Polygenic risk score tutorial

Sarah Medland

Quantitative Genetics, QIMR Berghofer 16/07/2014

Steps involved

- Preparing the files required to run plink profile
- Converting imputed data to plink-dosage format
- Calculating the risk scores
- Running the analysis

3 scenarios

- Subtle differences in the methods need to be taken into consideration
  - You want to take a published GWAS/MA result set and see if they predict trait X in your data
  - Someone asks you to run a PRS analysis based on their GWAS/MA (usually as part of a replication effort)
  - You ask someone to run a PRS analysis in their data based on your results
    - The output from a PRS analysis run on the discovery sample is not meaningful!

Preparing the files required to run plink profile

- Ideally start with the full results set
  - SNP identifier
    - rs999
    - chr:BP 2:2450
    - chr:BP:Alleles 2:2450:AAA_T
    - Chr:BP: SNP/INDEL 2:2450:SNP or 2:2450:INDEL
  - Both Alleles
  - Pvalue
  - Effect
    - Beta from association with continuous trait
    - OR from an ordinal trait
      - Convert to log(OR)
    - Z score & MAF from an N weighted MA
      - Convert
        - ~~beta = Z/sqrt((1/N)*2pq)~~
          - p =MAF

Correction of typo (/ instead of *)15/5/15
SE (on a standardised scale) = sqrt((1/N)*2pq) where
N= number of chromosomes available for that snp, for an autosome this is 2*sample size for that snp, p =MAF and q=1-MAF
Beta (on a standardised scale) = Z* sqrt((1/N)*2pq)
This would be a standardised beta but if you had the trait variance you could multiply it up

Preparing the files required to run plink profile

- Select the SNPs you can use for PRS
  - Consider strand
    - Option 1: drop ambiguous SNPs & indels
    - Option 2: keep ambiguous SNPs & indels
      - then check alignment via MAF
        - Can't do this if you are asking someone to run a PRS for you

What is a strand ambiguous SNP

- Remember bases are paired A-T C-G
- If the fwd strand has a A/C snp the reverse strand has a T/G snp
- A/T and C/G snps are ambiguous because you can't differentiate the strands with out AF info
  - Easy enough is the snp has MAF <.3 but easy to mess up
- There is no 1 source of strand info
  - Data bases don't agree
  - Strand can change between builds
  - Illumnia dodges this TOP/BOT strand
  - Affy provides strand – based on what?

Preparing the files required to run plink profile

- Select the SNPs you can use for PRS
  - Consider strand
    - Option 1: drop ambiguous SNPs & indels

```
awk '{ print $1, $2 $3, $4, $5}' GWAS.result > temp
awk '{ if ($2 == "AC" || $2 == "AG" || $2 == "CA" || $2 == "CT" || $2 == "GA" || $2
    == "GT" || $2 == "TC" || $2 == "TG" ) print $0}' temp > GWAS.noambig
```

  - Consider reference
    - If score calculated from called genotypes usually use a subset of founders from the target or original data set
    - If score calculated from imputed data usually use imputation ref panel data
      - Check naming format matches the GWAS.noambig and target data formats
      - Check strand of the GWAS.noambig file against the reference file

Preparing the files required to run plink profile

- Select the SNPs you can use for PRS
  - Consider the SNPs you have available
    - Make lists of snps available in the target dataset – will be used with an extract command in plink to subset the reference data
    - Called genotypes
      - Make a list of snps available using the bim file from the target dataset

```
awk '{print $1}' file.bim > available.snps
```

    - Imputed genotypes
      - Makes a list of snps available using the info file selecting only those snps with high enough MAF and Rsq

```
for ((i=1;i<=22;i++))
do
awk '{ if ($5<=.01 & $5<=.99 & $6>=.8) print $1}' file"$i".info >> available.snps
done
```

Preparing the files required to run plink profile

- Consider LD
  - PRS can be calculated without pruning but convention is to prune prior to calculation

```
#Clump data in 2 rounds using plink2
#1st clumping & extract tops snps for 2nd round
    for ((i=1;i<=22;i++))
    do
    plink2 --bfile reference --chr "$i" –extract available.snps --clump GWAS.noambig
        --clump-p1 1 --clump-p2 1 --clump-r2 .5 --clump-kb 250  --out traitX"$i".round1
   awk '{print $3, $5}' traitX"$i".round1.clumped > traitX"$i".round2.input
   awk '{print $3}' traitX"$i".round1.clumped > traitX"$i".extract2
   done
#2nd clumping & extract tops snps for profile
    for ((i=1;i<=22;i++))
    do
    plink2 --bfile reference --chr "$i" --extract traitX"$i".extract2 --clump traitX"$i".round2.input
        --clump-p1 1 --clump-p2 1 --clump-r2 .2 --clump-kb 5000 --out traitX"$i".round2
   awk '{print $3}' traitX"$i".round2.clumped > traitX"$i".selected
    done
```

Preparing the files required to run plink profile

- Make the files to run --profile in plink
  - The traitX"$i".selected files will contain the lists of top snps
  - Merge the alleles, effect & P values onto these files
- Check the strand of the selected snps against the target alleles
  - Merge the alleles of the target set onto these files
  - Check using awk

```
for ((i=1;i<=22;i++))
do
awk '{ if ($7==$9 || $7==$10  ) print $0, "match" ; if ($7!=$9 && $7!=$10 ) print $0, "mismatch"}' traitX."$j".merged > strandcheck.traitX."$i"
grep mismatch strandcheck.traitX*
done
```

  - If any SNPs are flagged as mismatched you will have to manual update the merged file - ie flip the strands but leave the effect as is
  - Use awk to make the score and P value files
    - Score files contain SNPid EffectAllele Effect
    - P files contain SNPid Pvalue

Steps involved

- Preparing the files required to run plink profile
  - Extracted and clumped GWAS results
  - Checked strand
  - Made score and P files
- Converting imputed data to plink-dosage format
- Calculating the risk scores
- Running the analysis

Steps involved

- Preparing the files required to run plink profile
- Converting imputed data to plink-dosage format
- Calculating the risk scores
- Running the analysis

Converting imputed data to plink-dosage format

- DON'T DO THIS IF YOU ARE USING GENIEPI DATA
  - Scott has done this for us and its sitting in the GWAS area

- What is the difference?

minimac format (effect allele is A1)

-no header-

F1->I1 DOSE 0.00 0.00 1.99

F2->I2 DOSE 1.01 0.00 0.99

F3->I3 DOSE 0.00 0.94 0.00

plink format (effect allele is A1)

SNP A1 A2 F1 I1 F2 I2 F3 I3

rs0001 A C 0.00 1.01 0.00

rs0002 G A 0.00 0.00 0.94

rs0003 A C 1.99 0.99 0.00

Converting imputed data to plink-dosage format

- If the data is stored in minimac dosage format
  - Use dose2plink
    http://www.genepi.qimr.edu.au/staff/sarahMe/dose2plink.html
- If the data is in any other format
  - Use fcGENE

    http://sourceforge.net/projects/fcgene/
    - Can't work with minimac/mach dosage format but can work with probs format
- Once converted merge the files together so 1 file per chr
  - if the same individuals are in all files and the order of individuals is the same you can cat the files together - remembering to remove the headers on the subsequent files
  - Plink can merge files on the fly:
    plink --dosage myfile.lst list --fam mydata.fam
    where myfile.lst is a list of file names e.g.
    chr1.dose
    chr2.dose

Steps involved

- Preparing the files required to run plink profile
- Converting imputed data to plink-dosage format
- Calculating the risk scores
- Running the analysis

Calculating the risk scores

- This is the easy part
  - For gzipped files:
    plink --noweb --dosage chr21.pdat.gz format=1 Z --fam chr.21.pfam --score traitX."$j".score --out example
  - For non-gzipped files:
    plink --noweb --dosage chr21.pdat format=1 --fam chr.21.pfam --score traitX."$j".score --out example
  - For called genotype files:
    plink --noweb --bfile example --score traitX."$j".score --out example
- Once you have the scores sum across chromosomes

Calculating the risk scores

- Other options
  - If you want to calculate a series of scores using different P value cutoffs

```
plink --noweb --dosage chr21.pdat.gz format=1 Z --fam chr.21.pfam --
    score traitX."$j".score --q-score-file traitX."$j".P
    --q-score-range p.ranges --out example
```

```
p.ranges
S1  0.00 0.000001
S2  0.00 0.01
S3  0.00 0.10
S4  0.00 0.50
S5  0.00 1.00
```

Calculating the risk scores

- Important points
  - Strand
    - If you are using called genotypes

      Reading set of predictors from [flip.score]

      Read 3 predictors; 3 mapped to SNPs;

      0 to allelesWriting problem SNPs in predictor to [flippedHC.nopred]

      Writing profiles to [flippedHC.profile]
    - You don't get this with imputed data
  - Allele flipping - reverse coding
    - Works in both called and imputed data

Calculating the risk scores

- Important points
  - Scaling
    - If you are using called genotypes

      Plink will divide the total score by the number of snps so you need to /22 after merging across the chromosomes
    - You don't get this with imputed data

      Plink doesn't divide by Nsnps
  - Note: Plink treats the effect as if its the difference between homozygotes not the additive increment of a single risk allele
    - Same for both called and imputed genotypes

Steps involved

- Preparing the files required to run plink profile
- Converting imputed data to plink-dosage format
- Calculating the risk scores
- Running the analysis

Running the analysis

- Unrelateds/ 1 person per family
  - run the regression analyses for each trait in R (or a similar software)

  - ```
    base    <- lm (ICV ~        age + sex + PC1 + PC2 +PC3 +PC4 + other-covariates, data =mydata)
    score1 <- lm (ICV ~ S1 + age + sex + PC1 + PC2 +PC3 +PC4 + other-covariates, data =mydata)
    score2 <- lm (ICV ~ S2 + age + sex + PC1 + PC2 +PC3 +PC4 + other-covariates, data =mydata)
    score3 <- lm (ICV ~ S3 + age + sex + PC1 + PC2 +PC3 +PC4 + other-covariates, data =mydata)
    score4 <- lm (ICV ~ S4 + age + sex + PC1 + PC2 +PC3 +PC4 + other-covariates, data =mydata)
    score5 <- lm (ICV ~ S5 + age + sex + PC1 + PC2 +PC3 +PC4 + other-covariates, data =mydata)
    model_base <- summary(base)
    model_score1 <- summary(score1)
    model_score2 <- summary(score2)
    model_score3 <- summary(score3)
    model_score4 <- summary(score4)
    model_score5 <- summary(score5)
    model_base$r.squared
    model_score1$r.squared
    model_score2$r.squared
    model_score3$r.squared
    model_score4$r.squared
    model_score5$r.squared
    anova(base,score1)
    anova(base,score2)
    anova(base,score3)
    anova(base,score4)
    anova(base,score5)
    ```


    ```

Running the analysis

- Relateds
  - Run the analysis in Mx
  - Run the analysis in genable
    - Scripts will be available from my webpage
  - Run the analysis in R using sandwhich estimator