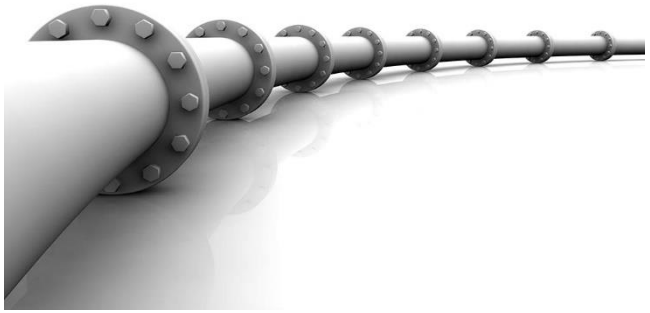
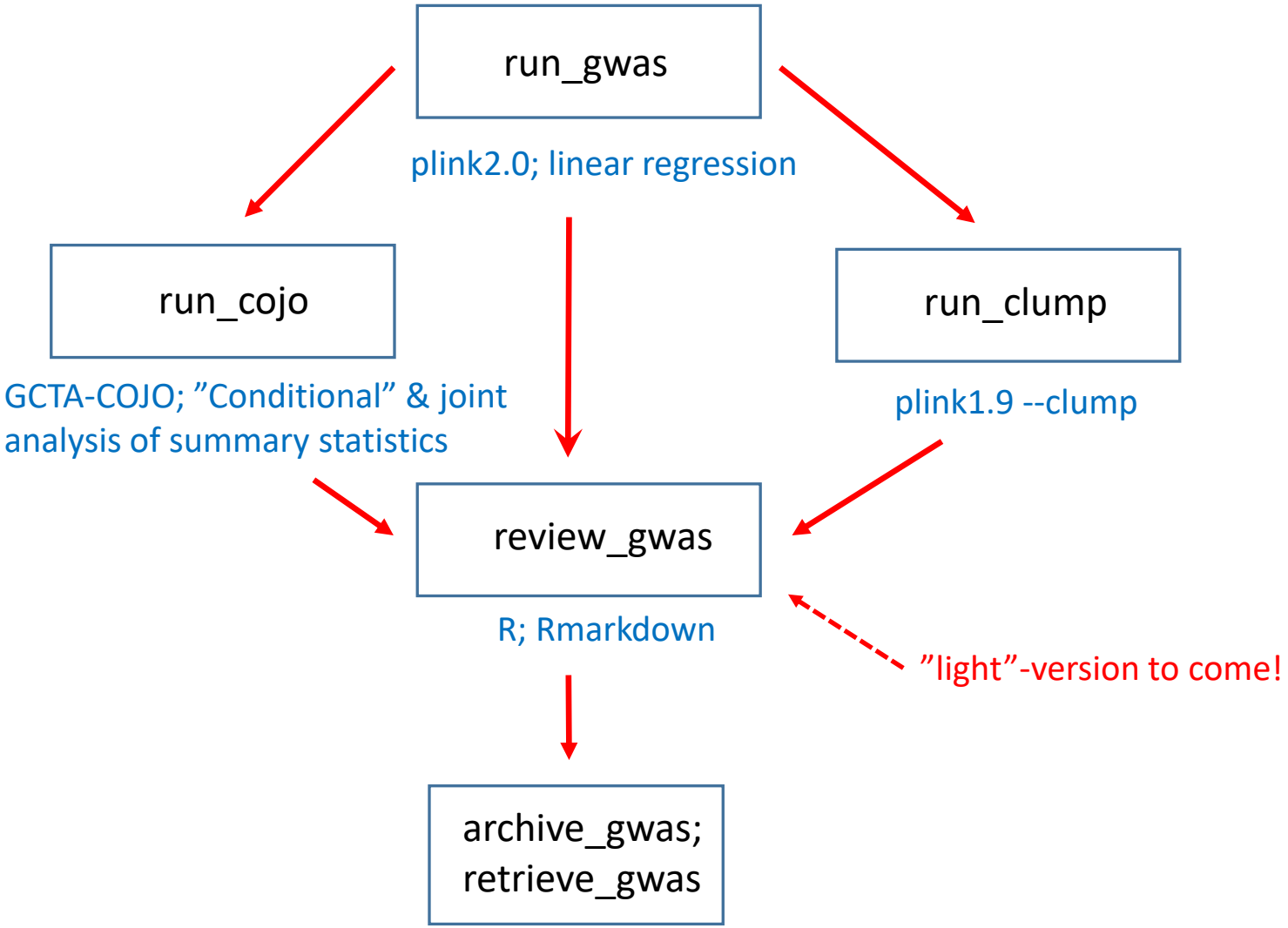
GWAS-Pipeline

April 2020



Workflow


run_gwas
plink2.0; linear regression
run_cojo
GCTA-COJO; ”Conditional” & joint analysis of summary statistics
run_clump
plink1.9 --clump
review_gwas
R; Rmarkdown
”light”-version to come!
archive_gwas;
retrieve_gwas

Preparation

- add to ~/.bashrc :
  - export SCRIPT_FOLDER="/proj/sens2019016/GWAS_SCRIPTS"
  - PATH=$SCRIPT_FOLDER:$PATH
  - (the path can be chosen arbitrarily, but the variable name must be SCRIPT_FOLDER !)
- if just edited:
  - source ~/.bashrc (only in windows just open when edited)

Place a number of files in your home directory (cd ~):

- gwas_settings.sh
- cojo_settings.sh
- clumpo_settings.sh
- review_settings.sh
- review_settings.R
- archive_settings.sh

Regression

https://www.cog-genomics.org/plink/2.0/

PLINK 2.00 alpha

PLINK 2.0 alpha was developed by **Christopher Chang**, with support from **GRAIL, Inc.** and **Human Longevity, Inc.**, and substantial input from Stanford's Department of Biomedical Data Science. (More detailed credits.) (Usage questions should be sent to the **plink2-users Google group**, not Christopher's email.)

Binary downloads

	Build	
Operating system	**Development (28 Mar)**	Alpha 2.3 final (24 Jan)
Linux AVX2 Intel[1]	**download**	download
Linux 64-bit Intel[1]	download	download
Linux 32-bit	download	download
macOS AVX2	**download**	download
macOS 64-bit	download	download
Windows AVX2	**download**	download
Windows 64-bit	download	download
Windows 32-bit	download	download

1: These builds can still run on AMD processors, but they're statically linked to Intel MKL, so some linear algebra operations will be slow. We will try to provide an AMD Zen-optimized build as soon as supporting libraries are available.

Source code and build instructions are available on GitHub. (Here's another copy of the source code.)

run_gwas

```
[umenzel@sens2019016-bianca GWAS_TEST]$ run_gwas

  Usage: run_gwas
         -i|--id <string>                no default
         -p|--phenofile <file>           no default
         -pn|--phenoname <string>        no default
         -g|--genoid <string>            /home/umenzel/gwas_settings.sh
         -c|--chr <int>[-<int>]          /home/umenzel/gwas_settings.sh
         -h|--hwe <real>                 /home/umenzel/gwas_settings.sh
         -pf|--phenofolder <folder>      /home/umenzel/gwas_settings.sh
         -cf|--covarfile <file>          /home/umenzel/gwas_settings.sh
         -cn|--covarname <string>        /home/umenzel/gwas_settings.sh
         --mac <int>                     /home/umenzel/gwas_settings.sh
         --mr2 <range>                   /home/umenzel/gwas_settings.sh
         -m|--minutes <int>              /home/umenzel/gwas_settings.sh
         --ask <y|n>                     /home/umenzel/gwas_settings.sh
```

plink2.0: https://www.cog-genomics.org/plink/2.0/

run_gwas

Call

With mandatory command line parameters only:

```
run_gwas --id LIV_MULT5 --phenofile liver_fat_faked.txt \
        --phenoname liv1,liv2,liv3,liv4,liv5,liv6,liv7,liv8,liv9,liv10
```

- all other parameters are read from ~/gwas_settings.sh
- runs phenotypes and chromosomes in parallel, uses 22 nodes or cores)
- files needed: liver_fat_faked.txt (where tp place?: see ~/gwas_settings.sh)
- can be started wherever you find enough disk space (script checks available space)

```
[umenzel@sens2019016-bianca GWAS_TEST]$ head liver_fat_faked.txt
#FID    IID     liv1    liv2    liv3    liv4    liv5    liv6    liv7    liv8    liv9    liv10
1000401 1000401 4.6117716835    4.6117716835    4.6117716835    4.6117716835    4.6117716835    4.611
1000435 1000435 6.4229896333    6.4229896333    6.4229896333    6.4229896333    6.4229896333    6.422
1000456 1000456 3.1423040689    3.1423040689    3.1423040689    3.1423040689    3.1423040689    3.142
1000493 1000493 2.5408244938    2.5408244938    2.5408244938    2.5408244938    2.5408244938    2.540
1000843 1000843 8.4709656153    8.4709656153    8.4709656153    8.4709656153    8.4709656153    8.470
1000885 1000885 21.8226416204   21.8226416204   21.8226416204   21.8226416204   21.8226416204   21.82
1001146 1001146 1.1720321422    1.1720321422    1.1720321422    1.1720321422    1.1720321422    1.172
1001215 1001215 11.7379679852   11.7379679852   11.7379679852   11.7379679852   11.7379679852   11.73
1001310 1001310 6.5725167581    6.5725167581    6.5725167581    6.5725167581    6.5725167581    6.572
```

tab-separated

run_gwas

Parameter settings hierarchy

- ~/gwas_settings.sh (sourced by run_gwas.sh and gwas_chr.sh)
- parameters not changing frequently

```
# GWAS settings (command line paramters overwrite these settings):

# this file is sourced by run_gwas.sh and gwas_chr.sh

chrom="1-22"                                    # all autosomes
genofile_id="MF"                                # filtered for ethnicity, kinship
#genofolder="/proj/sens2019016/GENOTYPES/PGEN"  # location of input genotype file
genofolder="/proj/sens2019016/GENOTYPES/PGEN1"  # location of input genotype files
# phenofolder="/proj/sens2019016/PHENOTYPES"    # location of input phenotype and covariate f
phenofolder="."
covarfile="GWAS_covariates.txt"                 # covariates file (located in phenofolder)
covarname="PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10,array,sex,age"  # covariates, all but age

mac=30                                          # minor allele count
machr2="0.8 2.0"                                # mach-r2 range   (imputation quality)
hwe_pval=1.0e-6                                 # Hardy-Weinberg p-value filter

ask="n"                                         # ask the user for confirmation of the input
plink2_version="plink2/2.00-alpha-2-20190429"
minutes=20                                      # required runtime for each chromosome in min
partition="node"                                # partition , "core"  might run out of memory
minspace=1000000000                             # 1 TByte    minimum required disk space for



## +++ Genofile_id identifiers:
#
#   ukb_imp_v3   487.409 samples   complete genotype dataset
#        FTD     337.482 samples   filtered (ethnic background, kinship)
#        MRI      37.869 samples   MRI samples, not filtered
#        MF       27.212 samples   MRI & filtered
```

OOPS: stick to bash syntax!

- settings are overwritten by command line parameters
  - parameters changing frequently (phenofile, phenoname)
  - some command line parameters are mandatory

run_gwas

Genotype datasets

- /proj/sens2019016/GENOTYPES/...
  - PGEN_ORIG (4 datasets) + PGEN (MF only , unique marker names)
  - BED_ORIG (MF only) + BED (MF only, unique marker names)
- run_gwas (*plink2.0*) uses .pgen (.pgen, .pvar, .psam)
- run_cojo and run_clump (*plink1.9*) use .bed (.bed, .bim, .fam)

geno-id	#samples	description
ukb_imp_v3	487.409	complete genotype dataset
FTD	337.482	filtered (ethnicity, kinship)
MRI	37.869	MRI samples, not filtered
MF	27.212	MRI & filtered

MRI = Magnetic Resonance Imaging

run_gwas

Genotype filename convention

Example: - -genoid MF requires:

- MF_chr1.pvar ; MF_chr1.psam ; MF_chr1.pgen
- MF_chr2.pvar ; MF_chr2.psam ; MF_chr2.pgen
- MF_chr3.pvar ; MF_chr3.psam ; MF_chr3.pgen
- ...
- MF_chr21.pvar ; MF_chr21.psam ; MF_chr21.pgen
- MF_chr22.pvar ; MF_chr22.psam ; MF_chr22.pgen

... in the genotype folder

run_gwas:

```
pgen_prefix=${genofolder}"/"${genofile_id}"_chr"$chrom
psam=${pgen_prefix}".psam"  # MF_chr22.psam
pvar=${pgen_prefix}".pvar"  # MF_chr22.pvar
pgen=${pgen_prefix}".pgen"  # MF_chr22.pgen
```

run_gwas

Input verification

```
## +++ Check if the variables are defined
```

```
Checking the phenotype names with phenotype file ${phenopath}
```

```
## +++ Chromosomes
```

```
## +++ Let the user confirm the choice of parameters if $ask = "y"
```

```
## +++ Check available disk space:
```

```
## +++ Check availability of input files and genotype files:
```

... different checks for different programs

ERROR (scriptname): ...

run_gwas

Output

```
umenzel sens2019016    12248 Mar 23 10:48 LIV_MULT4_gwas.log
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv1.glm.linear
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv2.glm.linear
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv4.glm.linear
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv3.glm.linear
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv5.glm.linear
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv6.glm.linear
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv8.glm.linear
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv7.glm.linear
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv9.glm.linear
umenzel sens2019016 88234950 Mar 23 10:55 LIV_MULT4_gwas_chr1.liv10.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv1.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv2.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv4.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv3.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv5.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv6.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv7.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv8.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv10.glm.linear
umenzel sens2019016 96485309 Mar 23 10:55 LIV_MULT4_gwas_chr2.liv9.glm.linear
umenzel sens2019016     6938 Mar 23 10:55 LIV_MULT4_gwas_chrom1.log
umenzel sens2019016     6938 Mar 23 10:55 LIV_MULT4_gwas_chrom2.log
umenzel sens2019016 81464338 Mar 23 10:56 LIV_MULT4_gwas_chr3.liv1.glm.linear
umenzel sens2019016 81464338 Mar 23 10:56 LIV_MULT4_gwas_chr3.liv3.glm.linear
umenzel sens2019016 81464338 Mar 23 10:56 LIV_MULT4_gwas_chr3.liv2.glm.linear
umenzel sens2019016 81464338 Mar 23 10:56 LIV_MULT4_gwas_chr3.liv4.glm.linear
umenzel sens2019016 81464338 Mar 23 10:56 LIV_MULT4_gwas_chr3.liv5.glm.linear
umenzel sens2019016 81464338 Mar 23 10:56 LIV_MULT4_gwas_chr3.liv6.glm.linear
```

run_gwas
Output

- regression results:

```
[umenzel@sens2019016-bianca LIV_MULT5]$ head LIV_MULT5_gwas_chr8.liv1.glm.linear
#CHROM  POS     ID      REF     ALT1    A1      A1_FREQ OBS_CT  BETA    SE      P
8       34440   rs556230355_TTTTTG_T    TTTTTG  T       TTTTTG  0.0800938       18771   0.030
8       46296   rs62485412_C_T  C       T       C       0.0728311       18771   0.024052
8       46307   rs561412547_T_G T       G       T       0.0103746       18771   0.118219
8       79779   rs577617180_A_C A       C       A       0.00109837      18771   -0.889391
8       125528  rs138554754_TC_T        TC      T       T       0.0888837       18771   -0.04
8       141575  rs750339685_T_TG        T       TG      TG      0.0919899       18771   0.059
8       143207  rs183349913_G_T G       T       G       0.0434309       18771   -0.0576837
8       146791  rs4141094_G_C   G       C       C       0.0814949       18771   0.0632585
8       151755  rs62485446_C_T  C       T       T       0.0840013       18771   0.0121125
```

- logfiles: ${ident}_gwas_chr${chrom}.log, e.g. LIV4_gwas_chr22.log
- grep –i error *.log
- grep –i warn *.log

```
Warning: --hwe observation counts vary by more than 10%.  Consider using
Warning: --hwe observation counts vary by more than 10%.  Consider using
Warning: --hwe observation counts vary by more than 10%.  Consider using
```

run_gwas
Output

- parameter file: ${ident}_gwas_params.txt

```
[umenzel@sens2019016-bianca LIV_MULT5]$ cat LIV_MULT5_gwas_params.txt
plink2_version plink2/2.00-alpha-2-20190429
workfolder /proj/sens2019016/GWAS_TEST/LIV_MULT5
ident LIV_MULT5
cstart 1
cstop 22
genotype_id MF
phenofile liver_fat_faked.txt
phenoname liv1,liv2,liv3,liv4,liv5,liv6,liv7,liv8,liv9,liv10
covarfile GWAS_covariates.txt
covarname PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10,array,sex,age
mac 30
hwe_pval 1.0e-6
machr2_low 0.8
machr2_high 2.0
```

- used by subsequent scripts
- complemented by subsequent scripts

Pruning I

https://cnsgenomics.com/software/gcta/#COJO

COJO

GCTA-COJO: multi-SNP-based conditional & joint association analysis using GWAS summary data

--cojo-file test.ma

Input the summary-level statistics from a meta-analysis GWAS (or a single GWAS).

> Input file format
> test.ma

```
SNP A1 A2 freq b se p N
rs1001 A G 0.8493 0.0024 0.0055 0.6653 129850
rs1002 C G 0.0306 0.0034 0.0115 0.7659 129799
rs1003 A C 0.5128 0.0045 0.0038 0.2319 129830
...
```

Columns are SNP, the effect allele, the other allele, frequency of the effect allele, effect size, standard error, p-value program. Important: "A1" needs to be the effect allele with "A2" being the other allele and "freq" should be the frequ

Note: 1) For a case-control study, the effect size should be log(odds ratio) with its corresponding standard error. 2) I focuses on a subset of SNPs because the program needs the summary data of all SNPs to calculate the phenotypic the COJO analysis in a certain genomic region.

--cojo-slct

Perform a stepwise model selection procedure to select independently associated SNPs. Results will be saved in a SNPs.

Why not using a true conditional analysis ?

Conditional and joint multiple-SNP analysis of GWAS summary statistics identifies additional variants influencing complex traits

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3593158/

Conditional analysis has been used as a tool to identify secondary association signals at a locus[3,9,10], involving association analysis conditioning on the primary associated SNP at the locus to test whether there are any other SNPs significantly associated. A more general and comprehensive strategy would be to perform a conditional analysis, starting with the top associated SNP, across the whole genome followed by a stepwise procedure of selecting additional SNPs, one by one, according to their conditional *P* values. Such a strategy would allow the discovery of more than two associated SNPs at a locus[7,11]. For meta-analysis of a large number of participating studies, however, pooled individual-level genotype data are usually unavailable, such that conditional analysis can only be performed at the level of individual studies. Summary results from individual studies are then collected and combined through a second round of meta-analysis. This procedure is administratively onerous. It often takes months to organize and perform a single round of this kind of conditional meta-analysis, and it would be extremely time-consuming and therefore impractical to implement a stepwise selection procedure in this manner.

run_cojo

Parameters

```
[umenzel@sens2019016-bianca GWAS_TEST]$ run_cojo

  Usage: run_cojo
         -i|--id <string>                no default
         -pn|--phenoname <string>        defaults to all gwas results
         -p|--pval <real>                /home/umenzel/cojo_settings.sh
         -w|--window <integer>           /home/umenzel/cojo_settings.sh
         -m|--minutes <int>              /home/umenzel/cojo_settings.sh
         --ask <y|n>                     /home/umenzel/cojo_settings.sh
```

--phenoname:

- a single phenotype name
- a comma-separated list with multiple phenoytype names
- if not invoked: all phenotype names that have been run through GWAS

--pval

- threshold to define a genom-wide significant hit
- default values in ~/cojo_settings.sh

run_cojo

Call

With mandatory command line parameters only:

```
run_cojo --id LIV_MULT5 --phenoname liv2,liv10   # two phenotypes
```

- start from within GWAS folder (the folder created by run_gwas)
- all other parameters are read from ~/cojo_settings.sh
- runs phenotypes and chromosomes in parallel, uses 25 * #phenotypes nodes or cores, see below)
- reads also parameters from parameter file (see above)
  - 3rd method to acquire parameters (besides ~/clumps_settings.sh and command line)

run_cojo

Input verification

```
## +++ Check if the variables are defined
```

```
## +++ Check folder:
```

must be the folder where the gwas results reside!

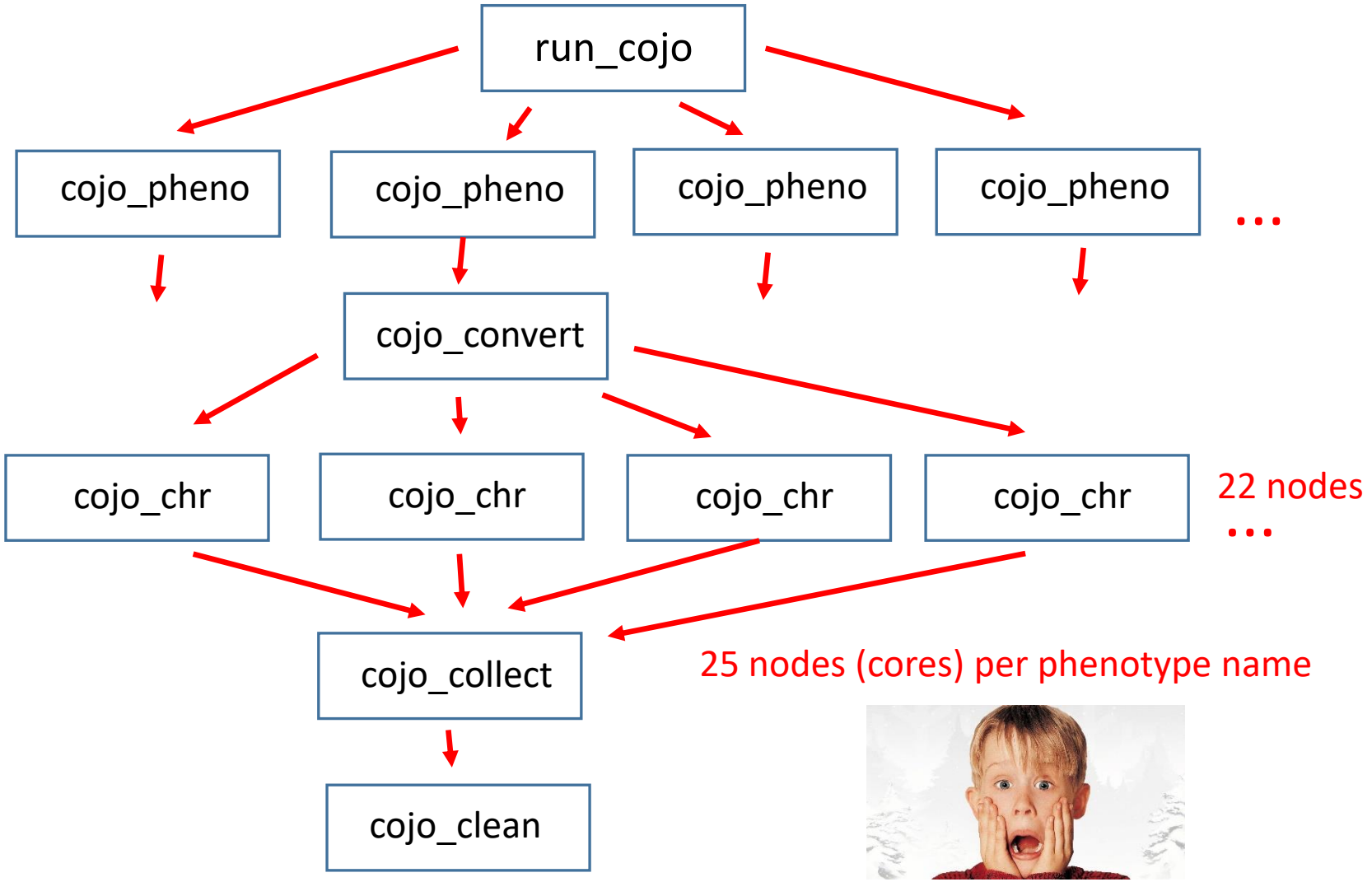
```
## +++ Check if $phenoname is valid (user input)
```

(compares with parameter file)

```
## +++ Check chromosomes:
```

```
## +++ Check available disk space:
```

ERROR (scriptname): ...



- Trick: sleep_between_pheno=100 (to be set in ~/cojo_settings.sh)
  - pauses 100 minutes between phenotype names (or whatever number you choose)
- Emergency break: scancel –u <username>

cojo_pheno

- you need not to care about this (sub-)program !
- runs a single phenotype
- some parameters are read from the ”parameter file” (already created by run_gwas)

```
paramfile="${ident}_gwas_params.txt"
```

```
genoid=$( awk '{if($1 == "genotype_id") print $2}' ${paramfile} )        # MF
cstart=$( awk '{if($1 == "cstart") print $2}' ${paramfile} )             # 1
cstop=$( awk '{if($1 == "cstop") print $2}' ${paramfile} )               # 22
```

- the parameter file has always ”the last word” (in all scripts)
  - as an example, it does not make sense to use different genotype data in run_gwas and in run_cojo

cojo_convert

- you need not to care about this (sub-)program !
- cojo_convert concatenates input data, must finish first
- subsequent jobs must wait (”Dependency”)

```
4473      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4474      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4475      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4476      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4477      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4478      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4479      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4480      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4481      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4482      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4483      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4484      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4485      node LIV_MULT  umenzel PD       0:00      1 (Dependency)
4486      node  COLLECT  umenzel PD       0:00      1 (Dependency)
4487      node    CLEAN  umenzel PD       0:00      1 (Dependency)
4463      node  CONVERT  umenzel  R       0:10      1 sens2019016-b17
4438      node  CONVERT  umenzel  R       0:13      1 sens2019016-b15
```

cojo_convert

- subsequent jobs start when cojo_convert is finished
- cojo_collect and cojo_clean are still waiting ... (”Dependency”)

```
4476      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4477      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4478      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4479      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4480      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4481      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4482      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4483      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4484      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4485      node LIV_MULT  umenzel PD       0:00      1 (Priority)
4486      node  COLLECT  umenzel PD       0:00      1 (Dependency)
4487      node    CLEAN  umenzel PD       0:00      1 (Dependency)
4447      node LIV_MULT  umenzel  R       0:13      1 sens2019016-b17
4440      node LIV_MULT  umenzel  R       0:16      1 sens2019016-b19
4441      node LIV_MULT  umenzel  R       0:16      1 sens2019016-b21
4443      node LIV_MULT  umenzel  R       0:16      1 sens2019016-b26
4444      node LIV_MULT  umenzel  R       0:16      1 sens2019016-b28
4445      node LIV_MULT  umenzel  R       0:16      1 sens2019016-b30
4446      node LIV_MULT  umenzel  R       0:16      1 sens2019016-b32
```

cojo_chr

- you need not to care about this (sub-)program !
- chromosomes without significant markers ($p \leq 5 \cdot 10^{-8}$) are not run through cojo.

```
[umenzel@sens2019016-bianca LIV_MULT4]$ more LIV_MULT4_liv2_cojo_chrom1.log

  Mon Mar 23 11:36:39 CET 2020
  Job identifier:  LIV_MULT4
  Genotype identifier: MF
  Starting job for chromosome 1
  Summary statistics: LIV_MULT4_liv2_cojo.ma
  GCTA-COJO p-value: 5.0e-8
  GCTA-COJO window: 5000
  List of sign. markers loaded: LIV_MULT4_liv2_gwas_signif.txt
  Output file prefix: LIV_MULT4_liv2_cojo_chr1


  No significant marker on chromosome 1 according to the list LIV_MULT4_liv2_gwas_sig
  Cancelling analysis of this chromosome.

  Mon Mar 23 11:36:39 CET 2020
```

cojo_chr

- chromosomes with just one significant marker ($p \leq 5 \cdot 10^{-8}$) are not run through cojo.
  - (depending on the parameter ignore_single_hits in ~/cojo_settings.sh

```
Mon Mar 23 11:36:52 CET 2020
Job identifier:  LIV_MULT4
Genotype identifier: MF
Starting job for chromosome 10
Summary statistics: LIV_MULT4_liv2_cojo.ma
GCTA-COJO p-value: 5.0e-8
GCTA-COJO window: 5000
List of sign. markers loaded: LIV_MULT4_liv2_gwas_signif.txt
Output file prefix: LIV_MULT4_liv2_cojo_chr10


Just one significant marker on chromosome 10 according to the list LIV_MULT4_liv2_gwas_
This marker is independent and being added to the output list "LIV_MULT4_liv2_cojo_chr1

Mon Mar 23 11:36:52 CET 2020
```

cojo_collect

- you need not to care about this (sub-)program !
- collects results from all chromosomes
- output file: *.jma = significant and independent markers

```
[umenzel@sens2019016-bianca LIV_MULT5]$ head LIV_MULT5_liv2_cojo.jma
ID      CHR     POS     OTHER   A1      A1_FREQ OBS_CT  BETA    SE      P
rs532548475_G_A 2       13807712        A       G       0.000599056     17984.6 5.44971 0.980691        2.74435e-08
rs577713064_G_C 2       46305393        C       G       0.000906995     18368.2 4.71597 0.788764        2.24604e-09
rs555658286_A_T 2       77127642        T       A       0.000809429     16422.9 4.83488 0.882974        4.35836e-08
rs559097503_G_A 2       113003928       A       G       0.000913064     16673.7 4.57242 0.82512 2.9986e-08
rs539575470_T_C 2       174472518       C       T       0.000702046     16823.4 5.14191 0.936694        4.03254e-08
rs760790704_G_A 2       189415103       A       G       0.000588085     16416.1 6.11306 1.036   3.62072e-09
rs193273071_T_C 2       236592939       C       T       0.000827696     16725   4.83546 0.865261        2.29121e-08
rs768127167_A_T 3       53460492        T       A       0.000566038     16860.3 5.94079 1.04196 1.18737e-08
rs184010787_G_C 3       74071228        C       G       0.00396593      18562.8 2.16206 0.375798        8.75443e-09
```

cojo_clean

- you need not to care about this (sub-)program !
- collects warnings and errors from all logfiles (*.log) of the current project
- deletes some files
- tars some other files (.tar.gz) to save disk space

Pruning II

http://zzz.bwh.harvard.edu/plink/clump.shtml

LD-based result clumping procedure

This page describes PLINK's ability to group SNP-based results across one or more datasets or analyses, based on empirical estimates written by Ben Voight.

There are probably two main applications for this method:

- To report the top *X* single SNP results from a genome-wide scan in terms of a smaller number of *clumps* of correlated SNPs (i.e. to
- To provide a quick way to combine sets of results from two or more studies, when the studies might also be genotyped on different

Basic usage for LD-based clumping

The --clump command is used to specify one or more result files (i.e. precomputed analyses of some kind). By default, PLINK scans th

```
plink --file mydata --clump mytest1.assoc
```

which generates a file

```
plink.clumped
```

The actual genotype dataset specified here (i.e. the mydata.* fileset) may or may not be the same dataset that was used to generate th disequilibrium between the SNPs that feature in mytest1.assoc (i.e. the analyses are not re-run).

There are four main parameters that determine the level of clumping, listed here in terms of the command flag used to change them and th

```
--clump-p1 0.0001          Significance threshold for index SNPs

--clump-p2 0.01            Secondary significance threshold for clumped SNPs

--clump-r2 0.50            LD threshold for clumping

--clump-kb 250             Physical distance threshold for clumping
```

run_clump

```
[umenzel@sens2019016-bianca LIV_MULT5]$ run_clump

  Usage: run_clump
         -i|--id <string>              no default
         -pn|--phenoname <string>      defaults to all gwas results
         -p1|--p1 <real>               /home/umenzel/clump_settings.sh
         -p2|--p2 <real>               /home/umenzel/clump_settings.sh
         -r2|--r2 <real>               /home/umenzel/clump_settings.sh
         -kb|--kb <integer>            /home/umenzel/clump_settings.sh
         -m|--minutes <int>            /home/umenzel/clump_settings.sh
```

- with plink1.9 (no clumping in plink2.0)
- on .bed .bim .fam genotype files
- run time < 1 min. (chromosome) [not wallclock time!]
- defaults: ~/clump_settings.sh
- reads also parameters from parameter file (see above)
  - 3rd method to acquire parameters (besides ~/clumps_settings.sh and command line)

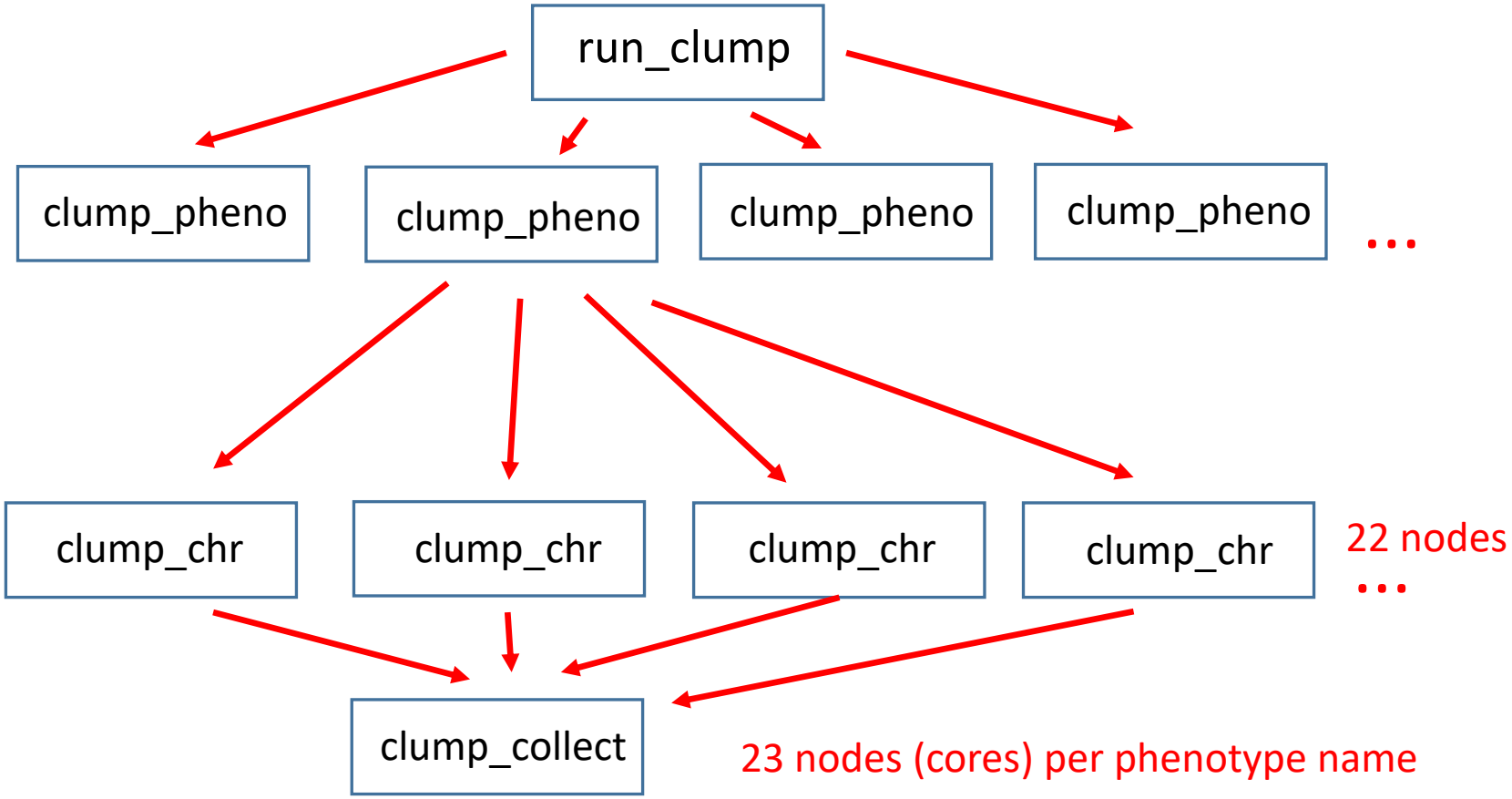
run_clump

Call

With mandatory command line parameters only:

```
run_clump --id LIV_MULT5 --phenoname liv2,liv8   # two phenotypes
```

- start from within GWAS folder (folder created by run_gwas)
- all other parameters are read from ~/clump_settings.sh
- runs phenotypes and chromosomes in parallel, uses 23 * #phenotypes nodes or cores, see below)



- Trick: sleep_between_pheno=100 (to be set in clump_settings.sh)
  - pauses 100 minutes between phenotypes (or whatever number you enter)
- Emergency break: scancel –u <username>

clump_collect

- you need not to care about this (sub-)program !
- collects results from all chromosomes:
- output file: *.jma

```
[umenzel@sens2019016-bianca LIV_MULT5]$ head LIV_MULT5_liv8_clump.jma
ID      CHR     POS     OTHER   A1      A1_FREQ OBS_CT  BETA    SE      P
rs185378124_A_C 2       46310348        C       A       0.00222024      18771   2.9526  0.51766 1.19e-08
rs193273071_T_C 2       236592939       C       T       0.000827696     18771   4.83546 0.864479        2.26e-08
rs532548475_G_A 2       13807712        A       G       0.000599056     18771   5.44971 0.979876        2.71e-08
rs539575470_T_C 2       174472518       C       T       0.000702046     18771   5.14191 0.935883        3.98e-08
rs555658286_A_T 2       77127642        T       A       0.000809429     18771   4.83488 0.882194        4.3e-08
rs559097503_G_A 2       113003928       A       G       0.000913064     18771   4.57242 0.824385        2.95e-08
rs577713064_G_C 2       46305393        C       G       0.000906995     18771   4.71597 0.788018        2.21e-09
rs760790704_G_A 2       189415103       A       G       0.000588085     18771   6.11306 1.03493 3.55e-09
rs183683247_T_C 3       74126342        C       T       0.00136597      18771   3.86751 0.674966        1.02e-08
```

- same format as for cojo
- same suffix as for cojo

clump_chr

- you need not to care about this (sub-)program
- clumps a single chromosome
- issue with sorting (one marker only, connected to sort order of "_" and character):

```
:join: /dev/fd/62:6850: is not sorted: 10:12665040_TTCCC_T      T       TTCCC   0.
:join: /dev/fd/62:23115: is not sorted: 13:96167434_TTTG_T      TTTG    T       0.
:join: /dev/fd/62:7018: is not sorted: 15:52543838_TTTTA_T      T       TTTTA   0.
:join: /dev/fd/62:4354: is not sorted: 16:27839177_TTCCATCCA_T  T       TTCCATCCA
:join: /dev/fd/62:1977: is not sorted: 19:16604540_TATA_T       T       TATA    0.
:join: /dev/fd/62:3877: is not sorted: 20:25456265_TG_T T       TG      0.0376941
```

```
LIV_MULT5_liv2_clump_chrom10.log
LIV_MULT5_liv2_clump_chrom13.log
LIV_MULT5_liv2_clump_chrom15.log
LIV_MULT5_liv2_clump_chrom16.log
LIV_MULT5_liv2_clump_chrom19.log
LIV_MULT5_liv2_clump_chrom20.log
LIV_MULT5_liv2_clump_chrom22.log
```

Parameter file

- parameter file: ${ident}_gwas_params.txt

```
[umenzel@sens2019016-bianca LIV_MULT5]$ cat LIV_MULT5_gwas_params.txt
plink2_version plink2/2.00-alpha-2-20190429
workfolder /proj/sens2019016/GWAS_TEST/LIV_MULT5
ident LIV_MULT5
cstart 1
cstop 22
genotype_id MF
phenofile liver_fat_faked.txt
phenoname liv1,liv2,liv3,liv4,liv5,liv6,liv7,liv8,liv9,liv10
covarfile GWAS_covariates.txt
covarname PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10,array,sex,age
mac 30
hwe_pval 1.0e-6
machr2_low 0.8
machr2_high 2.0
cojo_out liv2 LIV_MULT5_liv2_cojo.jma
cojo_out liv10 LIV_MULT5_liv10_cojo.jma
clump_out liv5 LIV_MULT5_liv5_clump.jma
clump_out liv8 LIV_MULT5_liv8_clump.jma
clump_out liv2 LIV_MULT5_liv2_clump.jma
```

- complemented by subsequent scripts (here with cojo and clump data)

review_gwas

```
[umenzel@sens2019016-bianca LIV_MULT5]$ review_gwas

  Usage: review_gwas
         -i|--id <string>                no default
         -pn|--phenoname <string>        no default
         -c|--chr <int>[-<int>]          1-22
         -m|--minutes <int>              60
```

- writes html : ${ident}_${phenoname}_report.html
  - with embedded plots
  - Spreadsheet must be moved separately to make the link functional

- default parameters:
  - gwas_settings.sh
  - gwas_settings.R

A ”light”-version will be there soon!

review_gwas

Call

With mandatory command line parameters only:

```
review_gwas --id LIV_MULT5 --phenoname liv2          # just one phenotype allowed
```

- start from within GWAS folder (created by run_gwas)
- only one phenotype name (but you can of course start multiple review_gwas)
- all other parameters are read from ~/review_settings.sh & ~/review_settings.R

archive_gwas & retrieve_gwas

```
[umenzel@sens2019016-bianca LIV_MULT5]$ archive_gwas

  Usage: archive_gwas
         -i|--id <string>                    no default
```

```
[umenzel@sens2019016-bianca GWAS_TEST]$ retrieve_gwas

  Usage: retrieve_gwas
         -i|--id <string>                   no default
```

- creates ${ident}.tar.gz from the whole project folder (created by run_gwas)
  - .jma files kept separately (main reults)
  - parameter file kept separately
  - file with significant markers kept separately
- defaults: ~/archive_settings.sh
- run in sbatch: calling tar_gwas.sh and untar_gwas.sh, respectively

archive_gwas; retrieve_gwas

Call

The only command line parameter is the ID:

```
archive_gwas --id LIV_MULT3
```

```
retrieve_gwas --id LIV_MULT3
```

- start from outside GWAS folder ! (the whole folder is compressed)
- all other parameters are read from ~/archive_settings.sh

Program hierarchy

- run_gwas
  - gwas_chr (a single chromosome, multiple phenotypes)
  - parameters: ~/gwas_settings.sh
- run_cojo
  - cojo_pheno (a single phenotype)
    - cojo_convert (format conversion)
    - cojo_chr (a single chromosome)
    - cojo_collect (collect results for chromosomes)
      - cojo_allele.R (get ”other” allele)
    - cojo_clean (delete & gzip)
- run_clump
  - clump_pheno
    - clump_chr
    - clump_collect
- review_gwas (.sh)
  - review_gwas.R
  - link_nearest_gene.R
  - manhattan.plot.R
  - gwas_report.Rmd
- archive_gwas.sh & retrieve_gwas.sh
  - tar_gwas.sh & untar_gwas.sh

Settings files

- must reside in your home directory (cd ~)
- ~/gwas_settings.sh (sourced by run_gwas and gwas_chr, bash syntax)
- ~/clump_settings.sh (sourced by run_clump , bash syntax)
- ~/cojo_settings.sh (sourced by run_cojo , bash syntax)
- ~/review_settings.sh (sourced by review_gwas.sh , bash syntax)
- ~/review_settings.R (OOPS: R syntax!)
- ~/archive_settings.sh (sourced by archive_gwas.sh, retrieve_gwas.sh)

The whole test pass

```
## GWAS

  run_gwas --id LIV_MULT5 --phenofile liver_fat_faked.txt \
           --phenoname liv1,liv2,liv3,liv4,liv5,liv6,liv7,liv8,liv9,liv10

## cojo

  run_cojo --id LIV_MULT5 --phenoname liv2,liv10          # two phenotypes

## clump

  run_clump --id LIV_MULT5 --phenoname liv2,liv8          # two phenotypes

## review

  review_gwas --id LIV_MULT5 --phenoname liv2             # just one phenotype allowed

## archive

  archive_gwas --id LIV_MULT3

## retrieve

  retrieve_gwas --id LIV_MULT3
```