Armidale Animal Breeding Summer Course 2015

Module 2

From Sequence Data to Genomic Prediction

Teachers: **Ben Hayes and Hans Daetwyler**
Department of Environment and Primary Industries
Bundoora, Victoria, Australia

2 February 2015 - 6 February 2015

Practicals

Contents

1. Sequence Data

The aim of this practical is to introduce sequence data from raw reads to alignment and finally to variant detection and genotyping. Checking of data quality and filtering is emphasised at each step. We will be using the Galaxy web interface https://usegalaxy.org This website provides most of the tools needed to manipulate, check and visualise sequence data. You will find lots of tutorials on how to run the various programs. All programs on it are also available as stand-alone open source programs. As a teaching tool it is useful because it is point and click without the need to know a lot of command line or R syntax. I have made small data files that will run in a reasonable time. For large datasets, the upload and download times will be too long. To resolve this, Galaxy can be setup as a web interface for an institution's own computer cluster and can be integrated with their local job scheduling software, removing the need to up and download all data.

First, a user account needs to be created on Galaxy:

Go to: https:// usegalaxy.org
Click on "User" and follow the links to create an account. You should have immediate access after that.
Now login under the "User" tab and you are ready to start analysing sequence data!

1.a Raw reads and quality control

The file that contains the raw reads and base quality scores is called a fastq file. The format for fastq files has been introduced with more detail in the lecture slides. Briefly, each read has 4 lines dedicated to it. The first line starts with an @ and is followed by text identifying the sequencer, and flow cell, and whether it passed the chastity filter, the second line contains the basecalls, the third line contains a + signifying that the next line are the phred scale quality scores per base in symbol form. Each symbol is unique and stands for a certain phred quality.

NOTE: There are several different keys/formats to assign a symbol to a phred score. A good description of them can be found here:
http://en.wikipedia.org/wiki/FASTQ_format
Our data is in Illumina Casava 1.8+ format and is consistent with Sanger Phred+33.

Example.fastq

@HWI-ST690:130:D058YACXX:4:1101:1220:2180 1:Y:0
GGNCANAAATAAAGNATNNAT…
+
^$>>BBBCTT##...

We now want to **upload some data** to your Galaxy account. In Galaxy, click on "Get Data" in the left panel. Then click on "Upload File". In the middle panel you can browse for the file you want to upload. Galaxy will automatically detect what file you are uploading. Click "Excecute".

Upload **SeqExample1-bbmg.fastq** and **SeqExample2-thm.fastq**

You should now see these datasets in right hand side panel (you may have to refresh by clicking the refresh button (arrow circle) at the top of the right panel. Any files created on Galaxy can be downloaded by clicking the floppy disk in the right panel.

In the left panel, click on "NGS: QC and Manipulation".

We now **need to "groom" the fastq** file. There are various formats (different symbols for same phred score) and this step converts all formats to Sanger format. We need to do this even though our fastq files are already in Sanger format.

Click on "FASTQ Groomer", select the fastq file, select "Sanger" and "excecute". Do this for both fastq files. (The example data is Illumina Casava 1.8+, therefore it is Sanger encoded)

As before, the new files are shown in the right panel.

Summary statistics can be calculated by clicking on "FASTQ Summary Statistics" in the left panel and selecting the groomed fastq in the middle panel. You can view the results by clicking on "the eye" in the left hand panel.

The statistics are calculated for each base position across all reads. For example, if read are 100 bases long then 100 rows of statistics are calculated.

FastQC is another program that can be used to calculate quality statistics, including GC and per base sequence content, etc. You can run FastQC by clicking "FastQC: Read QC" in the left panel. Inspect it by clicking "the eye" in the right panel. It makes plots of the statistics for easy inspection. (it can also be downloaded here http://www.bioinformatics.babraham.ac.uk/projects/fastqc/)

We now have a good idea about the unfiltered quality statistics of both fastq files. **Which file** do you think **is higher quality**?

We now want to trim the end of the read based on their quality scores. Click on "Filter FASTQ" on left panel. There are various options. To start set the minimum base quality at 20. Do this for both fastq files.

Now rerun either FastQC or FASTQ Summary Statistics on the filtered files. Have they improved?

Now try "filtering" by just trimming a certain number of bases of the ends of reads. Compare the amount of data and summary statistics to the previous filters based on quality.

Try different thresholds and repeat filters and quality summaries.

1.b Alignment bam files: QC and viewing

Aligning the reads means that we want to find the location of each read on the reference genome. This is also called mapping. In principle, this should be easy. All we need to do is go along the reference genome and find a matching sequence to our read. However, this gets tricky because there are some duplicated or repetitive regions in the genome where the reference assembly might be suboptimal and also our reads are likely to have some errors in them even after filtering. While you could align the fastq files given in part a) above on Galaxy, it would take too long to do during a practical. Also, they are only a subset of reads from the whole genome and the resulting alignment bam would be very low coverage in any given region.

Instead we have prepared two bam files: one higher coverage (~40x) **HighCov10000-1010000.bam** and one lower coverage (~4x) **LowCov10000-1010000.bam**. The regions on them is very small, just 1 megabase on *bos taurus* Chr1 bp 10000 to 1010000 on cow genome. They are aligned to UMD3.1.

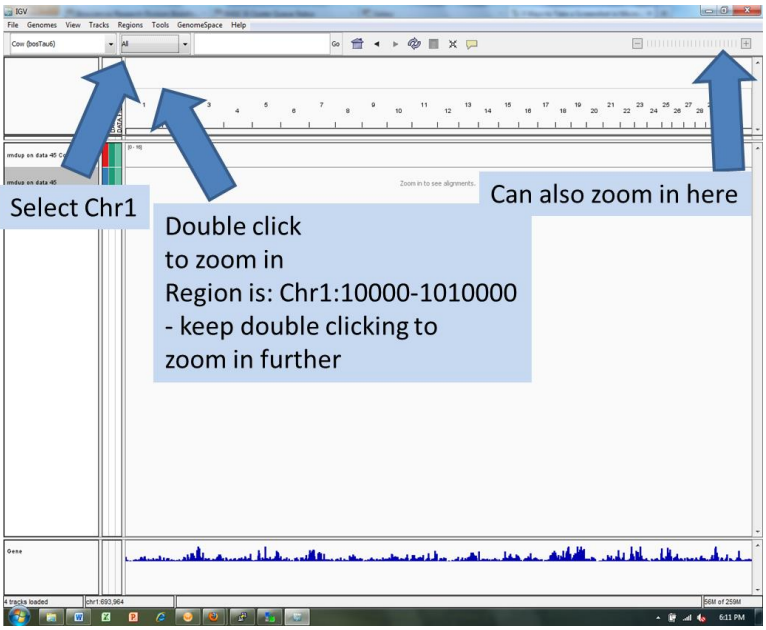
Upload the two bam files to Galaxy. It will detect that it is a bam file. Choose "Cow Nov. 2009 (Bos_taurus_UMD_3.1/bosTau6)(bosTau6)" as the reference. It may take a few minutes for the larger one.

Let's first run a FastQC report on both bam files to see what the quality of the mapped reads are. Hint: same procedure as in 6.10.a but select the bam file of choice.

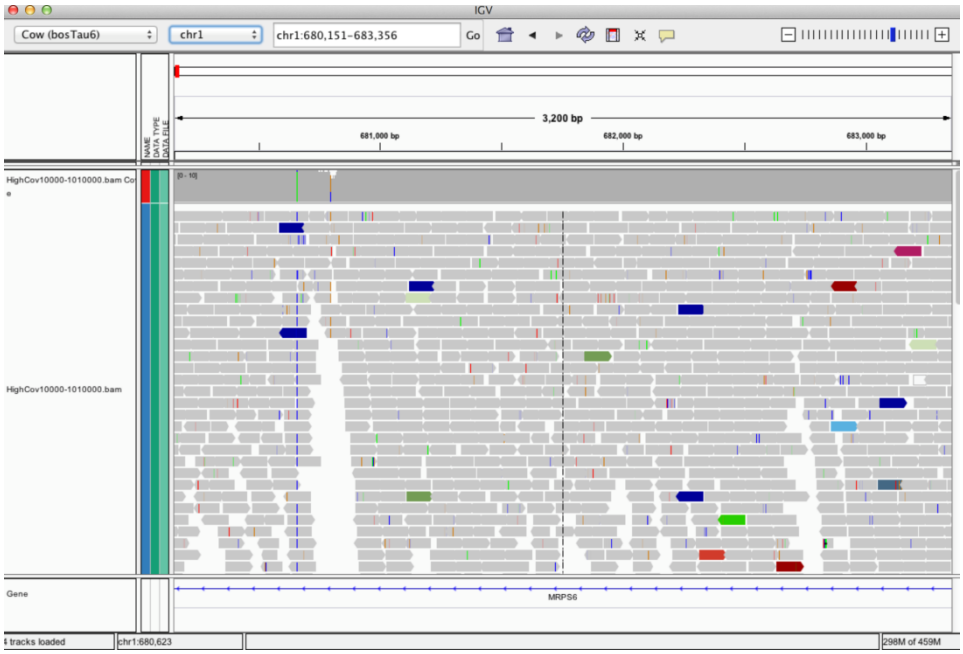
Flick through the various windows and look at the statistics. Do you think they are filtered adequately?
Considering some of the filtering metrics for reads discussed in the lecture, do you see any reads and bases below phred 20?

Let's have a look at the low coverage bam file in IGV (Intregrated Genome Viewer, Broad Institute, http://www.broadinstitute.org/igv/). Go to the appropriate box in the right panel. Click on the heading to expand it. Scroll down and click on "display with IGV web current". This will ask you to install some java program on your computer. Allow this.

Eventually a new window should appear looking like this:



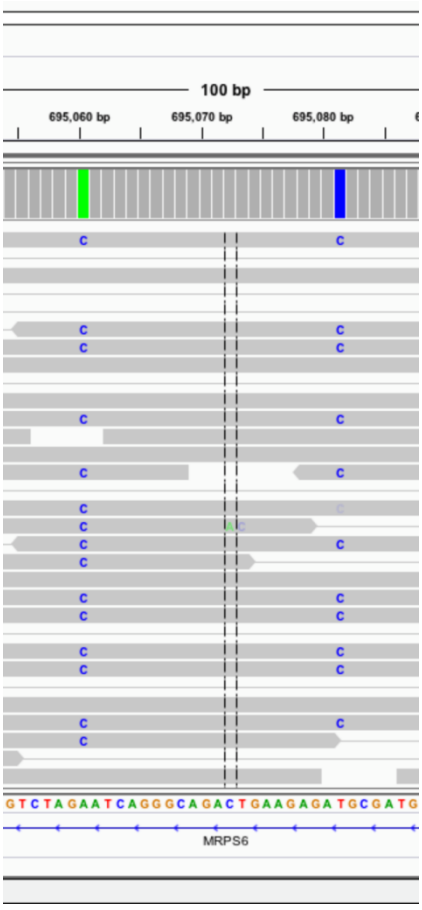
Now we need to zoom in on our region. Click on "ALL" on top just beside the reference "Cow (bosTau6)". Select Chr1 from drop down menu. Now, double click on the bar in the top panel/track. Keep double clicking to zoom in to the region. You know you're there when you start seeing reads (i.e. a bunch of layered bars in second/middle panel).
The more you zoom in the more detail you will see.



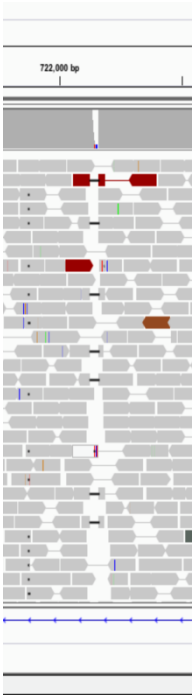
If you right click on the reads you can choose how you would like to view the data. There are many colour options available. One useful option is to see reads as pairs. Show all alternative bases is useful for seeing SNP.

This picture shows a **read** that seems to **not** have **mapped correctly**. There are just too many differences in it. By chance, its paired read (to its left) has perfectly mapped to the reference.

The picture below **shows two SNP** that have been identified in the high coverage animal. The animals seems to be heterozygous for these SNP, although the distribution between the two is not 50/50. A priory we have an equal probability of observing either allele at a heterozygous position within a diploid individual.



The two pictures below show **deletions,** which is a section of reference genome that does not occur in this individual. The one on the left is just a 1 bp deletion and the individual is heterozygous for it. The deletion on the right is quite a bit longer.



Inspect the low coverage bam file in IGV. Do you notice anything odd? Also open up the high coverage file and compare the two. Aside from the obvious difference in read depth, is there a different pattern to the way the reads are mapped to the reference genome?
If you notice a difference, which one do you think is better? In the file that is inferior, can you think of a way to filter it to remove the issue?

If you can think of a way, use that filter (Hint: it is listed under "SAM tools" in the left panel). Redo the FastQC analysis and also reinspect the bam in IGV after filtering. Is it fixed and have the stats improved?

1.c Variant and Genotype Calling and Quality Control

We could now use these two bam files to **call variants** (SNP and indel) and to call genotypes at those variants with Samtools or GATK. However, they were aligned outside of Galaxy with a slightly different reference than the default cow reference on Galaxy. You could upload to reference to Galaxy and then use it to call the variants. I am happy to provide this reference for you to try on your own. Alternatively, you can download it from http://www.1000bullgenomes.com/ As it is a large file, it would take too long to do in this practical.

If you would like to call the variants: Click on "SAM tools" in the left panel, scroll down and click on "MPileup", select the bam files in the middle and execute. This produces a bcf file, the binary form of the vcf (variant call file) file discussed in the lecture.
You will need to start an additional job on Galaxy to convert this bcf file to a vcf file. In the left panel click on "bcftools view" and select the bcf file in the middle panel, execute.

Of course this can be accomplished with Samtools on the linux command line using:
samtools-0.1.18/samtools mpileup -r Chr1:10000-1010000 -P ILLUMINA -ugf all_UMD3.fa $bam1 $bam2 | samtools-0.1.18/bcftools/bcftools view -bcvg - > var.rawHighLow.bcf

samtools-0.1.18/bcftools/bcftools view var.rawHighLow.bcf > var.rawHighLow.vcf

I have prepared a vcf file using this command for you using the high (BBMg) and low (HOLAUSM000A00010228) coverage bam files and this produced a small vcf file with ~4000 variants (remember it's only a 1mb region) VarHighLow.vcf. There are various open source programs that you could use to filter this vcf file for various quality characteristics. For simplicity, we will use excel to filter the variants in this practical using the file **VarHighLow.xlsx** (both files provided).

The vcf file was tab delimited. Each column in the excel file is a tab field. Using a few excel commands we can further **subdivide columns and filter variants**.

First, it's likely best **to remove the header** where all file components are explained (i.e. rows 1-26). Next we want to **expand the INFO column into separate columns** according to the " ;" delimiter with the following commands (works in Windows 2010+).
Select columns I, J, and K. Cut and paste them to the right approximately 10 columns or more. Now select column H (INFO). Above, click on "Data", and then on "Text to Columns" . In the popup, click on delimited, next, select ";", Finish. The INFO column should now be expanded.
Notice now that the first column in the INFO field signifies whether the variant is an INDEL or not. This causes the remaining columns to be out of sync. It's best to sort on the INFO column now and cut and paste all indels to the next worksheet. How many indels versus SNP were called?

Working now with just the SNP, lets calculate the **mean and standard deviation of read depth and devise a filtering strategy** . You will have to split DP from the read depth by splitting the column as above. There are two reasons to filter on read depth: 1) It's too low so heterozygotes are not called consistently, and 2) it's too high because many reads have mapped to the same repetitive region of the genome. In the first case we can require a minimum read depth meaning that we have enough chances to observe both alleles. Try a minimum of 5 reads to start. In the second case we can filter reads that fall outside of certain range (i.e. extreme values). Assuming that read depth is normally distributed (it's actually Poisson but that makes little difference in this case) we could remove reads that are more than 3 standard deviations from the mean read depth.
How many reads are removed if you applied these filters?

Let's now **check** the data for variants with **2 alternative alleles**. Can you think of a biological case why this could occur? What sequencing problem could also result in two alternative alleles?

Another important criteria is whether alleles have been **observed on forward and reverse reads** of the sequence data. Sequencing reads occurs in both directions. If you only observe an allele in one direction then it is likely that something has gone wrong. It's good practice to remove reads where that alternative allele has only been observed in one direction. In our vcf file these stats are in the DP4 field. For example: DP4=2,4,3,5 where there are 2 observations of the reference allele in the forward direction, 4 in reference reverse, 3 alternative forward, and 5 alternative forward.

How many reads have a 0 for number of alternative and reverse reads? Check their QUAL scores. Would you also remove variants that have 0 reads for the reference allele? Why or why not?

Next **look at mapping quality** (MQ). Are there variants with <30 mapping quality? Can you see other problems with these variants?

Go through the filters discussed in the lecture and apply them (if possible). Which ones do you think have merit? Are there additional filters that you could think of? Are there some filters that you don't think are necessary?

Finally, let's have a look at the **genotypes and their quality**. They are in the following format, GT:PL:GQ , where GT is the genotype 0/0, 0/1, or 1/1, PL are the phred scaled genotype probabilities for each possible genotype (i.e. 3 probabilities if there are 2 alleles), and GQ is the phred scaled genotype quality. Note that in the case of genotype probabilities, lower phreds are better where as in the various quality metrics a higher phred score is better.

Inspect the genotype probabilities and compare them with the actual genotype called. Do they agree? Do you see cases of equal probabilities (e.g. 0,0,0)? Can you determine the meaning of this by looking at the other quality statistics for this variant?

Are the genotype quality scores different for the two animals? Why would that be? If you see low GQ scores what other metrics are also low at that variant?

2. Using Beagle to impute missing genotypes

In this practical you will use the BEAGLE program (Browning and Browning 2007) to impute from sparse genotypes to denser genotypes in a data set from a 50K dairy cattle data set.

Inspect the data. The first file, reference_50.txt, contains genotypes for 22 animals that have been genotyped for all markers. These genotypes are from chromosome 1, the first 50 markers. The first line of the file is the animal ids, from one to 22. There are two columns for each animal, one column for each allele at each marker. The second row is the genotypes for marker one, two alleles per individual. The third row is the genotypes for the second marker and so on. The genotypes are unphased at this point. The alleles are coded 1,2, and 0, with 0 for missing.

The second file to check is target_50.txt. These are genotypes for 3 animals for 5 markers, which are an evenly spaced subset of the 50 markers above (eg this would be an approximately 5K array).

The other file you will need is the map file, telling the BEAGLE program the alleles at each marker. The map file is reference_map_50.txt, the first three lines of which are

```
Hapmap43437-BTA-101873 113641          1 2
ARS-BFGL-NGS-16466 244698              1 2
Hapmap34944-BES1_Contig627_1906 369418 1 2
```

Now to run the BEAGLE program you will need to open a command prompt, and make sure that the BEAGLE executable beagle.jar and the data files are in the same location.

Change directory
C:> D:

Change folder
D: cd <foldername>

See all files in a directory
dir

The command for running beagle with the data above, with a reference and target population, is

java -Xmx1000m -jar beagle.jar unphased=reference_50.txt unphased=target_50.txt markers=reference_map_50.txt missing=0 out=5K

Note that command is all on one line. The out command will in this case give all the out files the prefix 5K.

You will need to use the 7z program to look at the output files, as they have been zipped using a program called gzip.

The file 5K.target_50.txt.phased.gz contains the imputed, and phased genotypes. Again, there are two alleles for each marker, but these are now phased, eg the first allele of the first marker is on the same chromosome segment as the first allele of the second marker.

Now we will check how accurately BEAGLE has imputed the missing genotypes. The file target_true.txt contains the real genotypes at all markers for our three "target" animals.

Compare the true and imputed genotypes to calculate an accuracy of imputation. There are a few steps to doing this, which can be done either in excel or R. For excel, paste the true genotypes and the imputed genotypes beside each other in a spreadsheet. As the true genotypes are unphased (eg the alleles could be in any order), in order to compare the genotypes you will need to calculate basically an X matrix for both three true and imputed genotypes. This matrix has dimensions number of markers (50) by number of individuals (3). The elements are the number of two alleles, which can be calculated from the genotypes as allele1 + allele 2 – 2.

Eg. If for the first animal at the first marker, the genotype was 1,1, the element of X would be zero.

Calculate a separate X for both the true genotypes and the imputed genotypes.

Then count up the number of genotypes that are the same in the imputed and true genotypes.

For example, for two markers the true and imputed X matrix could be (each column is an animal)

True Imputed
1 0 1 1 1 0
1 2 1 1 2 1

Then the accuracies of imputation for each animal are 2/2=100% for animal one, 1/2 = 50% for animal 2 and 1/2 =50% for animal 3. The counting up can be done with an IF statement in excel.

What are the accuracies of imputation for our three target animals? What are some possible reasons for the differences in accuracy of imputation?
Finally, have at look at the file 50_SNP.reference_50.txt.gprobs.gz. This file gives for each animal, the probability of each genotype for each animal. This is a measure of the uncertainty of the imputation. Each line of the file contains the marker name, the two alleles at the marker (1 and 2 for all markers in our case), then for each animal the probabilities of the three genotypes 11, 12 and 22 (or 0,1,2 in our X matrix).

Can you find maker where there is a lot of uncertainty in the imputation?

How would you build an X matrix for the genomic selection methods that takes account of uncertainty of imputation?

3. Assessing the extent of linkage disequilibrium in HaploView

We will use the HaploView program to calculate r^2 values. The data set we will use is 10 SNP markers on a section of chromosome 20 genotyped in 325 bulls.

The genotype (in linkage format) file for HaploView has the following format

Pedigree_ID Individual_ID Sire_ID Dam_ID Sex Affected Marker1_Allele1 Marker2_Allele2

You can find out more about the genotype input file in the Help tab of haploview

The map file consists of two columns, the marker name and the position, eg

Marker1 19992222
Marker2 23100202

Import the genotype file "325_bulls_genos.txt". Import the file "map.txt". Set the minumum distance to calculate r^2 to markers less than 5000kb apart.

Are all the markers in Hardy-Weinberg equilibrium? Which marker has the lowest minor allele frequency?

Set the HW cuttoff to 0.0000, and click on the box to make sure they are all included.

Then click on the LD plot tab. To make sure the values are r^2, click Display -> Show LD values -> R–squared. The boxes show the r^2 values between the markers from 0 to 100. If the markers are in 100% LD, there will be a red box with no number.

Which markers are in the highest LD? Are there any markers in perfect LD?

Does the LD decay uniformly across the chromosome segment (for example look at marker 1 versus the rest)? How would you describe the pattern of LD with distance in this small chromosome segment?

4. Genome wide association study

Now we will conduct a genome wide association study using the same data and phenotypes.

Before we use go further, let's take a moment to get acquainted with R. We will use a simple example of multiplication of two matricies to obtain another matrix. Open the R graphical user interface by clicking on it. You should see the command prompt.

Let's multiply two matricies a and b to get a third matrix c.

The matrix a is a two by two matrix with elements:
1 1
2 2
The matrix b is a two by three matrix with elements:
1 2 2
2 3 4

We can input these matricies into the computer memory as:
> a <- matrix(c(1,1,2,2),ncol=2,byrow=TRUE)
> b <- matrix(c(1,2,2,2,3,4),ncol=3,byrow=TRUE)

To check the dimensions of a and be are correct type:
> dim(a)
> dim(b)

You can print a matrix at any time, eg
> print(a)

Now lets multiple matricies a and b to get a new matrix c:
> c <- a%*%b (%*% is the symbol for matrix multiplication)

Check the dimensions of c are correct,
> dim(c)
And that the c matrix has the correct elements:
> print(c) (you can compare this to the result in excel for example)

A matrix can be transposed using t(a), eg
> d <-t(a)

For convenience, a genotype file with genotypes re-coded as 0, 1 or 2 (the number of copies of the second allele) is given in xvec_day4.inp. For the 325 bulls, phenotypes for protein % in their daughters milk are given in the file yvec_day4.inp.

Now we write a small R script to read in the data, and fit a regression on the number of 2 alleles for each SNP.

To start a new script, click file and then New Script. Remember to save your script.

Then read in the data. The easiseast way to do this is to set your work directory to whever the files are stored first, then read in the data as a table:

```
setwd("C:/course_piacenza")
phenotypes <- read.table("yvec_day4.inp",header=F)   !No header on file
genotypes <- read.table("xvec_day4.inp",header=F)
```

Now for each SNP we are going to fit the model

y = mu + **X**b+**e**

Where **y** are the phenotypes, mu is the mean, **X** is the design matrix allocating phenotypes to genotypes for each SNP, b is the effect of the SNP and **e** is a vecot of random residuals. This can be done in R with the lm command (for linear model)

Lets fit the first SNP. We can do this as

```
lm(phenotypes[,1] ~ genotypes[,1])
```

The [,1] for genotypes tells R to use the first column of genotypes, eg the first SNP

The result gives the intercept (mean), and the regression coefficent, which in our case is the effect of the 2 allele. If you want just the regression coefficent returned,
`lm(phenotypes[,1] ~ genotypes[,1])$coeff[2]`

Now we would like to know how significant the SNP is. We can get this with the anova command,

```
anova(lm(phenotypes[,1] ~ genotypes[,1]))
```

If you want just the P value returned,

```
anova(lm(phenotypes[,1] ~ genotypes[,1]))$P[1]
```

Now to run the genome wide association study, get the effect of each SNP and it's P value and store them. This can be done by writing a loop for the number of SNP (10) and fitting the models above each time.

Now read in the map file (map_10_markers.txt).

Plot –log10(P value) against map position for the SNP. Which is the most siginficant SNP(s). Can you explain this result in terms of the linkage disequilibrium among the SNP in the previous practical?

Now plot the SNP effects against –log10 of their P values. Are the SNP with the largest effects the most significant? Why/why not. Will this always be the case in a GWAS study? And why do some of the SNP have the same effect?

4. Genomic selection using BLUP

In this practical you will perform genomic selection in a small data set using BLUP. The data set consists of a reference population of 325 bulls with daughter yield deviations (DYDs) for protein %. This phenotype is an accurate predictor of genotype, eg the heritability is close to one. The bulls have been genotyped for 10 SNPs.

Then there are a set of 31 calves who are selection candidates for this years progeny test team. They are genotyped for the same 10 markers. Your task is to predict GEBV for these 31 selection candidates. To do this we will need to predict the effects of the 10 SNPs in the reference population, using the equations:

$$\begin{bmatrix} \mathbf{1_n'1_n} & \mathbf{1_n'X} \\ \mathbf{X'1_n} & \mathbf{X'X+I\lambda} \end{bmatrix} \begin{bmatrix} \hat{\mu} \\ \hat{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{1_n'y} \\ \mathbf{X'y} \end{bmatrix}$$

Where g are the SNP effects, 1n is a vector of ones (325 x 1, **X** is a design matrix allocating SNP genotype to records, μ is the overall mean. We will use R to solve these equations. The **X** matrix has already been built for you, and is contained in the file xvec_day4.inp. The y matrix is contained in the file yvec_day4.inp.

What you need to do is write a small R script to solve the equations. This can be done by starting the script in notepad, then opening it in the R console.

The first lines should declare the parameters number of markers and number of records. A this point we will also specify the value of lamda as 10.

```
nmarkers <- 10      #number of markers
nrecords <- 325     #number of records
lamda    <- 10     #value for lamda
```

Next we will read in the files. Change the path to the location where you have stored the files. Note that these statements should all be on one line. Have a look at these files before opening them.

```
x <-
matrix(scan("d:/iowacourse/practicals/day4/realDataExample/xvec_day4.
inp"),ncol=nmarkers,byrow=TRUE)
y <-
matrix(scan("d:/iowacourse/practicals/day4/realDataExample/yvec_day4.
inp"),byrow=TRUE)
```

So now we have the matrix x, the vector y. We still need a vector of ones and a identity matrix dimension number of markers x number of markers…..

```
ones <- array(1,c(nrecords))
ident_mat <-diag(nmarkers)
```

The next step is to build the coefficient matrix. This can be done in blocks, eg….

```
coeff <- array(0,c(nmarkers+1,nmarkers+1))
coeff[1:1, 1:1] <- t(ones)%*%ones
coeff[1:1,2:(nmarkers+1)] <- t(ones)%*%x
```

You will need to build the other blocks. You will also need to build the right hand side of the equation.

The solutions can be obtained easily by using the inbuilt function solve,

```
solution_vec <- solve(coeff,rhs)
```

Print out this vector of solutions (eg print(solution_vec)). What is the solution for the mean? Which SNP has the largest effect?

Next we want to print GEBV for the selection candidates. This is done with the equation:

$$\mathbf{GEBV} = \mathbf{X}\hat{\mathbf{g}}$$

The g_hat are the solutions for the SNP effects you have just solved. The xvector for the selection candidates is in the file xvec_prog.inp. Can you write a small R script to calculate the GEBV?

Fours years later, all the selection candidates receive a phenotypic record from a progeny test. The results are in the file yvec_prog.inp. What is the correlation between your GEBV and the TBV? (Don't expect this to be to high with only 10 SNPs).

5 Genomic selection using a Bayesian approach

For the first exercise, we will analyse a small data set using the method BayesA of Meuwissen et al. (2003). We will analyse the data with a script written in the R language, meuwissenBayesA.R. The script considers single markers rather than marker haplotypes, but would be easy to extend to haplotypes. The script estimates single marker effects (**g**), a variance for each of these effects (**gvar**), and overall mean **mu** and the error variance (**vare**). A description of the program is given here (descriptions in bold).

R coding of genomic selection from Meuwissen et al. (2001)

Set the number of markers, the number of markers and the number of iterations #

```
nmarkers <- 3      #number of markers
nrecords <- 25     #number of records
numit    <- 1000   #number of iterations
```

The next section reads in the data from two files. The first is the x vector, with 0 for the 1 1 SNP genotype, 1 for 1 2 and 2 for 2 2. The second file is a vector of phenotypic records. Set the path to the location of your files. -

```
x <-
matrix(scan("d:/iowacourse/practicals/day5/smallExample/xvec.inp"),nc
ol=nmarkers,byrow=TRUE)
y <-
matrix(scan("d:/iowacourse/practicals/day5/smallExample/yvec.inp"),by
row=TRUE)
```

Set up some storage vectors and matricies to store parameter values across iterations

```
gStore <- array(0,c(numit,nmarkers))
gvarStore <- array(0,c(numit,nmarkers))
vareStore <- array(0,c(numit))
muStore <- array(0,c(numit))
ittstore <- array(0,c(numit))
```

The Gibbs cycles begin.

Step 1. Initialization of g and mu, declaration of other arrays.

```
g <- array(0.01,c(nmarkers))
mu <- 0.1
gvar <- array(0.1,c(nmarkers))
ones <- array(1,c(nrecords))
e <- array(0,c(nrecords))
```

Begin the iterations

```
for (l in 1:numit) {
```

Step 2. Sample vare from an inverse chi-square posterior

```
      e <- y - x%*%g - mu    # First calculate the vector of residuals
      vare <- (t(e)%*%e)/rchisq(1,nrecords-2)
```

Step 3 Sample the mean from a normal posterior

```
      mu <- rnorm(1,(t(ones)%*%y -
t(ones)%*%x%*%g)/nrecords,sqrt(vare/nrecords))
```

Step 4. Sample the gvar from the inverse chi square posterior

```
      for (j in 1:nmarkers) {

#       gvar[j] <- (0.002+g[j]*g[j])/rchisq(1,4.012+1)  # Meuwissen
#et al. (2001) prior
#       gvar[j] <- (g[j]*g[j])/rchisq(1,1)        # Xu (2003)  #prior
        gvar[j] <- (g[j]*g[j])/rchisq(1,0.998)    # Te Braak et # al.
(2006) prior
      }
```

Step 5 Sample the g from a normal distribution

```
      z <- array(0,c(nrecords))
      for (j in 1:nmarkers) {
       gtemp <- g
       gtemp[j] <- 0
       for (i in 1:nrecords) {
        z[i] <- x[i,j]
       }
       mean <- ( t(z)%*%y-t(z)%*%x%*%gtemp-t(z)%*%ones*mu ) /
(t(z)%*%z+vare/gvar[j])    # Calculating the mean of the distribution
       g[j] <- rnorm(1,mean,sqrt(vare/(t(z)%*%z+vare/gvar[j])))
      }
```

The final step in each iteration is to store the parameter values

```
      for (j in 1:nmarkers) {
        gStore[l,j] <- g[j]
        gvarStore[l,j] <- gvar[j]
      }
      vareStore[l] <- vare
      muStore[l] <- mu
      ittstore[l] <- l
}
```

This is the end of the program.

Consider a data set with three markers. The data set was simulated as: the effect of a 2 allele at the first marker is 3, the effect of a 2 allele at the second marker is 0, and the effect of a 2 allele at the third marker was -2. The mu was 3 and the vare was 1. The data set is:

Animal	Phenotype	Marker1 allele 1	Marker1 allele 2	Marker2 allele 1	Marker 2 allele 2	Marker3 allele 1	Marker 3 allele 2
1	9.68	2	2	2	1	1	1
2	5.69	2	2	2	2	2	2
3	2.29	1	2	2	2	2	2
4	3.42	1	1	2	1	1	1
5	5.92	2	1	1	1	1	1
6	2.82	2	1	2	1	2	2
7	5.07	2	2	2	1	2	2
8	8.92	2	2	2	2	1	1
9	2.4	1	1	2	2	1	2
10	9.01	2	2	2	2	1	1
11	4.24	1	2	1	2	2	1
12	6.35	2	2	1	1	1	2
13	8.92	2	2	1	2	1	1
14	-0.64	1	1	2	2	2	2
15	5.95	2	1	1	1	1	1
16	6.13	1	2	2	1	1	1
17	6.72	2	1	2	1	1	1
18	4.86	1	2	2	1	1	2
19	6.36	2	2	2	2	2	2
20	0.81	1	1	2	1	1	2
21	9.67	2	2	1	2	1	1
22	7.74	2	2	2	1	1	2
23	1.45	1	1	2	2	2	1
24	1.22	1	1	2	1	2	1
25	-0.52	1	1	2	2	2	2

The first step is to make the files yvec.inp and xvec.inp. In the case of yvec.inp, this is simply the list of phenotypes (no headers or identifiers). For xvec.inp, the number of 2 alleles at each marker for each animal, as a 25 x 3 matrix. The first line of this file would be (for animal 1) "2 1 0".

Save these files in a convenient location. Next open the R graphical interface, and open the script "meuwissenBayesA.R". Check the number of markers is set to 3, and the number of records 25. You will have to change the path of the files as well.

Choose a number of iterations, say 1000.

Run the script using the run all command. As the script runs, it stores values for g, gvar, mu and vare for each iteration. After the script has run, you can use the plotting facilities in R to investigate changes in the parameters over iterations.

For example, to look at the effect of the third marker across iterations, you would enter the command

```
> plot(ittstore[1:1000],gStore[1:1000,1])
```

Use this command to investigate each of the parameters in turn, and determine if they appear to be fluctuating about the correct values.

We can also plot the posterior distribution, for example for the effect of the third marker. We would discard the first 100 iterations of the program as "burn in":

```
> plot(density(gStore[100:1000,1]))
```

Does the distribution appear to be normal? What about the distributions of the other parameters?

To get the mean of the distribution, you would type:
mean(gStore[100:1000,1])
Do the means of the parameters agree with the true value of these parameters?

Now a new set of animals (selection candidates without phenotypes) are genotyped for the three markers. Their genotypes are:

Animal	Marker1 allele 1	Marker1 allele 2	Marker2 allele 1	Marker2 allele 2	Marker3 allele 1	Marker3 allele 2	TBV
26	2	2	2	1	2	1	4
27	2	1	1	2	2	1	1
28	1	1	1	2	2	2	-4
29	1	2	2	2	2	1	1
30	1	1	2	2	1	2	-2
31	2	1	1	2	2	1	1
32	2	2	2	2	2	2	2
33	2	2	2	2	1	2	4
34	2	2	2	1	1	2	4
35	1	1	1	2	2	2	-4

Calculate the GEBV for these animals as:

$$\mathbf{GEBV} = \mathbf{X}\hat{\mathbf{g}}$$

What is the correlation with the True breeding values ? (given in the table above, TBV).

Next we will use the script to estimate SNP effects in the reference population in practical 5.6. So you will need to read in the x matrix in xvec_day4.inp, the y vector in yvec_day4.inp. The number of markers in the program will need to be changed to 10 and the number of records to 325.

Run the script.

The next thing you want to do is extract SNP solutions. After the script has run, you can do this by typing:
> mean(gStore[100:1000,1])
This will give you the mean value of the SNP effect for SNP 1 from iterations 100 to 1000 (eg, excluding burn in). So for SNP 6 you would type
>mean(gStore[100:1000,6]).

Compare your SNP solutions from the Bayes program to those from BLUP (practical 5.6). One of the reasons for using the Bayesian approach is to allow different variances of SNP effect across chromosome segments. In particular, the Bayes approach should set some variances (and so SNP effects) to very close to zero. Does this seem to have happened? How many QTL would you say are on the chromosome segment?

Can you predict GEBV for the selection candidates in practical 5.6 using the SNP solutions from the Bayesian approach? Are they more highly correlated with the TBV than the GEBV from the BLUP approach?

Now change the R script to use the prior distribution of chromosome segment variances of effects to that of Meuwissen et al. (2001), eg. $\chi^{-2}(4.012,0.002)$. Now re-run the script. How do the SNP solutions compare with those when the Xu (2003) prior is used? Are the accuracy of the GEBV improved?

6 Bayesian approach a large weight at zero (BayesB)

In this exercise, we will modify the BayesA script from the previous exercise to sample from a prior distribution for the chromosome segment variances with a large weight at zero. This incorporates our prior knowledge that many of the chromosome segments will not contain any QTL with an effect on the quantitative trait.

The prior of the variance of chromosome segment effects is now

$$\sigma^2_{gi} = 0 \text{ with probability } \pi,$$
$$\sigma^2_{gi} \sim \chi^{-2}(\nu, S) \text{ with probability } (1-\pi),$$

Unlike BayesA, the posterior of the variance of chromosome segment effects does not have a known distribution and cannot be sampled directly in the Gibbs chain. We will therefore implement a Metropolis Hastings (MH) step with the Gibbs chain to sample **gvar** and **g** simultaneously.

To modify the code, you will need first specify the number of MH cycles you wish to do:

```
# Parameters
nmarkers <- 10      #number of markers
nrecords <- 325     #number of records
numit    <- 1000   #number of iterations
propSegs <- 0.66    #Prior proportion of segments having a non zero
effect
numMHCycles = 20 # Number of metropolis hastings cycles when sampling
variance of segments
```

The next step is to correct the phenotypic records for all number of MH cycles when sampling the gvar and g (Steps 4 and 5). We will store the corrected records in a vector called ycorr:

```
# Step 4.  Sample the gvar and g using Metropolis Hastings algorithm
(Independance sampling)
      for (j in 1:nmarkers) {

# First correct records for all other effects including mean and
other markers
       gtemp <- g
       gtemp[j] <- 0
       ycorr <- array(0,c(nrecords,1))
       Ival <- array(0,c(nrecords,nrecords))
       for (i in 1:nrecords) {
        ycorr[i] <- y[i] - mu
        Ival[i,i] <- vare
        for (k in 1:nmarkers) {
         ycorr[i] = ycorr[i] - x[i,k]*gtemp[k]
```

```
        }
      }
```

In this step we have also built a matrix which is nrecords x nrecords and has **vare** on the diagonal, as we will need this in the next step.

The next step is to calculate the likelihood of the data given the current gvar, before we sample a new one. The formula for the likelihood is:

$L(\mathbf{y}^* \mid \sigma^2_{gi} = \frac{1}{2\pi^{1/2n} \mid \mathbf{V} \mid^{1/2}} e - 1/2(\mathbf{y}corr' \mathbf{V}^{-1} \mathbf{y}corr)$ where $\mathbf{V} = \mathbf{X}i(\mathbf{I}\sigma^2_{gi})\mathbf{X}i' + \mathbf{I}\sigma^2_{\mathbf{e}})$ and

|V| is the determinant of **V**. In R we can do this calculation as:

```
# Now calculate likelihood with current gvar[j] p(gvar[j]|ycorr)
going into the chain
       V = (x[,j]*gvar[j])%*%t(x[,j])+Ival
       LH1 <- 1/(2*pi^(1/2*nrecords)*sqrt(det(V)))*exp(-
0.5*t(ycorr)%*%ginv(V)%*%ycorr)
```

The ginv function calculates the generalised inverse of V. You will have to load the R package MASS to get this function. (Load packages in the

It is also useful to calculate the likelihood of the data when the gvar is zero, as we will sample gvar=0 many times in the MH cycles.

```
# And likelihood if variance is zero
       V = Ival
       LH0 <-  1/(2*pi^(1/2*nrecords)*sqrt(det(V)))*exp(-
0.5*t(ycorr)%*%ginv(V)%*%ycorr)
```

Now we can run the MH cycles, sampling a new gvar, comparing the likelihood of the data with the new gvar to the old gvar. If the likelihood improves, we will replace the old gvar with the new gvar. If it does not improve, we will replace it with a probability LH(new **gvar**)/LH(old **gvar**). If we do replace gvar, we will also sample the effect of the SNP with the new gvar.

```
       for (kk in 1:numMHCycles) {
        if (runif(1,0,1)<propSegs) {      # sample segment variance
from (1-progSegs)*0 + propSegs*chi-square
# Sample new gvar[j] from driver distribution
         gvar_new <- 1/rchisq(1,4.012)
         V = (x[,j]*gvar_new)%*%t(x[,j])+Ival
         LH2 <- 1/(2*pi^(1/2*nrecords)*sqrt(det(V)))*exp(-
0.5*t(ycorr)%*%ginv(V)%*%ycorr)
```

```
         alpha <- min(LH2/LH1,1) # replace gvar with prob LH(new
#gvar)/LH(old gvar).
         if (runif(1)<alpha) {
# Acceptance
          gvar[j] = gvar_new
          LH1 <- LH2
         }
        }
        else {          # if zero variance sampled
         alpha <- min(LH0/LH1,1)
         if (runif(1)<alpha) {
# Acceptance
          gvar[j] = 0
          LH1 <- LH0
         }
        }
       }
       if (gvar[j]>0) {
        meanval <- ( t(x[,j])%*%y-t(x[,j])%*%x%*%gtemp-
t(x[,j])%*%ones*mu ) / (t(x[,j])%*%x[,j]+(vare)/gvar[j])
        g[j] <-
rnorm(1,meanval,sqrt((vare)/(t(x[,j])%*%x[,j]+(vare)/gvar[j])))
       }
       else {
        g[j] <-0
       }
      }
```

Once you have finished coding the method, save your R script as a new file (BayesB.R for example).

Now run the script with the small data set from practical 5.7 (3 markers and 25 records) Use 20 MH cycles. Look at the values sampled for each of 3 segments across the Gibbs chain. Do any of the **g** get set consistently to zero? Now choose different values for the proportion of segments set to zero and the parameters of the inverse chi square parameters where gvar new is sampled from (both these for the prior of the gvar). How sensitive are the results to the parameters of the prior distribution of the variances of chromosome segment effects?