## high-throughput GWAS on Arabidopsis

### Trait file:

Necessary format trait file:

|  |  |  |  |
| --- | --- | --- | --- |
| ecotype\_id | Trait001 | Trait002 | Trait003 |
| 1 | 0.001129448 | 5.381654767 | 623.6019452 |
| 23 | … |  |  |
| 66 |  |  |  |
| 81 |  |  |  |

1st column must contain the numeric ecotype ids according to Nordborg’s lab

Headers of traits can be anything but according to R conventions, they will be outputted as trait1, trait2,…

One value for each ecotype

Missing values : NA

Sorting on the first column is not necessary

Copy the trait file to your working dir

### Necessary scripts for GWAS analysis

* copy “emma.r” to your working dir;
* copy “amm\_gwas\_vesto\_v3.R to your working dir;
* copy “htu\_gwas\_vesto\_v3.R " to your working dir ;
* copy “htu\_gwas\_vesto\_v3\_noplots.R " to your working dir ;
* copy “htu\_gwas\_log10\_vesto\_v3.R " to your working dir ;
* copy “htu\_gwas\_log10\_vesto\_v3\_noplots.R " to your working dir ;
* copy “htu\_gwas\_v3.sh" to your working dir;
* copy “htu\_gwas\_v3\_noplots.sh" to your working dir;
* copy “htu\_gwas\_log10\_v3.sh" to your working dir;
* copy “htu\_gwas\_log10\_v3\_noplots.sh" to your working dir;
* copy “snps2016.Rdata” to your working dir;
* copy “kinship.Rdata” to your working dir;

### GWAS analysis

Default settings are: MAF<5% (p=0.05). This can be changed when calling the amm\_gwas\_vesto\_v3 function in one of the *htu….R* files

**To Run on a server:**

# Go to your working directory (the following is just an example)

[vesto@midas ~]$ **cd ..**

[vesto@midas /home]$ **cd ..**

[vesto@midas /]$ **cd group/biostat**

[vesto@midas biostat]$ **cd myGWASprojects/ARABIDOPSIS/SCRIPTS**

# Load necessary modules

[vesto@midas SCRIPTS]$ **module load gridengine**

[vesto@midas SCRIPTS]$ **module load R**

# default to date is *R/x86\_64/3.2.2*

# convert necessary txt inputfiles into unix format:

[vesto@midas SCRIPTS]$ **dos2unix traitfile.txt**

# run the GWAS:

# the GWAS can be run

* on untransformed data with **htu\_gwas\_v3.sh** producing Manhattan plots
* on untransformed data with **htu\_gwas\_v3\_noplots.sh** suppressing the Manhattan plots (faster)
* on log10 transformed data with **htu\_gwas\_log10\_v3.sh**
* on log10 transformed data with **htu\_gwas\_log10\_v3\_noplots.sh** suppressing the Manhattan plots (faster).

# the shell script

* **htu\_gwas\_v3.sh** calls the R program **htu\_gwas\_vesto\_v3.R**
* **htu\_gwas\_v3\_noplots.sh** calls the R program **htu\_gwas\_vesto\_v3\_noplots.R**
* **htu\_gwas\_log10\_v3.sh** calls the R program **htu\_gwas\_log10\_vesto\_v3.R**
* **htu\_gwas\_log10\_v3\_noplots.sh** calls the R program **htu\_gwas\_log10\_vesto\_v3\_noplots.R**

# Edit the working directory and the traitfile name in the R script of your preference

# edit in the shell script the number of traits in line 7: #$ -t 2-4 . Always start with 2 (column 1 is ecotype\_id).

# submit the analysis

[vesto@midas SCRIPTS]$ **qsub –l h\_vmem=4G htu\_gwas\_log10\_v3.sh**

# this will generate a manhattan plot for each trait and an outputfile

***Output\_traitT.txt*** and ***manhattan\_traitT.jpg***

Output\_traitT.txt contains:

1st column with no header: rownumber

SNP: snp ID

Chr: chromosome number

Pos: position on chromosome in bp

AC\_1: frequence allele 1

AC\_0: frequency allele 0

MAC: minor allele frequence

MAF: proportion of minor allele

Pval: pvalue

# !! The sum of AC\_1 and AC\_0 are the number of ecotypes analysed. Verify whether this is as you expected.

### High-throughput annotation

This requires following additional scripts:

* copy “append.sh” to your working dir (author Veronique);
* copy “append.R” to your working dir (author Veronique);
* copy “appendID.sh” to your working dir (author Veronique);
* copy “findpeaks\_v5.c” to your working dir (author Yvan);
* copy “analyze\_regions4.pl” to your working dir (author Yvan);
* copy “gene\_ontology\_ext05042012.obo” to your working dir (author Yvan);
* copy “ATH\_GO\_GOSLIM04032012.txt” to your working dir (author Yvan);
* copy “TAIR10\_GFF3\_genes.gff” to your working dir (author Yvan);

**To run on a server:**

# concatenate the generated output files

[vesto@midas SCRIPTS]$ **dos2unix append.sh**

[vesto@midas SCRIPTS]$ **dos2unix appendID.sh**

[vesto@midas SCRIPTS]$ **sh append.sh**

# this generates the file ***out.txt***

[vesto@midas SCRIPTS]$ **qsub –l h\_vmem=160G appendID.sh**

# this generates the file ***input\_findpeaks.txt***

[vesto@midas SCRIPTS]$ **gcc –O3 findpeaks\_v5.c –o findpeaks\_v5 –lm**

# gcc to compile C code

# -O3 (letter O for optimalisation, 3 is the optimalisation norm, can go up to 6, has something to do with floating points and roundings

# -o output

# -lm to indicate that the code contains mathematical functions

[vesto@midas SCRIPTS]$ **./findpeaks\_v5 input\_findpeaks.txt**

# the command findpeaks\_v5 searches for all SNPs with –logp>6, it then looks at 20 upstream and downstream SNPs, and considers a region as associated with the phenotype when there are at least 5 more SNPs with a –logp>3

# these 3 arguments can be changed with:

# winsize: how many SNPs should be condidered

# cothreshold: the required –logp values of the other SNP associations, defined by quorum

# quorum: number of SNPs required that fulfill the cothreshold

[vesto@midas SCRIPTS]$ **./findpeaks\_v5 input\_findpeaks.txt 20 3.0 5**

# this generates the file **input\_findpeaks.txt.regions**

# Finally search for a GO-term in associated regions

[vesto@midas SCRIPTS]$ **perl analyze\_regions4.pl input\_findpeaks.txt.regions GO:0008168 > GO\_0008168.txt**