Module 6: Association Practical

Need help? doug.speed@ucl.ac.uk

Check List

If all has gone to plan you should:

- Have `putty` installed and configured, which allows you to "ssh" into Hong's server, hong1.une.edu.au (username asc2016, pw feb01). This is where you will run `PLINK, GCTA, LDAK and IMPUTE2`
- Have `winSCP` installed and configured, which allows you to copy files from Hong's server to your Desktop (which you can then read into R), and files from your Desktop to Hong's server
- Have a copy of the slides, either from Julius' website `http://jvanderw.une.edu.au/AGSCcourse.htm`, or from the module6 folder on Hong's server

If so, you may now begin :)

Data Files

In addition to the files provided for this practical, we will use `genofinal.bed`, `genofinal.bim` and `genofinal.fam`, the files we constructed at the end of the last practical

1 - Single-SNP Association

`phen.pheno` contains three phenotypes. Perform linear regression for Phenotype 1 (--linear), including sex as a covariates, and save with the prefix `linear1`. Perform logistic regression for Phenotype 2 (--logistic) and save with the prefix `logistic2`. In both cases, add --ci .95 so that `PLINK` outputs SDs.

Make a manhattan for each set of results

Phenotype 3 almost exactly matches Phenotype 2; analyse this using linear regression and compare the (-log10) *p*-values from this analysis to those from logistic regression of Phenotype 2

1 - Single-SNP Association

```
../plink --linear hide-covar --bfile genofinal --out linear1 \
--no-sex --covar all.sex --pheno phen.pheno --mpheno 1 --ci .95
../plink --logistic hide-covar --bfile genofinal --out logistic2 \
--no-sex --covar all.sex --pheno phen.pheno --mpheno 2 --ci .95
../plink --linear hide-covar --bfile genofinal --out linear3 \
--no-sex --covar all.sex --pheno phen.pheno --mpheno 3 --ci .95
```

Could also test the autosomal SNPs using

```
../plink --assoc --bfile genofinal --out linear1 --no-sex \
--covar all.sex --pheno phen.pheno --mpheno 1--autosome --ci .95
```

```
> res1=as.matrix(read.table("linear1.assoc.linear",head=T))
> pva1=as.numeric(res1[,12])
> chr=as.numeric(res1[,1])
> plot(-log10(pva1))
```

1 - Single-SNP Association

```
> allchr=sort(unique(chr))
> marks=array(0,length(allchr)+1)
> for(i in 1:length(allchr)){aa=sum(chr==allchr[i]);marks[i+1]=marks[i]+aa}
> marks2=.5*marks[-1]+.5*marks[-length(marks)]

> plot(-log10(pva1),col=chr%%2+2,axes=F,xlab="Chromosome",ylab="-log10(P)")
> axis(1,at=marks,lab=rep("",4))
> axis(1,at=marks2,lab=allchr,tick=F)
> axis(2)

> res2=as.matrix(read.table("logistic2.assoc.logistic",head=T))
> pva2=as.numeric(res2[,12])
> plot(-log10(pva2),col=chr%%2+2,axes=F,xlab="Chromosome",ylab="-log10(P)")
> axis(1,at=marks,lab=rep("",4));axis(1,at=marks2,lab=allchr,tick=F);axis(2)

> res3=as.matrix(read.table("linear3.assoc.linear",head=T))
> pva3=as.numeric(res3[,12])
> plot(-log10(pva3),col=chr%%2+2,axes=F,xlab="Chromosome",ylab="-log10(P)")
> axis(1,at=marks,lab=rep("",4));axis(1,at=marks2,lab=allchr,tick=F);axis(2)

> plot(-log10(pva2),-log10(pva3),xlab="Logistic -log10(P)",ylab="Linear -log10(P)")
> abline(a=0,b=1,col=2,lty=3,lwd=3)
```

2 - Significance Threshold

Using Bonferroni correction, calculate the significance threshold corresponding to FWER 5% (use `nrow(res1)` or `length(pva1)` to get the number of SNPs). How many SNPs exceed this threshold for Phenotypes 1 and 2

`phen.effects` contains the true causal SNPs for each phenotype. How did these SNPs fare?

2 - Significance Threshold

```
> plot(-log10(pva1),col=chr%%2+2,axes=F,xlab="Chromosome",ylab="-log10(P)")
> axis(1,at=marks,lab=rep("",4))
> axis(1,at=marks2,lab=allchr,tick=F)
> axis(2)
> sig=.05/nrow(res1)
> abline(h=-log10(sig),lwd=3,lty=2,col=4)

> allpva=cbind(pva1,pva2,pva3)
> apply(allpva,2,function(x)sum(x<sig,na.rm=T))

> plot(-log10(pva1),col=chr%%2+2,axes=F,xlab="Chromosome",ylab="-log10(P)")
> axis(1,at=marks,lab=rep("",4))
> axis(1,at=marks2,lab=allchr,tick=F)
> axis(2)

> eff=as.matrix(read.table("phen.effects",head=T))
> mm=match(eff[1:3,2],res1[,2])
> lines(mm,-log10(pva1[mm]),pch=19,cex=2,type="p")
```

3 - QQ Plot and GIFs

Construct a QQ plot for Phenotype 1, comparing expected $\chi^2(1)$ quantiles with those observed. Hint: if you had ten SNPs, the expected quantiles would be `qchisq(1:10/11,1)`

Calculate the genomic inflation factor for Phenotype 1; the GIF measures how much higher the observed median test statistic is than the expected test statistic. Hint: the median $\chi^2(1)$ statistic is `qchisq(.5,1)`

3 - QQ Plot and GIFs

```
> chi1=qchisq(pva1,1,lower=F)
> gif1=median(chi1,na.rm=T)/qchisq(.5,1)

> use=which(!is.na(chi1))
> sort1=sort(chi1[use])
> N1=length(use)
> exp1=qchisq(1:N1/(N1+1),1,lower=T)
> plot(exp1,sort1,pch=19,xlab="Expected Chisq",ylab="Observed Chisq")
> abline(a=0,b=1,col=2,lty=2,lwd=3)
> abline(v=qchisq(.5,1),col=4)
> text(4,100,paste("GIF:",round(gif1,2)),cex=2)

> plot(exp1,sort1,pch=19,xlab="Expected Chisq",ylab="Observed Chisq",
xlim=c(0,1),ylim=c(0,1))
> abline(a=0,b=1,col=2,lty=2,lwd=3)
> abline(v=qchisq(.5,1),col=4)

> aa=which(sort1<8)
> plot(exp1,sort1,xlab=expression(paste("Exp. ",chi^2,"(1)",sep="")),
ylab=expression(paste("Observed ",chi^2,"(1)",sep="")),type="n")
> abline(a=0,b=1,col=2,lty=2,lwd=3)
> lines(exp1[aa],sort1[aa],lwd=3.5)
> lines(exp1[-aa],sort1[-aa],type="p",pch=19,cex=.5)
```

4 - Power

Write a function which returns the detection power (probability the p-value exceeds `sig`) of a SNP whose variance explained is `h2`. Hint: in R, the function would take the form

```
get_power=function(h2,n,sig)      # <- arguments go in here
{
# <- here are the operations to complete the function
# (i.e., those required to compute pow from h2, n and sig)
return(pow)      # <- don't forget to return the value
}
```

What chance did we have of detecting a variable with $h^2 = 0.1$

Repeat assuming the correlation squared between the causal SNP and best tag was 0.8. 0.5 ,0.2

4 - Power

```
> get_power=function(h2,n,sig)
{
ncp=n*h2/(1-h2)
pow=pchisq(qchisq(sig,1,lower=F),1,ncp=ncp,lower=F)
return(pow)
}

> h2s=1:200/1000
> pows=get_power(h2s,253,sig)
> plot(h2s,pows,type="l",xlab="Variance Explained",ylab="Power",lwd=3)

> tags=c(.8,.5,.2)
> for(tag in tags)
{
powb=get_power(h2s*tag,253,sig)
lines(h2s,powb,col=match(tag,tags)+1,lwd=3)
}
> legend(.015,.9,leg=c("100%","80%","50%","20%"),
title="Tagging",col=1:4,lwd=3)
```

5 - Replication

`logistic4.assoc.logistic` contains results from an independent study of Phenotype 2. For the significant SNPs in Study 2, what are their *p*-values in this study

Get 95% confidence intervals for log(OR)

When using `PLINK` for linear regression, the output file reports estimated effect size and SD for this. However, for logistic regression, the output file reports estimated odds ratio (OR) and the SD for log(OR).

To construct a confidence interval, note that for a normal distribution, 95% of values lie within 1.96 SDs of the mean

Based on these *p*-values and confidence intervals, do any of the significant SNPs from Study 2 replicate in Study 4?

5 - Replication

```
> top=which(pva2<sig)
> res4=as.matrix(read.table("logistic4.assoc.logistic",head=T))
> pva4=as.numeric(res4[,12])
> pva4[top]


#extract the odds ratios and sds (of log odds) for top snps
> ora=as.numeric(res2[top,7])
> sda=as.numeric(res2[top,8])
> orb=as.numeric(res4[top,7])
> sdb=as.numeric(res4[top,8])

#from these we can construct conf ints for log odds ratios
> cia=cbind(log(ora)-1.96*sda,log(ora)+1.96*sda)
> cib=cbind(log(orb)-1.96*sdb,log(orb)+1.96*sdb)
```

5 - Replication

```
> plot(1:2,xlim=c(-2,2),xlab="Log Odds Ratio",
> ylab="",type="n",axes=F)
> axis(1)
> axis(2,at=c(1.25,1.75),lab=res2[top,2],tick=F,padj=-2)#
> abline(h=1.5,lty=2,col=4,lwd=3)
> abline(v=0,lty=2,lwd=3)

> rect(cia[1,1],1.8,cia[1,2],1.9,col=2)
> segments(log(ora[1]),1.75,log(ora[1]),1.95,lwd=3)
> rect(cib[1,1],1.6,cib[1,2],1.7,col=3)
> segments(log(orb[1]),1.55,log(orb[1]),1.75,lwd=3)

> rect(cia[2,1],1.3,cia[2,2],1.4,col=2)
> segments(log(ora[2]),1.25,log(ora[2]),1.45,lwd=3)
> rect(cib[2,1],1.1,cib[2,2],1.2,col=3)
> segments(log(orb[2]),1.05,log(orb[2]),1.25,lwd=3)

> legend(0,1.6,fill=2:3,leg=c("Study 2","Study 3"),cex=1.5)
```

6 - Meta-Analysis

Perform a fixed-effect meta-analysis of the results from Studies 2 and 4. The standard approach is inverse-variance weightings

Suppose for SNP rs123, the estimates of the effect size are e_2=2 and e_4=4, with SDs s_2=1 and s_4=2, respectively. Then we weight Studies 2 & 4 in the ratio $1^2 : 1/2^2 = 1 : 1/4$; i.e., we give Study 2 four times the weight of Study 4. (Note, typically, the relative weightings align closely with sample size)

The weights must add to one, and therefore we will use weightings w_2=1/(1+1/4)=.8 and w_4=1/4/(1+1/4)=.2

Recall that if a r.v. X has variance 4, then $2X$ will have variance $2^2 \times 4$. Therefore, the weighted estimate $w_2 e_2 + w_4 e_4$ will have variance $w_2^2 s_2^2 + w_4^2 s_4^2 = .8^2 \times 1 + .2^2 \times 2^2 = 0.8$

A general expression for the variance is $1/(1/s_2^2 + 1/s_4^2)$

6 - Meta-Analysis

```
> orsa=as.numeric(res2[,7])
> sdsa=as.numeric(res2[,8])
> orsb=as.numeric(res4[,7])
> sdsb=as.numeric(res4[,8])

> metalog=array(NA,nrow(res2))
> metasd=array(NA,nrow(res2))
> for(j in 1:nrow(res2))
{
ss=1/sdsa[j]^2+1/sdsb[j]^2
wa=1/sdsa[j]^2/ss;wb=1/sdsb[j]^2/ss
metalog[j]=wa*log(orsa[j])+wb*log(orsb[j])
metasd[j]=1/ss^.5
}
> metawald=metalog/metasd
> metapva=pchisq(metawald^2,1,lower=F)
```

Meta-analysis available in old PLINK (Version 0.9)

```
plink --meta-analysis logistic2.assoc.logistic logistic4.assoc.logistic
```

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