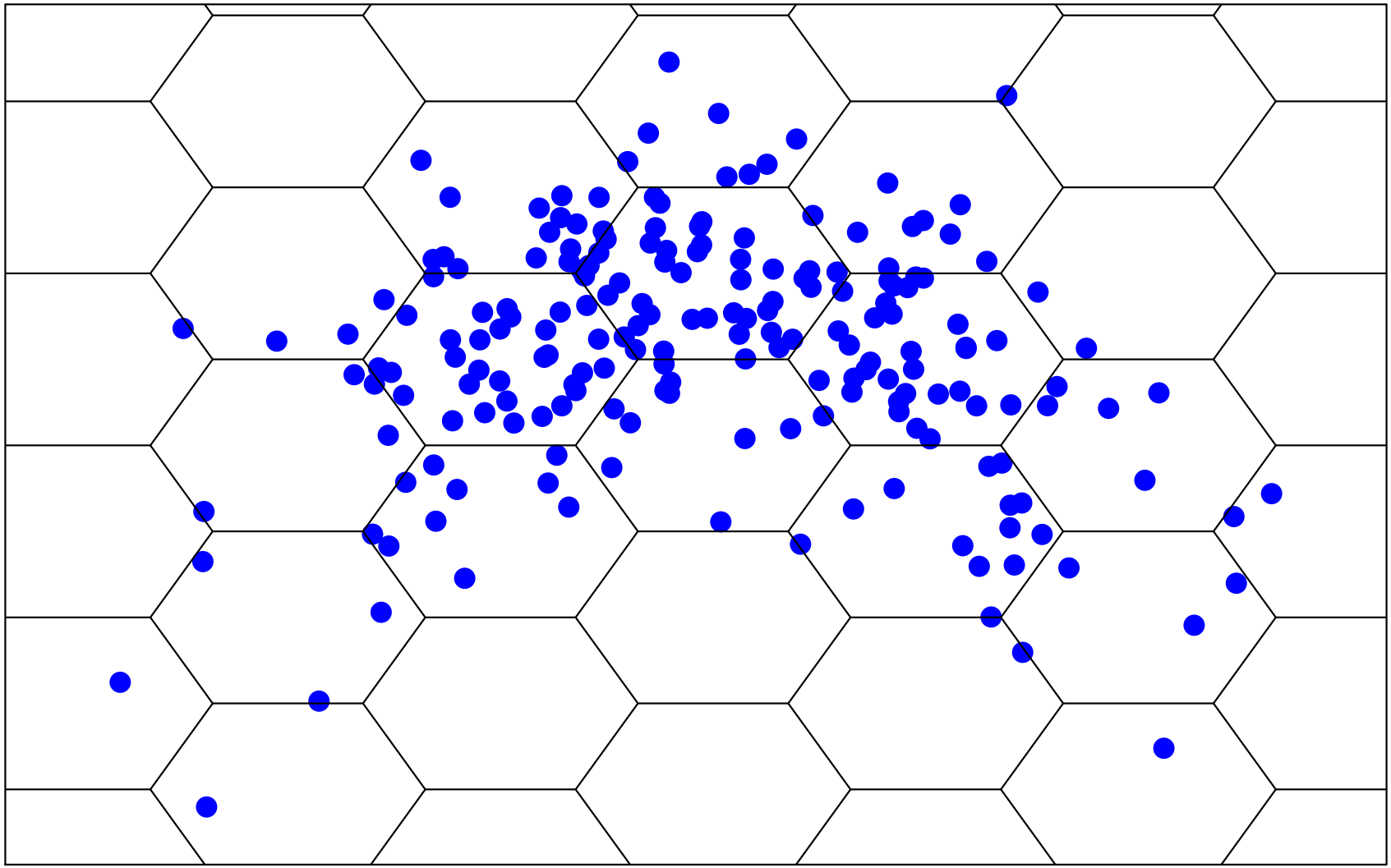
Bioconductor favorites: hexbin

Binning in two dimensions;



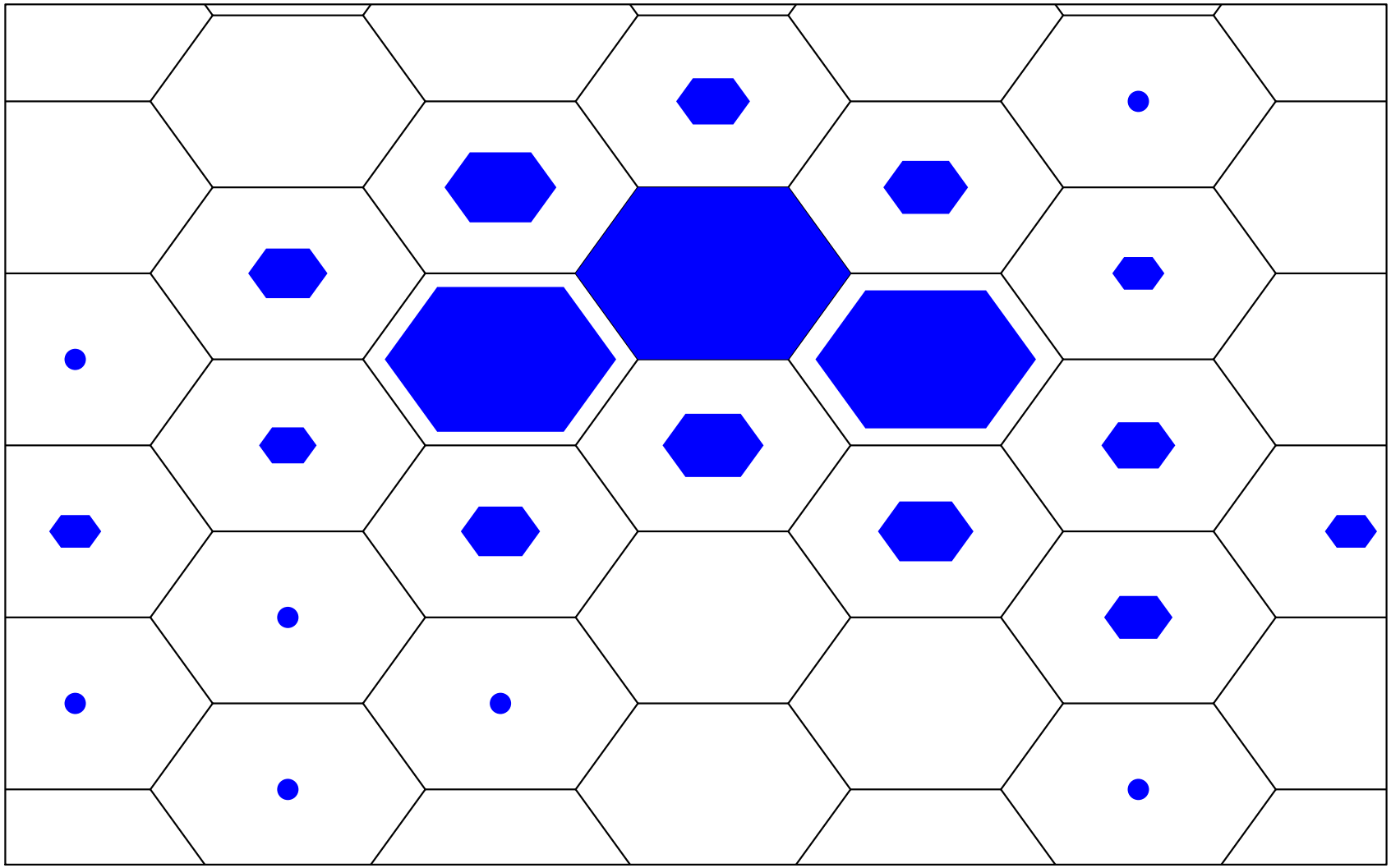
Bioconductor favorites: hexbin

Binning in two dimensions;



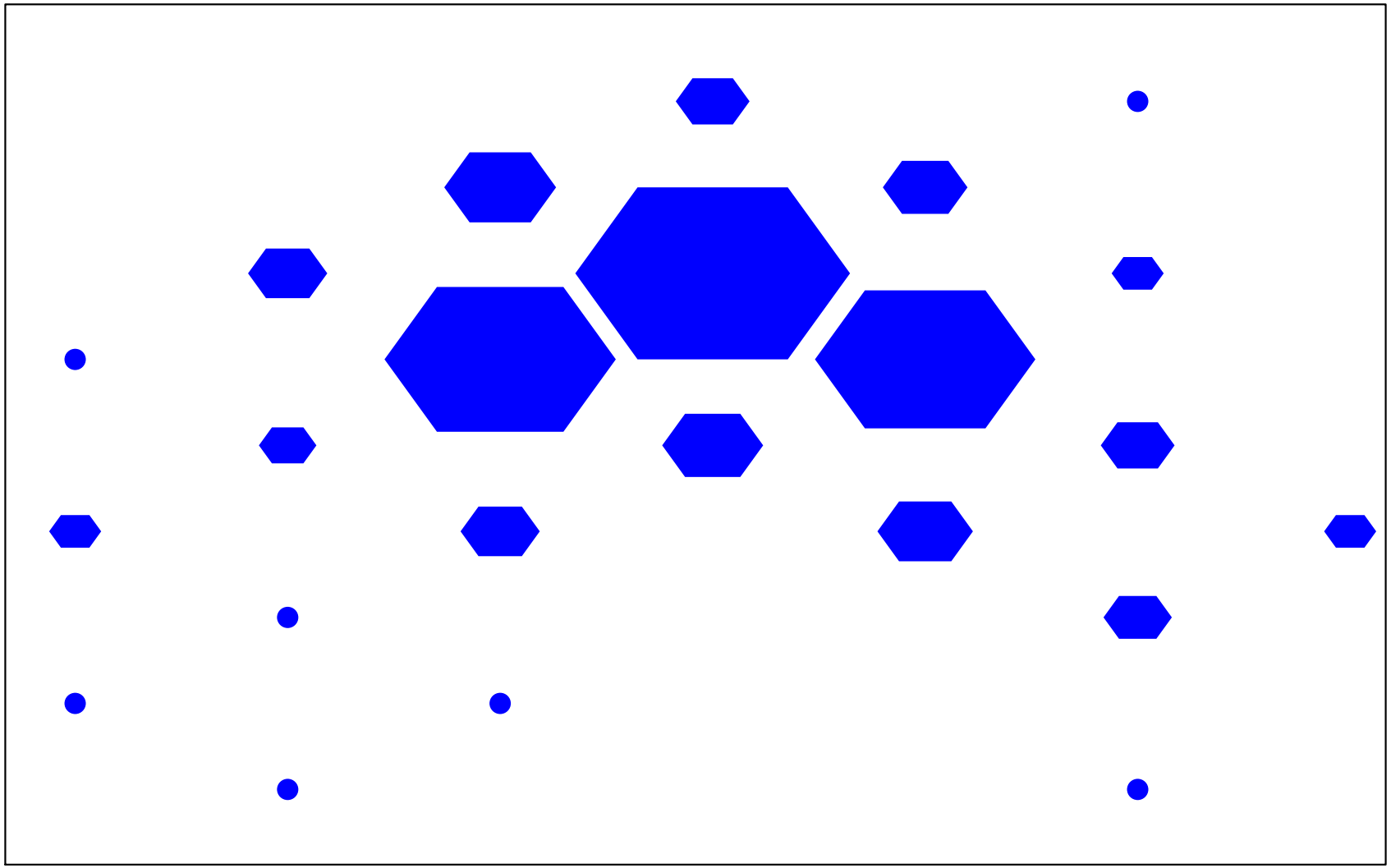
Bioconductor favorites: hexbin

Binning in two dimensions;



Bioconductor favorites: hexbin

Binning in two dimensions;

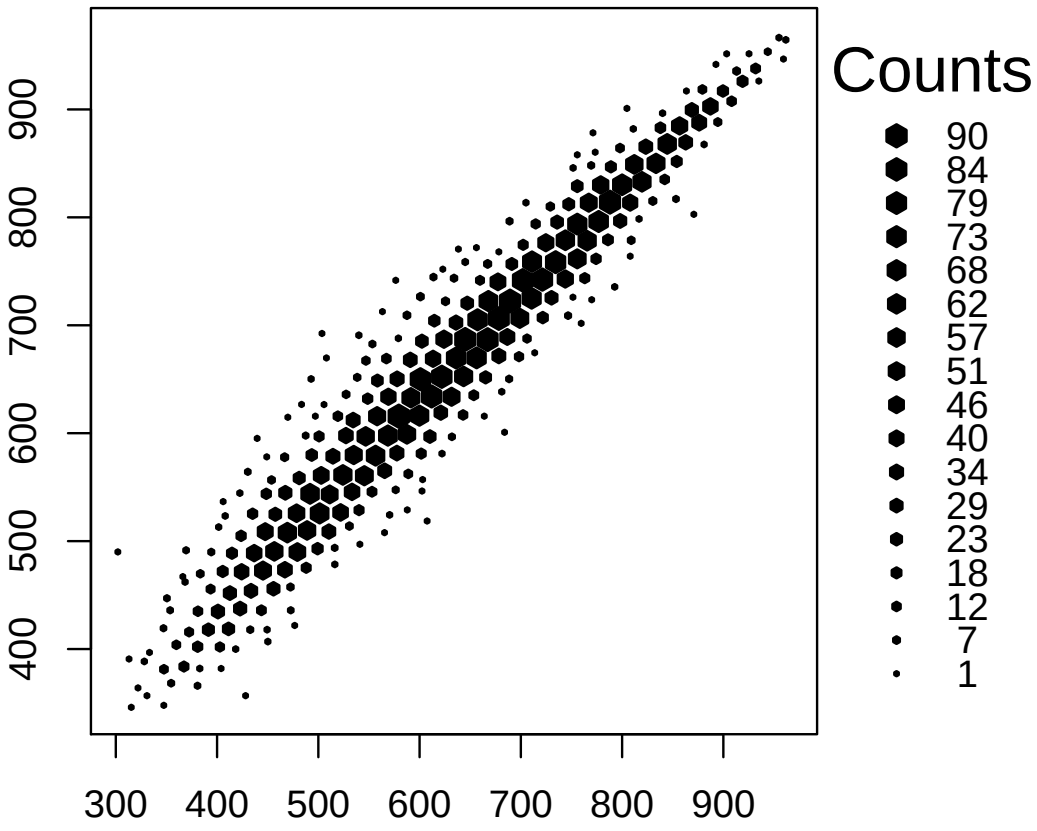


Bioconductor favorites: hexbin

Now with `hexbin`; recall we download from Bioconductor, not CRAN

```
> biocLite("hexbin")
> library(hexbin)
> with(apipop, plot(hexbin(api99,api00), style="centroids"))
```

Bioconductor favorites: hexbin



Bioconductor favorites: snpMatrix

`snpMatrix` is a Bioconductor package for GWAS analysis – maintained by David Clayton (analysis lead on Wellcome Trust)

```
biocLite("snpMatrix")
library(snpMatrix)
data(for.exercise)
```

A 'little' case-control dataset (Chr 10) based on HapMap – three objects; `snp.support`, `subject.support` and `snps.10`

Bioconductor favorites: snpMatrix

```
> summary(snp.support)
   chromosome    position          A1         A2
 Min.   :10   Min.   :   101955   A:14019   C: 2349
 1st Qu.:10   1st Qu.: 28981867   C:12166   G:12254
 Median :10   Median : 67409719   G: 2316   T:13898
 Mean   :10   Mean   : 66874497
 3rd Qu.:10   3rd Qu.:101966491
 Max.   :10   Max.   :135323432
> summary(subject.support)
       cc          stratum
 Min.   :0.0   CEU    :494
 1st Qu.:0.0   JPT+CHB:506
 Median :0.5
 Mean   :0.5
 3rd Qu.:1.0
 Max.   :1.0
```

Bioconductor favorites: snpMatrix

```
> show(snps.10) # show() is generic
A snp.matrix with  1000 rows and  28501 columns
Row names:  jpt.869 ... ceu.464
Col names:  rs7909677 ... rs12218790
> summary(snps.10)
$rows
   Call.rate      Heterozygosity
 Min.   :0.9879   Min.   :0.0000
 Median :0.9900   Median :0.3078
 Mean   :0.9900   Mean   :0.3074
 Max.   :0.9919   Max.   :0.3386
$cols
     Calls        Call.rate          MAF               P.AA
 Min.   : 975   Min.   :0.975   Min.   :0.0000   Min.   :0.00000
 Median : 990   Median :0.990   Median :0.2315   Median :0.26876
 Mean   : 990   Mean   :0.990   Mean   :0.2424   Mean   :0.34617
 Max.   :1000   Max.   :1.000   Max.   :0.5000   Max.   :1.00000
      P.AB             P.BB              z.HWE
 Min.   :0.0000   Min.   :0.00000   Min.   :-21.9725
 Median :0.3198   Median :0.27492   Median : -1.1910
 Mean   :0.3074   Mean   :0.34647   Mean   : -1.8610
 Max.   :0.5504   Max.   :1.00000   Max.   :  3.7085
                                    NA's   :  4.0000
```

Bioconductor favorites: snpMatrix

- 28501 SNPs, all with Allele 1, Allele 2
- 1000 subjects, 500 controls (`cc=0`) and 500 cases (`cc=1`)
- **Far too much** data for a regular `summary()` of `snps.10` – even in this small example

Bioconductor favorites: snpMatrix

We'll use just the column summaries, and a (mildly) 'clean' subset;

```
> snpsum <- col.summary(snps.10)
> use <- with(snpsum, MAF > 0.01 & z.HWE^2 < 200)

> table(use)
use
FALSE  TRUE
  317 28184
```

Bioconductor favorites: snpMatrix

Now do single-SNP tests for each SNP, and extract the p-value for each SNP, along with its location;

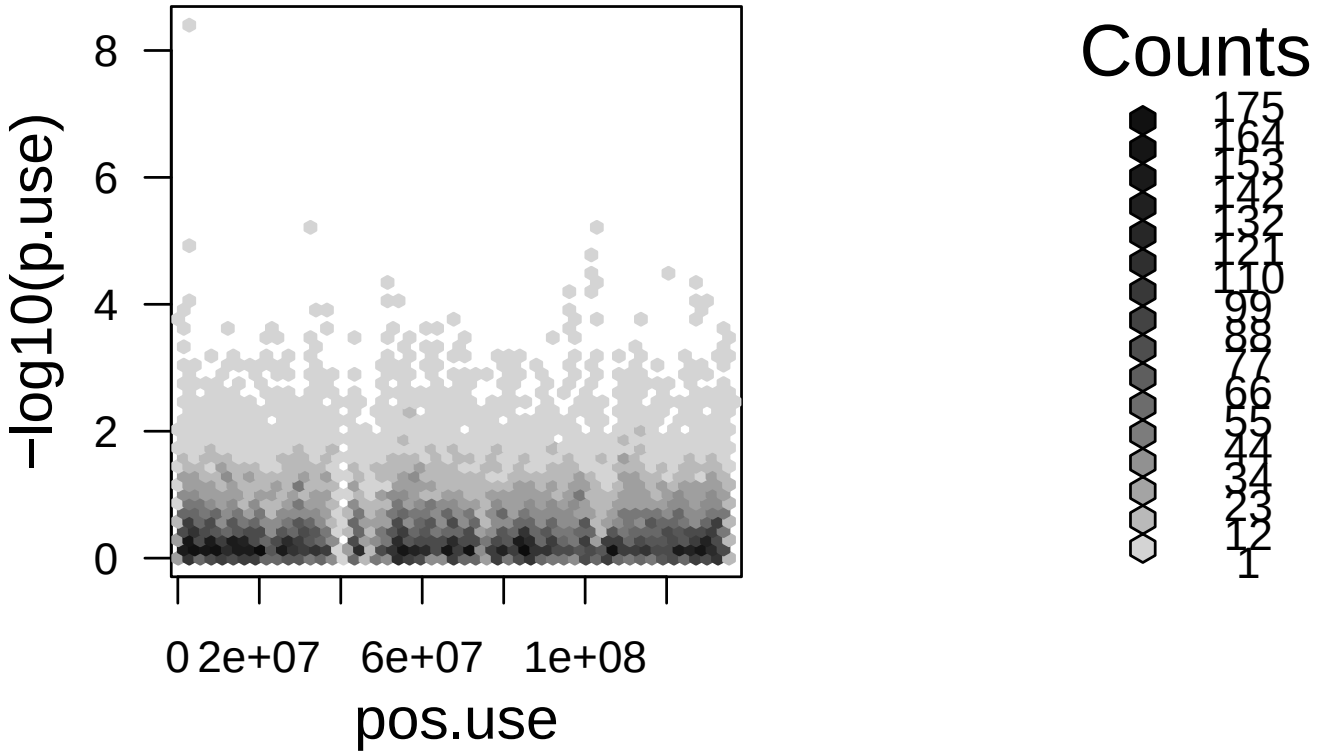
```
tests <- single.snp.tests(cc, data = subject.support,
+ snp.data = snps.10)

pos.use <- snp.support$position[use]
p.use   <- p.value(tests, df=1)[use]
```

We'd usually give a table of 'top hits,' but...

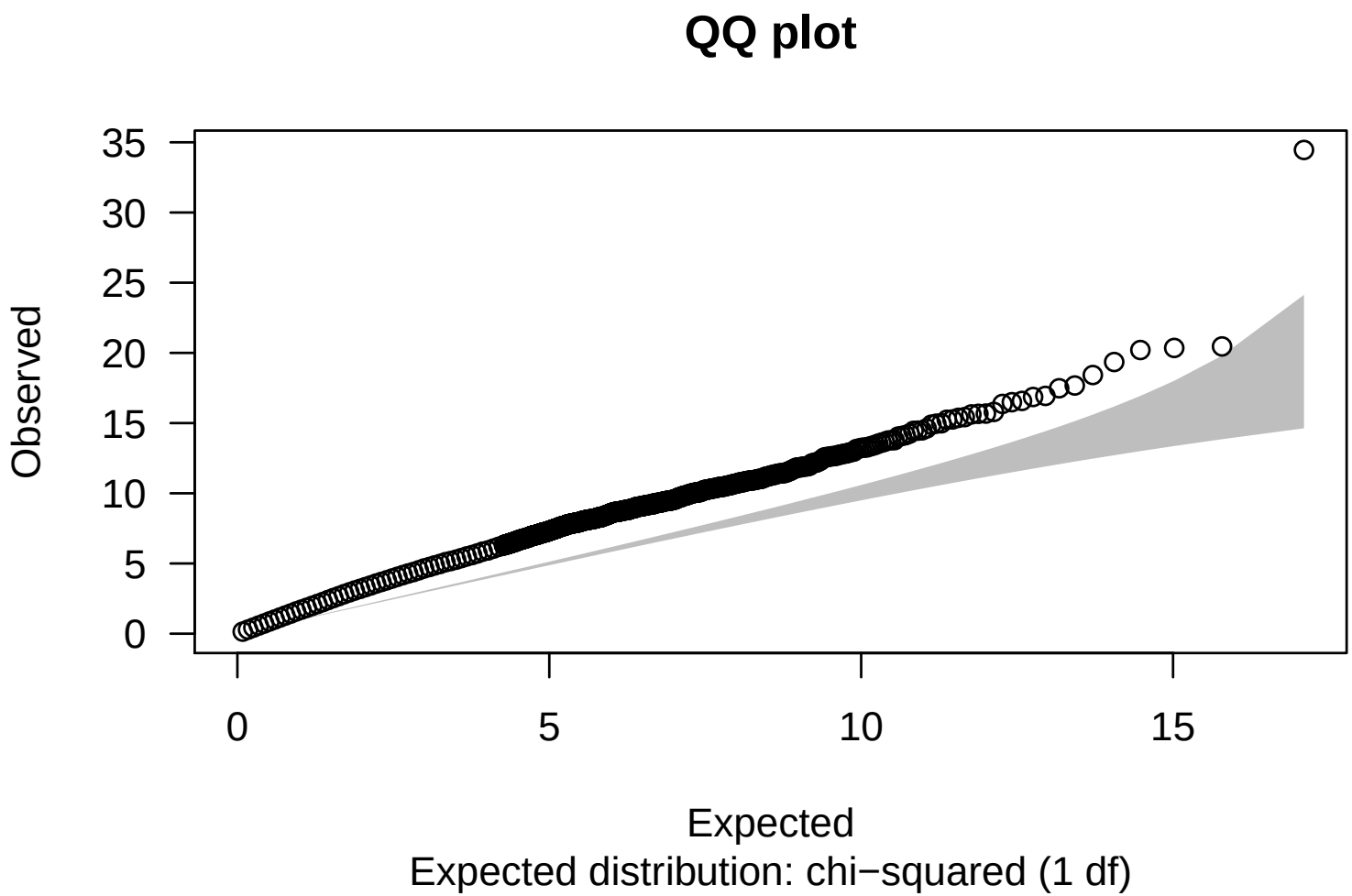
Bioconductor favorites: snpMatrix

```
plot(hexbin(pos.use, -log10(p.use), xbin = 50))
```



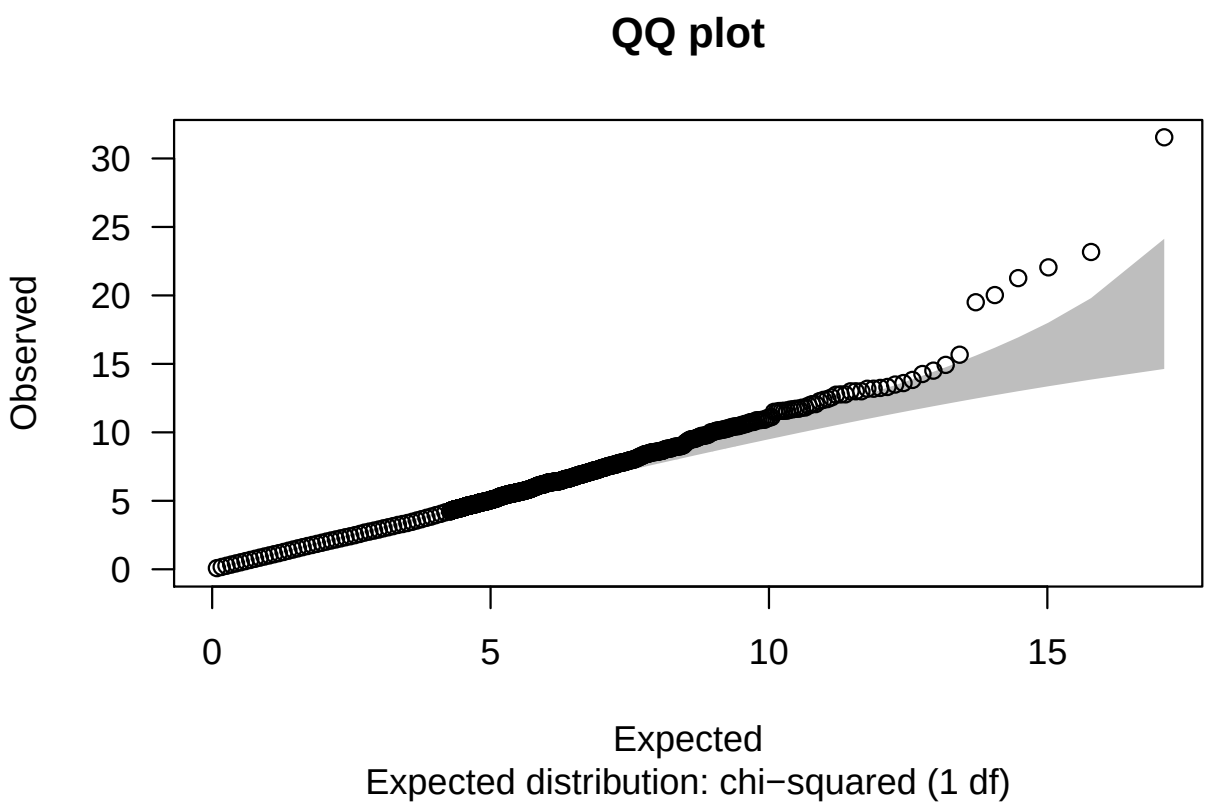
Bioconductor favorites: snpMatrix

```
qq.chisq(chi.squared(tests, df=1)[use], df=1)
```



Bioconductor favorites: snpMatrix

```
tests2 <- single.snp.tests(cc, stratum, data = subject.support,
+ snp.data = snps.10)
qq.chisq(chi.squared(tests2, 1)[use], 1)
```



Bioconductor favorites: snpMatrix

`snpMatrix` makes use of clever storage of 0/1/2 data, as well as quick implementation of the limited analysis jobs we often want to do in GWAS

- Recently updated to permit 'imputed dosages', which are $\in [0, 2]$
- Doesn't do the full range of regressions we may want – `lm()`, `glm()`, `coxph()`.
- Even with clever data storage, we'll run out of memory eventually – hence, in the GWAS I work on, we use netCDF and write our own code

Other packages – GenABEL

Yurii Aulchenko (one of my CHARGE co-authors) wrote the `GenABEL` package, which is on CRAN and here;

`http://mga.bionet.nsc.ru/~yurii/ABEL/`

It's very similar to `snpMatrix` – several CHARGE groups like it.

- Greater regression flexibility
- Comes with meta-analysis functions – which are part of life, in GWAS
- Also code for IBS, and computing principal components of SNP data (we use `C` to do this – and grad students)
- Lots of documentation/examples

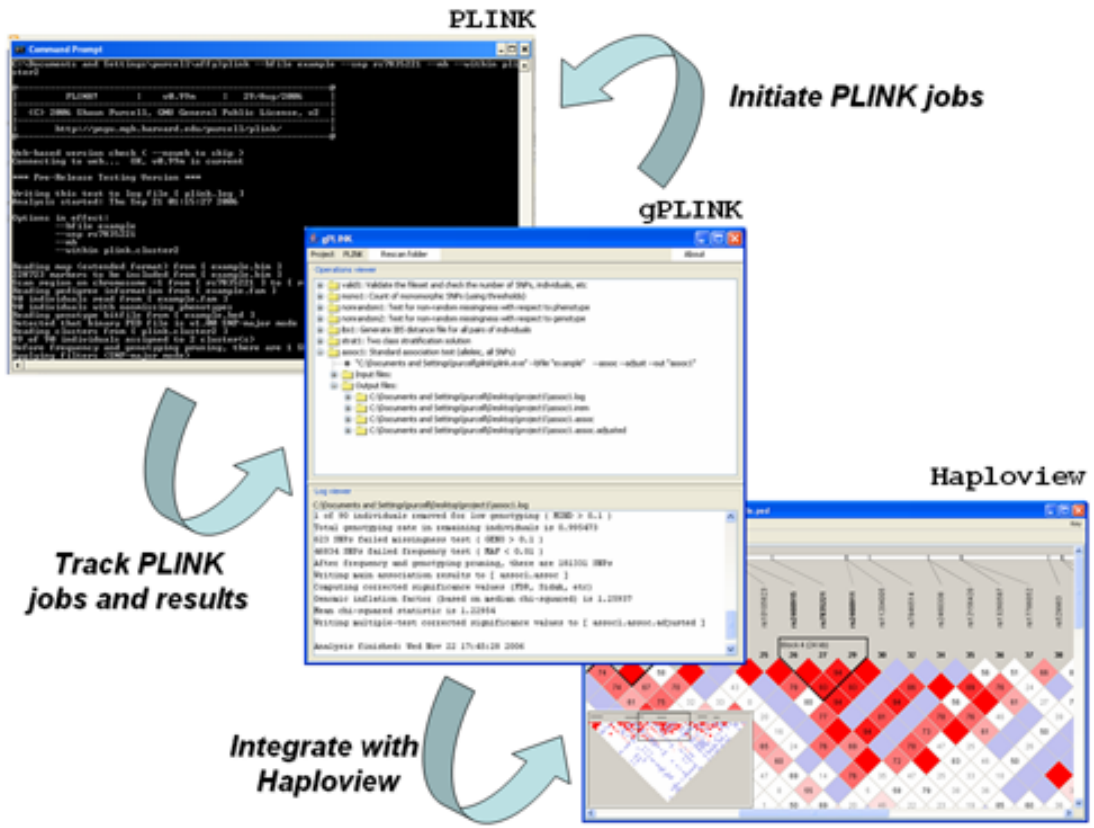
Other packages – GenABEL

Some things I am not so keen on;

- Still not as much regression flexibility as I'd like! (Yurii isn't an adopter of 'robust' standard errors...)
- I don't know how it treats e.g. non-convergence of `coxph()`. In practice, I want to know this
- ... it seems curmudgeonly, but I'm not a huge fan of 'packaging' basic commands stuck inside bigs loops. The learning-curve induced by all the weird things regression *can* do is very valuable – I want *someone* on each GWAS project to know that stuff

Other R-centric software

Expect to run into this;



http://pngu.mgh.harvard.edu/~purcell/plink/

Other R-centric software

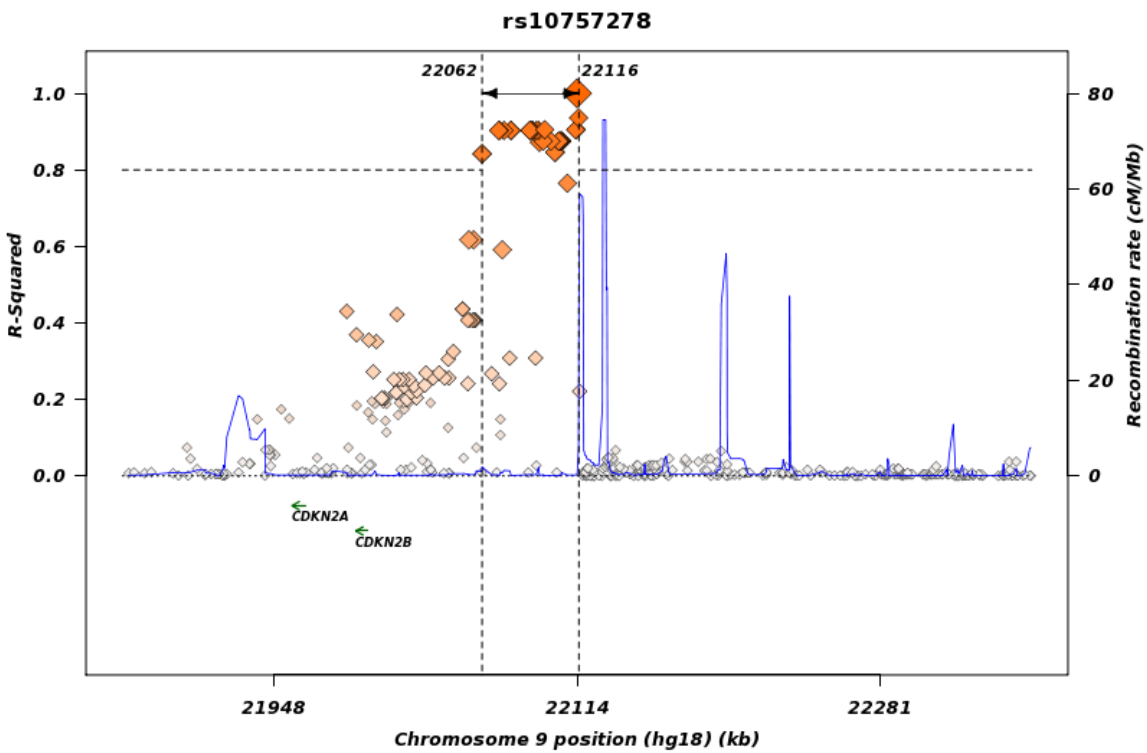
- PLINK (one syllable) handles the methods we've been talking about
- Latest version accepts `R` code! So you can e.g. persuade it to use `coxph()`
- gPLINK (two?) is a GUI interface to the command-line version
- Also does other jobs, including imputation (though concensus is that other methods are better, e.g. MACH, BIMBAM, IMPUTE, Beagle)

Dangerously pointy-clicky for my taste! I *want* people to think about e.g. patterns of missingess. *No-one's* intuition is great at $p < 10^{-\text{exciting}}$; are you sure of what you're getting?

Also, for some innocuous jobs, it'll do quirky things, e.g. for kinship coefficients there's a hidden (!) Hidden Markov Model

Other R-centric software

This is a 'regional association plot'



http://www.broadinstitute.org/mpg/snap/

Other R-centric software

No GWAS paper is complete without one!

- Original `R` code is (was?) available on Paul deBakker's website (Harvard)

- You could hack together your own quickly – it's p-value versus SNP location, with some funky colors/symbols (Getting the recombination rate data would be a hassle)

- These days, we use the SNAP site – for identifying nearby genes, this is fine. (For genome-wide inference you want a QQ plot – Manhattan plots are for 'sales pitches')