Module 10: SNP-Based Heritability Analysis 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 - NOT AS IMPORTANT IN THIS ONE
- Have a copy of the slides, either from Julius' website `http://jvanderw.une.edu.au/AGSCcourse.htm`, or from the module11 folder on Hong's server

If so, you may now begin :)

Running LDAK and GCTA

Both LDAK and GCTA are designed to "look like" PLINK, so share many options (e.g., `--bfile`, `--extract`, `--keep`, `--chr`). Note that, in LDAK, arguments ALWAYS come in PAIRS

They are both designed to run in `UNIX`

Both LDAK and GCTA are "installed" on Hong's server, so can be run from your home directory by typing `../ldak.out` or `../gcta64`

LDAK help pages are at `www.ldak.org`

GCTA help pages are at `http://cnsgenomics.com/software/gcta/`

Datafiles

Again, you should run all commands from YOUR HOME FOLDER

```
cd
cd XXX
cd $HOME/XXX
```

The final command of the PLINK practical was

```
../plink --bfile genonew --make-bed --keep keeppop.fam \
--hwe 1e-3 --maf 0.01 --geno 0.01 --out genofinal
```

We will use the output. So you can either run this yourself, else copy the required files from the module10 directory. We will also use phen.pheno

```
cd $HOME/XXX
cp ../module10/genofinal.{bed,bim,fam} ./
cp ../module10/phen.pheno ./
```

Check you can run LDAK and GCTA

`../ldak.out; ../gcta64`

asc2016@hong1:~/doug

File Edit View Search Terminal Help

```
[asc2016@hong1 doug]$ ../ldak.out

LDAK - Software for obtaining Linkage Disequilibrium Adjusted Kinship estimates and Loads More
Help pages at http://dougspeed.com/ldak

Arguments:

At least one pair of arguments is required, including ONE main argument

Main argument must be one of:

--cut-weights <folder> (requires bfile/chiamo/beagle/sp/sped) - divide genome into sections for calculating weights
--calc-weights <folder> (requires bfile/chiamo/beagle/sp/sped/speed and section) - calculate weights for section specified
--join-weights <folder> - join up weights
--thin <output> - thin predictors

--cut-kins <folder> (requires bfile/chiamo/beagle/sp/sped) - divide genome into partitions for calculating kinships
--calc-kins <folder> (requires bfile/chiamo/beagle/sp/sped/speed, weights and region) - calculate kinship for partition specified
--join-kins <folder> - join up kinships
--calc-kins-direct <output> (requires bfile/chiamo/beagle/sp/sped/speed, weights) - calculate kinship in one step

--add-grm <output> (requires mgrm) - add kinships
--sub-grm <output> (requires mgrm) - subtract kinships
--convert-gz <output> (requires grm) - convert kinships from gz format
--convert-raw <output> (requires grm) - convert kinships from raw format

--cut-genes <folder> (requires bfile/chiamo/beagle/sp/sped/speed, genefile/chunks/chunks-bp and weights) - calculate breakpoints and divide genome into partitions
--calc-genes-kins <folder> (requires bfile/chiamo/beagle/sp/sped/speed, weights and partition) - calculate kinships for each gene in partition specified
--calc-genes-reml <folder> (requires bfile/chiamo/beagle/sp/sped/speed, partition and weights) - performs regression for each gene in partition specified
--join-genes-reml <folder> - joins together regression results

--decompose <output> (requires grm) - eigen-decompose ready for reml
--reml <output> (requires pheno) - performs generalised reml analysis
--he <output> (requires grm, pheno) - performs Haseman Elston analysis
--calc-blups <output> (requires remlfile or blupfile) - calculates blups for reml

--calc-scores <output> - calculate one or more genetic risk profiles
--calc-pca-loads <output> - calculate one or more genetic risk profiles

--make-phenos <output> (requires bfile/chiamo/sp/sped/speed and her) - simulate phenotypes
--make-snps <output> (requires num-samples and num-snps) - simulate genotypes
--pca <output (requires grm) - compute top principal component axes
--filter <output (requires grm) - filter out related individuals
```

1 - Compute Unweighted Kinships across All SNPs

In LDAK:

```
../ldak.out --calc-kins-direct kin12X --bfile genofinal \
--ignore-weights YES --kinship-raw YES --kinship-gz YES
```

By default, kinships are saved in a `.grm.bin` file (binary format);

the option `--kinship-raw YES` tells LDAK to also save a copy to `.grm.raw` (raw text)

the option `--kinship-gz YES` tells LDAK to also save a copy to `.grm.raw` (raw text)

```
less -S kin12X.grm.raw
less -S kin12X.grm.gz
```

```
> kina=as.matrix(read.table("kin12X.grm.raw"))
> up=which(upper.tri(kina,diag=T))
> hist(kina[up],n=50,ylim=c(0,100))
```

1 - Compute Unweighted Kinships across All SNPs

In GCTA:

```
../gcta64 --make-grm --out kingcta --bfile genofinal \
--autosome-num 24
```

Note, GCTA treats Chr X in a more sophisticated fashion; so here, we are tricking it into ignoring that a sex chromosome has been included (for Chr X should instead use `--make-grm-xchr`)
http://cnsgenomics.com/software/gcta/estimate_grm.html

```
less -S kingcta.grm.gz
```

```
../ldak.out --calc-kins-direct kingctab --bfile genofinal \
--ignore-weights YES --kinship-gz YES --maf-stand YES
```

```
less -S kingctab.grm.gz
```

2 - Compute Kinships for each Chromosome Separately

```
../ldak.out --calc-kins-direct kin1 --bfile genofinal \
--ignore-weights YES --chr 1 --kinship-raw YES

../ldak.out --calc-kins-direct kin2 --bfile genofinal \
--ignore-weights YES --chr 2 --kinship-raw YES

../ldak.out --calc-kins-direct kinX --bfile genofinal \
--ignore-weights YES --chr 23 --kinship-raw YES

> kin1=as.matrix(read.table("kin1.grm.raw"))
> kin2=as.matrix(read.table("kin2.grm.raw"))
> kinX=as.matrix(read.table("kinX.grm.raw"))

> par(mfrow=c(1,2))
> plot(kin1[up],kin2[up])
> plot(kin1[up],kinX[up])
```

2b - Join these Three Kinships

```
echo "kin1
kin2
kinX" > mlist.txt
../ldak.out --add-grm kincheck --mgrm mlist.txt \
--kinship-raw YES --kinship-gz YES
```

Can check the kinships kin12X and kincheck match

```
less -S kin12X.grm.raw
less -S kincheck.grm.raw
```

```
> kinb=as.matrix(read.table("kincheck.grm.raw"))
> plot(kina[up],kinb[up])
```

2c - Alternative Approach

```
../ldak.out --cut-kins partitions --bfile genofinal \
--by-chr YES

../ldak.out --calc-kins partitions --bfile genofinal \
--ignore-weights YES --partition 1
../ldak.out --calc-kins partitions --bfile genofinal \
--ignore-weights YES --partition 2
../ldak.out --calc-kins partitions --bfile genofinal \
--ignore-weights YES --partition 3

../ldak.out --join-kins partitions --kinship-raw YES

less -S kin12X.grm.raw
less -S partitions/kinsh
```

LDAK is highly memory efficient, but this option is useful for genomic partitioning

3 - Calculated LD-Adjusted Kinships

First you have to "cut" the genome into sections

```
../ldak.out --cut-weights sections --bfile genofinal
```

This gives an error because the data are so small, but suggests how to fix

```
../ldak.out --cut-weights sections --bfile genofinal \
--section-length 200
```

```
cat sections/section_details.txt
cat sections/section_number
```

3 - Calculated LD-Adjusted Kinships

Then you compute weights for each sections

```
../ldak.out --calc-weights sections --bfile genofinal --section 1
../ldak.out --calc-weights sections --bfile genofinal --section 2
../ldak.out --calc-weights sections --bfile genofinal --section 3
```

Could type this for each of the (here 26) sections, else write a loop

```
for section in {4..26};
do
../ldak.out --calc-weights sections --bfile genofinal \
--section $section;
done
```

3 - Calculated LD-Adjusted Kinships

Finally, you join weights across sections

```
../ldak.out --join-weights sections
```

Now you can use these to compute LD-adjusted kinships

```
../ldak.out --calc-kins-direct kinweight --bfile genofinal \
--weights sections/weightsALL --kinship-raw YES
```

```
> kinw=as.matrix(read.table("kinweight.grm.raw"))
> plot(kina[up],kinw[up])
> abline(a=0,b=1,col=2,lty=2,lwd=3)
```

4 - Principal Component Analysis

```
../ldak.out --pca kin12X --grm kin12X --axes 5

../gcta64 --pca 5 --grm kin12X --out check

../ldak.out --pca kinweight --grm kinweight --axes 5
```

```
> pcaa=as.matrix(read.table("kin12X.vect"))
> pcaw=as.matrix(read.table("kinweight.vect"))

> plot(pcaa[,3:4],xlab="PCA 1",ylab="PCA 2",main="Unweighted")
> plot(pcaw[,3:4],xlab="PCA 1",ylab="PCA 2",main="Weighted")
```

kin12X.vect is already in correct form for a covariate file

5 - Estimating Heritability

```
../ldak.out --reml out12X --grm kin12X --covar kin12X.vect \
--pheno phen.pheno --mpheno 1

../gcta64 --reml --grm kin12X --qcovar kin12X.vect \
--pheno phen.pheno --out check

ls out12X*
out12X.fixed  out12X.indi.blp  out12X.indi.res
out12X.progress  out12X.reml  out12X.share  out12X.vars
```

5 - Results from LDAK

```
[asc2016@hong1 doug]$ cat out12X.reml
Num_Kinships 1
Num_Regions 0
Num_Covars 6
Blupfile out12X.indi.blp
Regfile none
Fixfile out12X.fixed
Total_Samples 236
With_Phenotypes 236
Null_Likelihood -313.846282
Alt_Likelihood -308.901268
LRT_Stat 9.8900
LRT_P 8.3088e-04
Component Heritability Her_SD Size Intensity Int_SD
Her_K1 0.597873 0.154412 5415.00 11.041055 2.851569
Her_All 0.597873 0.154412 5415.00 11.041055 2.851569
```

5 - Results from GCTA

```
[asc2016@hong1 doug]$ cat check.hsq
Source Variance SE
V(G) 0.544554 0.169955
V(e) 0.366131 0.130599
Vp 0.910685 0.092290
V(G)/Vp 0.597961 0.154373
logL -97.545
logL0 -102.490
LRT 9.890
df 1
Pval 0.0008309
n 236
```

5 - Estimating Heritability - Two Steps

```
../ldak.out --reml out12X --grm kin12X --covar kin12X.vect \
--pheno phen.pheno --mpheno 1

../gcta64 --reml --grm kin12X --qcovar kin12X.vect \
--pheno phen.pheno --out check


Performing eigen-decomposition of kinship matrix
Consider performing the analysis in two steps:
first add "--eigen-save <eigenstem>" to perform just the eigen-deco
then add "--eigen <eigenstem>" to recall this eigen-decomposition a

../ldak.out --reml check2 --grm kin12X --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --eigen-save kin12X

../ldak.out --reml check2 --grm kin12X --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --eigen kin12X
```

5 - Estimating Heritability - Restarting

```
../ldak.out --reml out12X --grm kin12X --covar kin12X.vect \
--pheno phen.pheno --mpheno 1


[asc2016@hong1 doug]$ cat out12X.progress
Iter Her_K1 Likelihood Difference Target Num_Constrained
Start 0.50000 -309.044854 n/a 0.000100 0
1 0.56775 -308.915657 0.129197 0.000100 0
2 0.59018 -308.902243 0.013414 0.000100 0
3 0.59613 -308.901322 0.000921 0.000100 0
4 0.59755 -308.901270 0.000052 0.000100 0

../ldak.out --reml checkb --grm kin12X --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --starts out12X.progress

Will resume analysis at Iteration 4 with heritabilities:
0.597550
```

5 - Testing for Inflation

```
#Compute SNP her of Whole Genome
../ldak.out --reml out12X --grm kin12X --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --constrain NO

#Compute SNP her of Chromosome 1
../ldak.out --reml out1 --grm kin1 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --constrain NO

#Compute SNP her of Chromosome 2
../ldak.out --reml out2 --grm kin2 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --constrain NO

#Compute SNP her of Chromosome X
../ldak.out --reml outX --grm kinX --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --constrain NO

#Compare Total with Sum of Individuals
grep Her_All out{12X,1,2,X}.reml

../ldak.out --reml outmulti --mgrm mlist.txt --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --constrain NO
grep Her_All out{multi,1,2,X}.reml
```

`constrain NO` is useful when you have small heritabilities

6 - Filtering Relatedness

```
../ldak.out --filter kin12X --grm kin12X
```

```
Will be filtering samples so no pairs remain with kinship above 0.131554
(the absolute value of the minimum observed kinship)
Use "--maxrel" to change this threshold

Remove 1416 NA10835 (has kinship 0.590267 with 1416 NA12249); Phenotypes -9999.0 and -9999.0
Remove 1459 NA12864 (has kinship 0.580967 with 1459 NA12873); Phenotypes -9999.0 and -9999.0
Remove 13292 NA07014 (has kinship 0.580824 with 13292 NA07051); Phenotypes -9999.0 and -9999.0
Remove 13281 NA12348 (has kinship 0.570592 with 13281 NA12344); Phenotypes -9999.0 and -9999.0
Remove 1346 NA12045 (has kinship 0.547988 with 1346 NA10852); Phenotypes -9999.0 and -9999.0
Remove 1330 NA12341 (has kinship 0.547784 with 1330 NA12335); Phenotypes -9999.0 and -9999.0
Remove 1459 NA12874 (has kinship 0.546068 with 1459 NA12865); Phenotypes -9999.0 and -9999.0
```

```
../ldak.out --filter kin12Xb --grm kin12X --maxrel 0.05
```

Do not rely on standard relatedness thresholds!

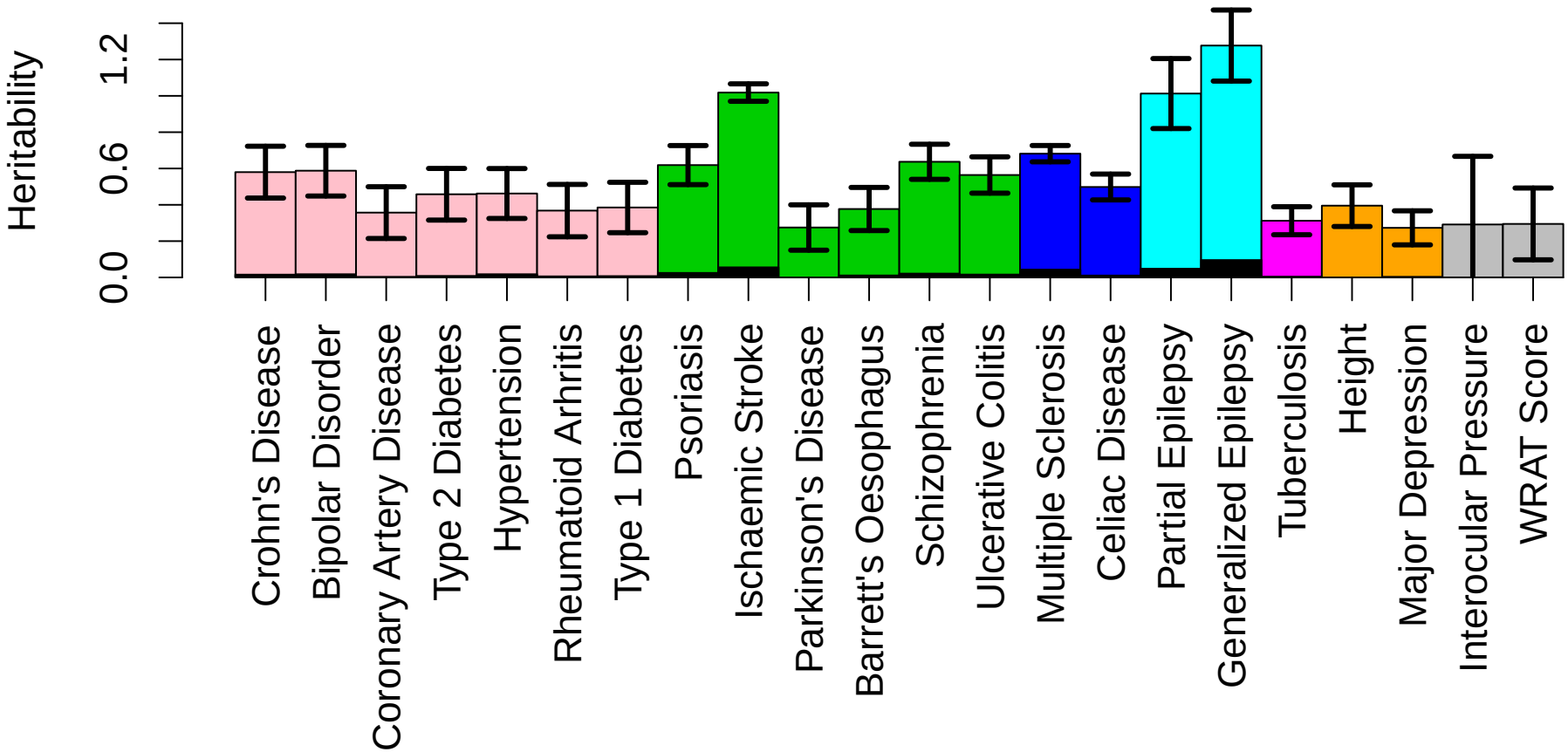
If you provide a phenotype file (`--pheno`) LDAK will selectively filter, excluding individuals missing phenotypes when possible

In GCTA, use `--grm-cutoff` (e.g., `--grm-cutoff 0.05`)

7 - Estimating Heritability using Unrelateds

```
../ldak.out --reml new12X --grm kin12X --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --keep kin12X.keep


#Compute SNP her of Chromosome 1
../ldak.out --reml new1 --grm kin1 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --constrain NO --keep kin12X.keep

#Compute SNP her of Chromosome 2
../ldak.out --reml new2 --grm kin2 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --constrain NO --keep kin12X.keep

#Compute SNP her of Chromosome X
../ldak.out --reml newX --grm kinX --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --constrain NO --keep kin12X.keep

#Compare Total with Sum of Individuals
grep Her_All new{12X,1,2,X}.reml
```

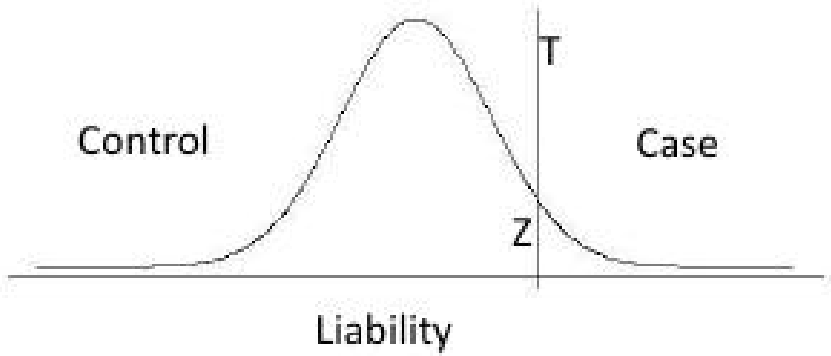
7 - Estimating Heritability using Unrelateds



8 - Binary Traits

We only observe whether L above or below a threshold T

T determined by disease prevalence K



If we knew the liability L, we could fit $L \sim \mathbb{N}(\alpha, K\sigma_L^2 + I\sigma_e^2)$ and estimate $h^2_{Liab} = \sigma^2_L / Var(L)$, the heritability estimate on the liability scale directly

But with L unknown, we instead analyse the phenotype pretending it is continuous, then use the following transformation:

$$h^2_{Liab} = h^2_{SNP} \frac{K^2(1-K)^2}{P(1-P)z^2}$$

where K is the prevalence, P the ascertainment, and z is the "height of the standard normal distribution" at the liability threshold, T

8 - Binary Traits

```
../ldak.out --reml bin12X --grm kin12X --covar kin12X.vect \
--pheno phen.pheno --mpheno 2 --keep kin12X.keep --prevalence .5

../gcta64 --reml --grm kin12X --qcovar kin12X.vect  --out checkc \
--pheno phen.pheno --mpheno 2 --keep kin12X.keep --prevalence .5


[asc2016@hong1 doug]$ cat bin12X.reml.liab
Num_Kinships 1
Num_Regions 0
Num_Covars 6
Blupfile bin12X.indi.blp
Regfile none
Fixfile bin12X.fixed
Total_Samples 169
With_Phenotypes 169
Null_Likelihood -118.584701
Alt_Likelihood -118.451885
LRT_Stat 0.2656
LRT_P 3.0314e-01
Component Heritability Her_SD Size Intensity Int_SD
Her_K1 0.240540 0.474129 5415.00 4.442107 8.755846
Her_All 0.240540 0.474129 5415.00 4.442107 8.755846
```

9 - Different Standardizations

LDAK Default (ignoring weightings):

$$K_{ij} = \frac{1}{N}\sum_{j=1}^{N}(X_{ij} - mean(X_j))(X_{ij} - mean(X_j)) \Big/ Var(X_j)$$

LDAK Default (with weightings):

$$K_{ij} = \frac{1}{\sum w_j}\sum_{j=1}^{N} w_j(X_{ij} - mean(X_j))(X_{ij} - mean(X_j)) \Big/ Var(X_j)$$

GCTA:

$$K_{ij} = \frac{1}{N}\sum_{j=1}^{N}(X_{ij} - 2p_j)(X_{ij} - 2p_j) \Big/ 2p_j(1 - 2p_j)$$

Habier / van Raden:

$$K_{ij} = \frac{1}{\sum_j 2p_j(1 - 2p_j)}\sum_{j=1}^{N}(X_{ij} - 2p_j)(X_{ij} - 2p_j)$$

9 - Different Standardizations

A general framework:

1) $$K_{ij} = \frac{1}{\sum_{w_j}} \sum w_j (X_{ij} - mean(X_j))(X_{ij} - mean(X_j)) \times [Var(X_j)]^{\alpha}$$

2) $$K_{ij} = \frac{1}{\sum_{w_j}} \sum w_j (X_{ij} - 2p_j)(X_{ij} - 2p_j) \times [2p_j(1-p_j)]^{\alpha}$$

LDAK Default is 1) with $\alpha = -1$ (possibly with $w_j = 1$)

GCTA (and default in human genetics) is 2) with $\alpha = -1$ and $w_j = 1$

In animal and plant breeding, the default is 2) with $\alpha = 0$ and $w_j = 1$

(Not exactly correct, but global scaling of K is irrelevent)

In my experience, choice of 1) or 2) is unimportant

Choice of weightings or α much more so

9 - Different Standardizations

```
../ldak.out --calc-kins-direct kinpower-1 --bfile genofinal \
--ignore-weights YES --kinship-raw YES --power -1

../ldak.out --calc-kins-direct kinpower-.5 --bfile genofinal \
--ignore-weights YES --kinship-raw YES --power -.5

../ldak.out --calc-kins-direct kinpower0 --bfile genofinal \
--ignore-weights YES --kinship-raw YES --power 0

> kina=as.matrix(read.table("kinpower-1.grm.raw"))
> kinb=as.matrix(read.table("kinpower-.5.grm.raw"))
> kinc=as.matrix(read.table("kinpower0.grm.raw"))

> plot(kina[up],kinc[up],xlab="Power -1",ylab="Power 0")
> abline(a=0,b=1,col=2,lty=2,lwd=3)
> plot(kina[up],kinc[up],xlab="Power -1",ylab="Power 0",
xlim=c(-.1,.1),ylim=c(-.1,.1))
> abline(a=0,b=1,col=2,lty=2,lwd=3)
```

9 - Different Standardizations

With related individuals

```
../ldak.out --reml power-1 --grm kinpower-1 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1

../ldak.out --reml power-.5 --grm kinpower-.5 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1

../ldak.out --reml power0 --grm kinpower0 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1

grep Her_A power{-1,-.5,0}.reml
```

9 - Different Standardizations

With unrelated individuals

```
../ldak.out --reml bower-1 --grm kinpower-1 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --keep kin12X.keep

../ldak.out --reml bower-.5 --grm kinpower-.5 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --keep kin12X.keep

../ldak.out --reml bower0 --grm kinpower0 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --keep kin12X.keep

grep Her_A bower{-1,-.5,0}.reml
```

9 - SCALING IS IMPORTANT!!!

The familiar form, $h^2 = \sigma_g^2/(\sigma_g^2 + \sigma_e^2)$, is only true for particular **K** (those with average value zero and average diagonal value 1

$$\mathbb{E}(h^2) \approx 1 - \frac{\mathbb{E}(V_R)}{\mathbb{E}(V_T)}$$

$$= \frac{1 - \left(1 - \frac{1}{n}\right)\sigma_e^2}{\left(\frac{\text{trace}(\mathbf{ZZ}^T)}{n}\sigma_g^2 - \frac{\text{sum}(\mathbf{ZZ}^T)}{n^2}\sigma_g^2 + \left(1 - \frac{1}{n}\right)\sigma_e^2\right)}.$$

(Equation B1)

With the usual column standardization, the genotype matrix has trace n and sum 0. Therefore, the heritability estimate will take the simpler form

$$\hat{h}^2 = 1 - \frac{\left(1 - \frac{1}{n}\right)\hat{\sigma}_e^2}{\left(\hat{\sigma}_g + \left(1 - \frac{1}{n}\right)\hat{\sigma}_e^2\right)} = \frac{\hat{\sigma}_g^2}{\hat{\sigma}_g^2 + \left(1 - \frac{1}{n}\right)\hat{\sigma}_e^2},$$

Improved Heritability Estimation, Speed et al, AJHG (2012)

9 - SCALING IS IMPORTANT!!!

By default, LDAK scales kinships to have mean zero and average diagonal one

```
../ldak.out --reml dower-1 --grm kinpower-1 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --keep kin12X.keep --stand-kinship NO

../ldak.out --reml dower-.5 --grm kinpower-.5 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --keep kin12X.keep --stand-kinship NO

../ldak.out --reml dower0 --grm kinpower0 --covar kin12X.vect \
--pheno phen.pheno --mpheno 1 --keep kin12X.keep --stand-kinship NO

grep Her_A dower{-1,-.5,0}.reml
```

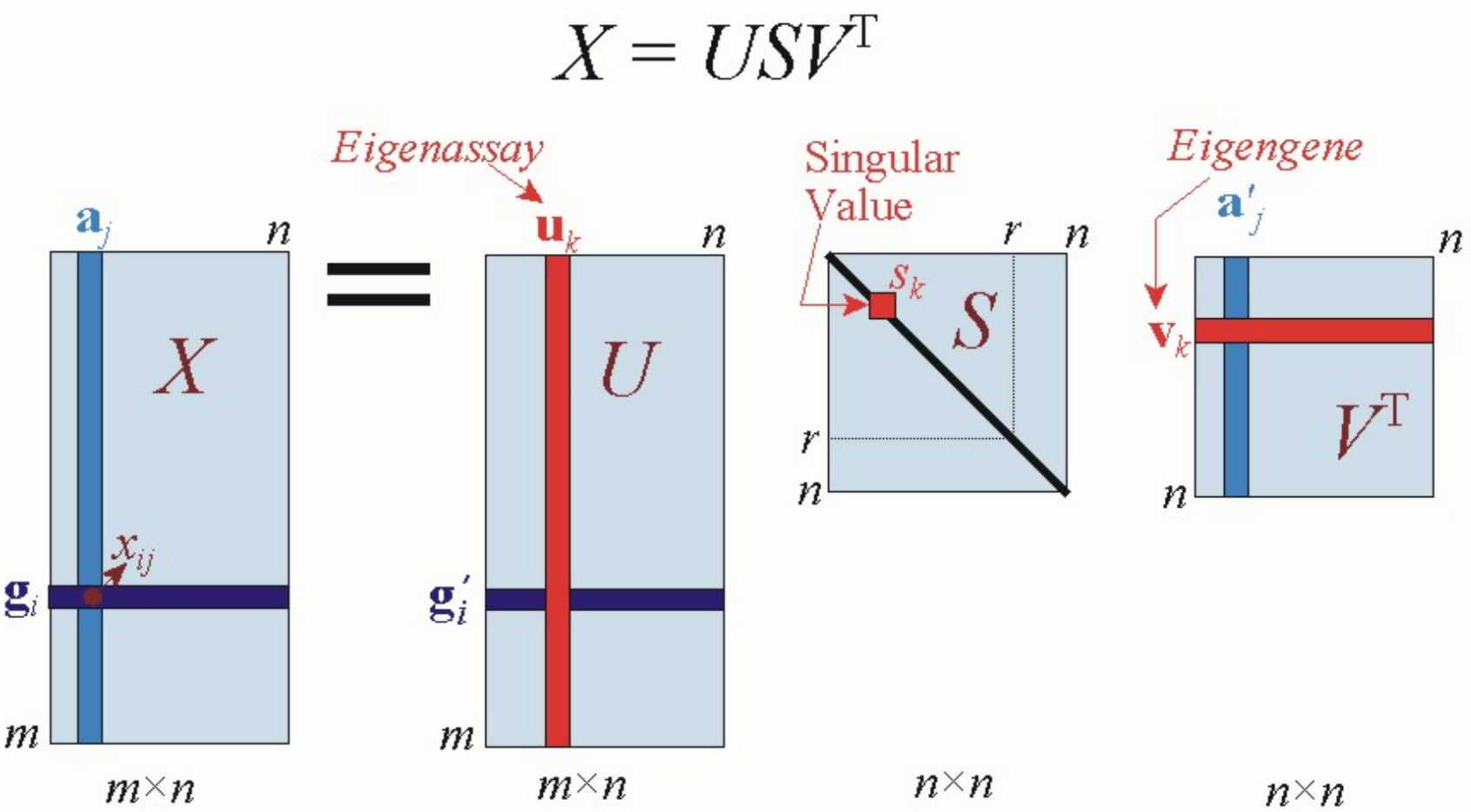
10 - Computing Population Axes

In Practical 1, I provided you with Population Axes

```
$ head hapmap.load1
rs28508199 A 2.204214e-04
rs1798246 A -2.111012e-05
rs7516150 C -1.002992e-05
rs875808 G 1.543642e-04
rs499416 T 3.165109e-05
rs4908605 T -2.698649e-05
```

These provide the (two) linear combinations of SNPs which best exhibit variation in the data

10 - Computing Population Axes



$\mathbf{X}$ is the data matrix, (by construction) $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^T = \mathbf{I}$, so $\mathbf{K} = \mathbf{X}\mathbf{X}^T = (\mathbf{U}\mathbf{S}\mathbf{V}^T)(\mathbf{V}\mathbf{S}\mathbf{U}^T) = \mathbf{U}\mathbf{S}^2\mathbf{U}^T$. So decomposing (PCA) $\mathbf{K}$ allows us to obtain $\mathbf{U}$, then $\mathbf{V} = \mathbf{X}^T\mathbf{U}\mathbf{S}^{-2}$

10 - Computing Population Axes

```
#compute K
../ldak.out --calc-kins-direct hapmap --bfile hapmap \
--ignore-weights YES

#decompose K to get U
../ldak.out --pca hapmap --grm hapmap --axes 2

> pca=as.matrix(read.table("hapmap.vect"))
> plot(pca[,3:4])

#multiply X^T U (divided by eigenvalues) to get V
../ldak.out --calc-pca-loads hapmap --bfile hapmap \
--grm hapmap --pcastem hapmap
```

You can now project new data onto `hapmap.load` (`--calc-scores`) in order to infer ancestry, detect population outliers

11 - Other Options

By default, LDAK scales kinships to have mean zero and average diagonal one Try out `--maf <number>`, `--calc-stats <output>`, `--make-snps` and `--make-phenos`

http://dougspeed.com/simulations/

http://dougspeed.com/data-filtering/

http://dougspeed.com/file-formats/