

Association Testing Draft : **meta**

H3Agwas Association Testing Pipeline

Tue Jul 12 09:54:41 SAST 2022

1 Introduction

This report gives a brief overview of the run of the association testing pipeline.

- You were testing for the following phenotypes **meta**
- You were using the following covariates []

Table 1 The top 10 SNPs found by GWAMA analysis

SNP	Beta	P
10_112678657_T_G	-15.4044	8.460E-60
1_117532790_T_C	-14.8600	3.150E-55
12_31367856_C_A	9.4109	1.430E-17
2_45832137_G_A	13.4128	6.720E-16
17_78757626_G_A	12.5427	3.970E-14
17_78699863_T_C	9.2614	2.430E-12
17_78755764_G_A	10.3777	1.470E-11
17_78562921_G_A	9.0154	4.670E-11
17_78727734_T_C	10.9312	5.940E-11
17_78705292_G_A	10.3525	9.290E-11

2 Result of meta analysis : GWAMA

All the results from the GWAMA analysis can be found in the **GWAMA** directory. The result of the GWAMA analysis is shown for program **GWAMA**. The file with association statistics is found in `gwama_res.out`. The top 10 results are shown in Table 1:

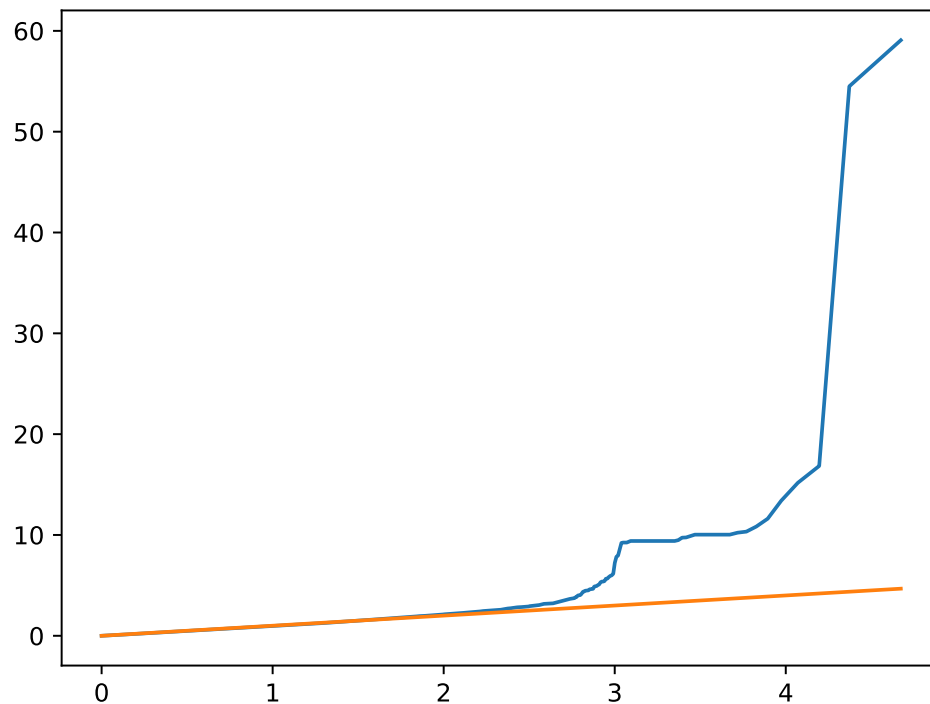
Figure 1 GWAMA testing: QQ-plot for GWAMA

Table 2 The top 10 SNPs found by Metal analysis

SNP	Beta	P
10_112678657_T_G	-14.3750	7.478E-47
1_117532790_T_C	13.9540	2.971E-44
17_78757626_G_A	9.3420	9.497E-21
12_31367856_C_A	8.0170	1.080E-15
2_45832137_G_A	7.6360	2.241E-14
17_78728813_G_A	-7.5880	3.249E-14
17_78727734_T_C	7.5300	5.056E-14
17_78714793_T_C	-7.3920	1.445E-13
17_78716417_G_A	7.3920	1.445E-13
17_78716122_G_A	-7.3920	1.445E-13

3 Result of meta analysis : Metal

All the results from the Metal analysis can be found in the **Metal** directory. The result of the Metal analysis is shown for program **Metal**. The file with association statistics is found in **metal_res1.stat**. The top 10 results are shown in Table 2:

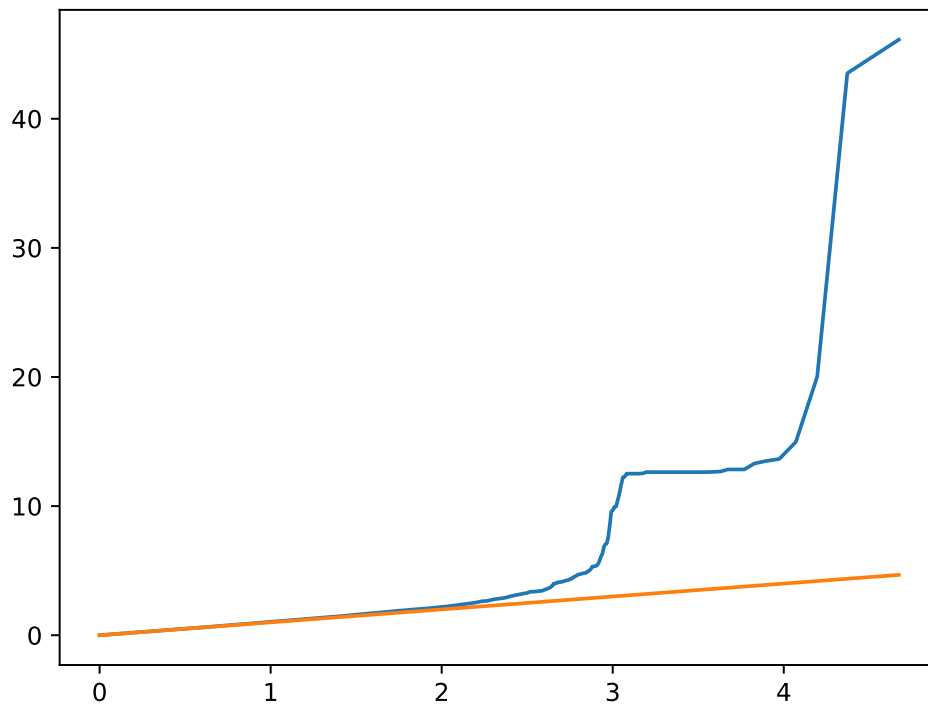
Figure 2 Metal testing: QQ-plot for Metal

Table 3 The top 10 SNPs found by MetaSoft (Han and Eskin Random Effects model) analysis

SNP	Beta	P
10_112678657_T_G	-15.4044	7.547E-60
1_117532790_T_C	-14.8600	2.822E-55
12_31367856_C_A	9.4109	1.351E-17
2_45832137_G_A	13.4128	6.392E-16
17_78757626_G_A	12.5427	3.795E-14
17_78699863_T_C	9.2614	2.337E-12
17_78755764_G_A	10.3777	1.418E-11
17_78562921_G_A	9.0154	4.509E-11
17_78727734_T_C	10.9312	5.742E-11
17_78699819_T_C	10.3525	8.983E-11

4 Result of meta analysis : MetaSoft (Han and Eskin Random Effects model)

All the results from the MetaSoft (Han and Eskin Random Effects model) analysis can be found in the **MetaSoft (Han and Eskin Random Effects model)** directory. The result of the MetaSoft (Han and Eskin Random Effects model) analysis is shown for program `MetaSoft(HanandEskinRandomEffectsmodel)`. The file with association statistics is found in `metasoft_res.format.res`. The top 10 results are shown in Table 3:

Figure 3 MetaSoft (Han and Eskin Random Effects model) testing: QQ-plot for MetaSoft (Han and Eskin Random Effects model)

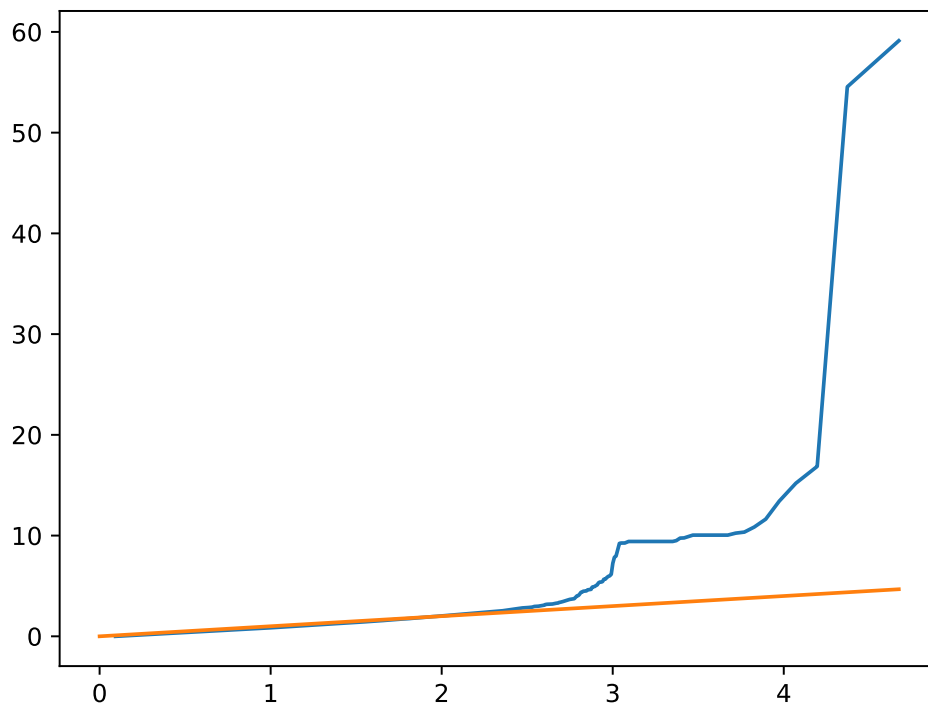


Table 4 The top 10 SNPs found by MRMEGA analysis

SNP	Beta	P
17_78757626_G_A	49.2142	-8.674E-19
1_117532790_T_C	120.4800	2.602E-18
10_112678657_T_G	131.0380	2.602E-18
12_31367856_C_A	35.0731	3.669E-16
2_45832137_G_A	30.8810	2.291E-14
17_78728813_G_A	28.4170	2.594E-13
1_117563582_T_C	28.0765	3.627E-13
17_78727734_T_C	27.9953	3.929E-13
17_78714793_T_C	26.5840	1.575E-12
17_78716417_G_A	26.5840	1.575E-12

5 Result of meta analysis : MRMEGA

All the results from the MRMEGA analysis can be found in the **MRMEGA** directory. The result of the MRMEGA analysis is shown for program MRMEGA. The file with association statistics is found in `mrmega_res.result`. The top 10 results are shown in Table 4:

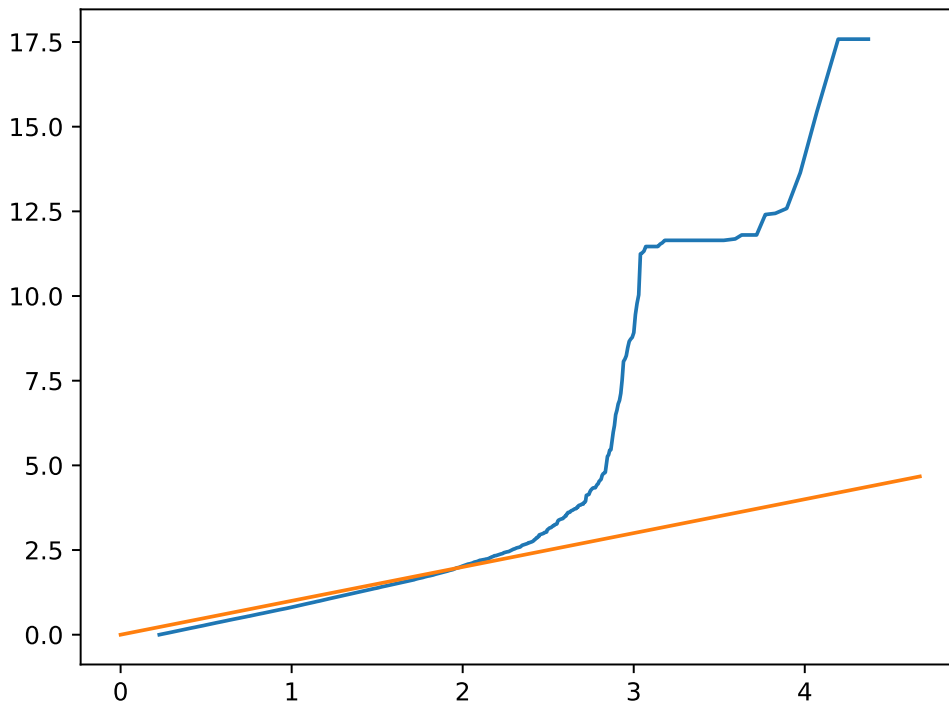
Figure 4 MRMEGA testing: QQ-plot for MRMEGA

Table 5 The top 10 SNPs found by BETA analysis

SNP	Beta	P
10_112678657_T_G	-15.4044	7.547E-60
1_117532790_T_C	-14.8600	2.822E-55
17_78757626_G_A	12.4521	1.044E-23
12_31367856_C_A	9.4161	5.699E-18
2_45832137_G_A	13.4128	6.392E-16
17_78727734_T_C	10.6977	1.662E-14
17_78728813_G_A	10.6109	3.079E-14
17_78716417_G_A	10.3073	8.476E-14
17_78716122_G_A	10.3073	8.476E-14
17_78714793_T_C	10.3073	8.476E-14

6 Result of meta analysis : BETA

All the results from the BETA analysis can be found in the **BETA** directory. The result of the BETA analysis is shown for program **BETA**. The file with association statistics is found in `out_plink.meta`. The top 10 results are shown in Table 5:

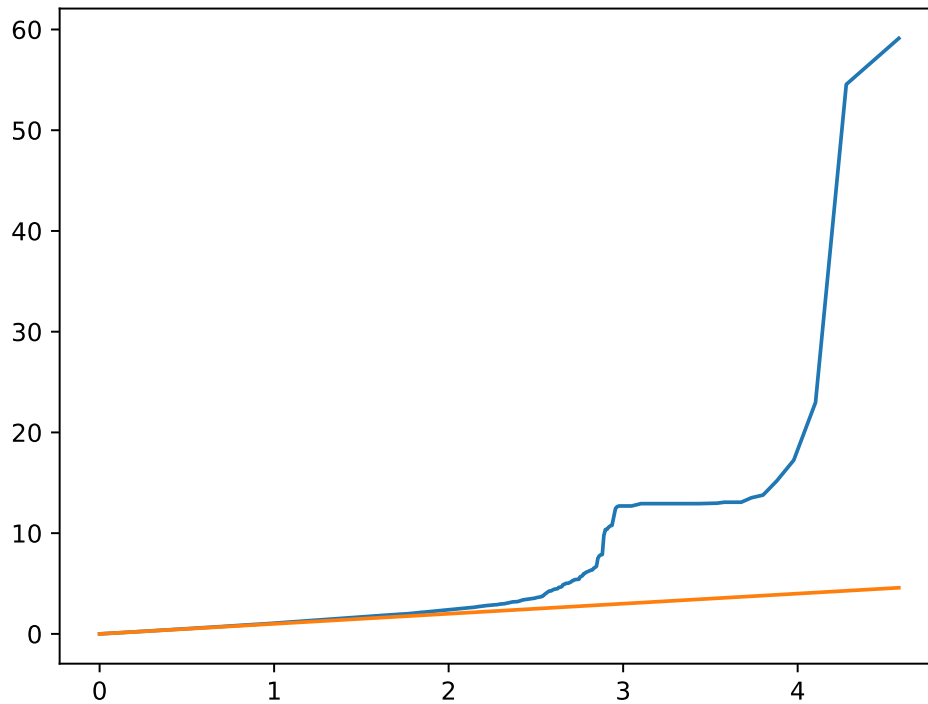
Figure 5 BETA testing: QQ-plot for BETA

Table 6 Docker Images Used

Nextflow process	Docker Image
default	quay.io/h3abionet_org/py3plink

7 Technical details

The analysis and report was produced by the h3aGWAS pipeline (<http://github.com/h3abionet/h3agwas>) produced by the Pan-African Bioinformatics Network for H3Africa (<http://www.h3abionet.org>).

The following tools were used:

- 22.04.3 [Di Tommaso et al, 2017]
- A local copy of the workflow was used
- The command line below was called [NB: if the command line is long, the linebreak may break oddly after a hyphen or dash so take care.]

```
nextflow run /home/jeantristan/Travail/git/h3agwas/meta/meta-assoc.nf --metal 1 --
  gwama 1 --metasoft 1 --mrmega 1 --plink 1 --file_config utils/input_meta.csv -
  resume -profile slurmSingularity --output_dir meta
```

- The profile slurmSingularity was used: the docker images used are found in Table 6
- The full configuration can be found in the appendix.

A Configuration

The Nextflow configuration files are shown below.

A.1 config

```
singularity.cacheDir = "/home/jeantristan/.singularity/"
```

A.2 nextflow

```
plugins {
    id 'nf-azure'
}
```

```
py3Image = "quay.io/h3abionet_org/py3plink"
gemmaImage="quay.io/h3abionet_org/py3plink"
boltImage="quay.io/h3abionet_org/py3fastlmm"
latexImage="quay.io/h3abionet_org/h3agwas-texlive"
py3saige="quay.io/h3abionet_org/py3saige"
gctaImage="quay.io/h3abionet_org/gcta"
MetaAnImage="quay.io/h3abionet_org/py3metagwas"
fastlmmImage="quay.io/h3abionet_org/py3fastlmm"
rImage="quay.io/h3abionet_org/py3rproject"
mtagImage="quay.io/h3abionet_org/py2mtag"
swarmPort = '2376'
queue = 'batch'
```

```
manifest {
    homePage = 'http://github.com/h3abionet/h3agwas'
    description = 'GWAS Pipeline for H3Africa'
    mainScript = 'main.nf'
}
```

```
aws {
    accessKey='*****'
    secretKey='*****'
    region    ='eu-west-1'
}
```

```
cloud {
    imageId = "ami-710b9108"      // specify your AMI id here
    instanceType = "m4.xlarge"
    subnetId = "null"
    sharedStorageId = "null"
    sharedStorageMount = "/mnt/shared"
    bootStorageSize = "20GB"      // Size of disk for images spawned
    // instanceStorageMount = ""   // Set a common mount point for images
    // instanceStorageDevice = ""  // Set a common block device for images
    autoscale {
        enabled = true
        maxInstances = 1
        terminateWhenIdle = true
    }
}
```

```
params {

    // Directories
    work_dir          = "$PWD"
    input_dir         = ""
    output_dir        = "${params.work_dir}/output"
```



```

scripts          = "${params.work_dir}/scripts"
output           = "out"

max_forks        = 95
// Data
phenotype = 0

samplesize       = 0 // How many individuals in each genotype report 0=ALL

accessKey='*****'
secretKey='*****'
  region      = "eu-west-1"
  AMI         = "ami-710b9108"
  instanceType = "m4.xlarge"
  bootStorageSize = "20GB"
  maxInstances = "1"

  plink_mem_req   = "750MB"
  other_mem_req   = "750MB"
  big_time        = '1000h'
  sharedStorageMount = "/mnt/shared"
  max_plink_cores = 4
}
profiles {

  // For execution on a local machine, no containerization. -- Default
  standard {
    process.executor = 'local'
  }

  slurm {
    process.executor = 'slurm'
    process.queue = queue
  }

  awsbatch {
    process.executor = "awsbatch"
    aws.region      = 'us-east-1'
    aws.uploadStorageClass = 'ONEZONE_IA'
    process.queue = 'h3a'
  }

  // Execute pipeline with Docker locally
  docker {
    process.executor = 'local'
    docker.remove    = true
    docker.runOptions = '--rm'
    docker.registry   = 'quay.io'
    docker.enabled    = true
    docker.temp       = 'auto'
    docker.fixOwnership= true
    docker.process.executor = 'local'
  }

  // Execute pipeline with Docker Swarm setup
  dockerSwarm {
    docker.remove    = true
    docker.runOptions = '--rm'
  }
}

```

```

    docker.registry      = 'quay.io'
    docker.enabled       = true
    docker.temp          = 'auto'
    docker.fixOwnership= true
    docker.process.executor = 'local'
    docker.engineOptions = "-H :$swarmPort"
}

// For execution on a PBS scheduler, no containerization.
pbs {
    process.executor = 'pbs'
    process.queue = queue
}

// For execution on a PBS scheduler with containerization.
pbsDocker {

    process.executor = 'pbs'
    docker.remove    = true
    docker.runOptions = '--rm'
    docker.registry  = 'quay.io'
    docker.enabled   = true
    docker.temp      = 'auto'
    docker.fixOwnership= true
}

// For execution on a SLURM scheduler, no containerization.
slurm {
    process.executor = 'slurm'
    process.queue = queue
}

// For execution on a PBS scheduler with containerisation.
slurmDocker {
    process.executor = 'slurm'
    docker.remove    = true
    docker.runOptions = '--rm'
    docker.registry  = 'quay.io'
    docker.enabled   = true
    docker.temp      = 'auto'
    docker.fixOwnership= true
}

singularity {
    singularity.cacheDir = "${HOME}/.singularity"
    singularity.autoMounts = true
    singularity.enabled = true
    process.executor = 'local'
}

// For execution on a SLURM scheduler with singularity
slurmSingularity {
    singularity.cacheDir = "${HOME}/.singularity"
    process.executor = 'slurm'
    singularity.autoMounts = true
    singularity.enabled = true
    singularity.runOption = "--cleanenv"
    process.queue = queue
}

```

```

slurmDocker {
    process.executor = 'slurm'
    process.queue = queue
    docker.remove    = true
    docker.runOptions = '--rm'
    docker.registry  = 'quay.io'
    docker.enabled   = true
    docker.temp      = 'auto'
    docker.fixOwnership= true
}

// For execution on a PBS scheduler, no containerization.
pbs {
    process.executor = 'pbs'
    process.queue = queue
}

// For execution on a PBS scheduler with containerization.
pbsDocker {

    process.executor = 'pbs'
    docker.remove    = true
    docker.runOptions = '--rm'
    docker.registry  = 'quay.io'
    docker.enabled   = true
    docker.temp      = 'auto'
    docker.fixOwnership= true
}

// For execution on Azure Batch
azurebatch {
    process.executor = 'azurebatch'
}

}

```

```

process {
    container = py3Image

    withLabel:bigMem {
        memory = '8GB'
    }

    withLabel: gemma {
        container = gemmaImage
    }

    withLabel: latex {
        container = latexImage
    }
    withLabel: gcta{
        container = gctaImage
    }
    withLabel: metaanalyse {
        container = MetaAnImage
    }
    withLabel: bolt {
        container = boltImage
    }
    withLabel: R{
        container = rImage
    }
}

```

```
    withLabel : fastlmm{
        container = fastlmmImage
    }
    withLabel : saige{
        container = py3saige
    }
    withLabel : mtag{
        container = mtagImage
    }
}

//timeline {
//    enabled=true
//    file = "nextflow_reports/timeline.html"
//}

//report {
//    enabled = true
//    file = "nextflow_reports/report.html"
//}
```