

Association Testing Draft : SEX, PHE

H3Agwas Association Testing Pipeline

Tue Jan 22 16:00:12 UTC 2019

1 Introduction

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

- You were testing for the following phenotypes SEX, PHE
- You were using the following covariates [PAT, MAT]

2 PLINK Results for phenotype PHE, test assoc

The raw results can be found in the **assoc** directory. The QQ plot can be found in Figure 1, and the Manhattan plot in Figure 2.

Genomic inflation: : The estimate of genomic inflation λ (based on median chisq) = 1.06221.

- The top ranking SNPs for EMP2 according to permutation testing are found in Table 3
- The top ranking SNPs for BONF according to Bonferroni correction are found in Table 2

Table 1 Top SNPs found by plink using permutation testing for EMP2 test PHE

SNP	P
-----	---

Table 2 Top SNPs found by plink using Bonferroni correction for BONF test PHE

SNP	P
-----	---

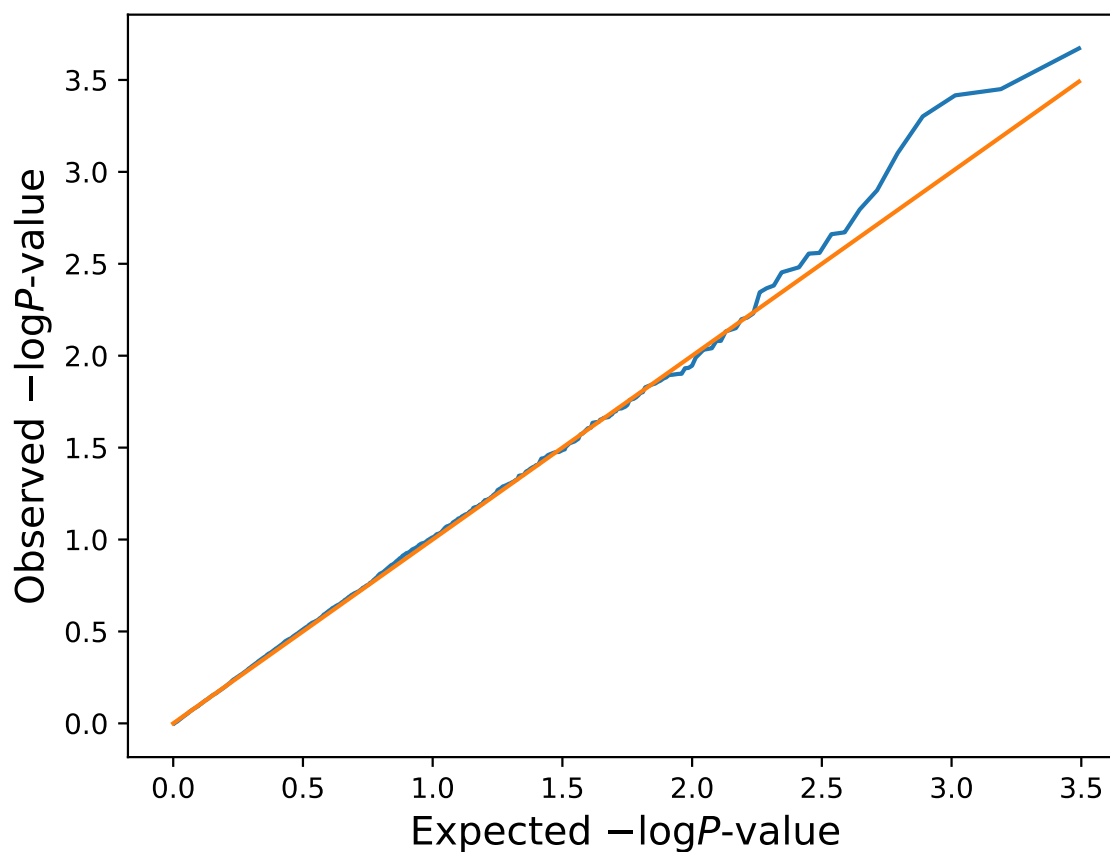
Figure 1 QQ plot for PLINK testing – PHE, test assoc – PHE.assoc

Figure 2 PLINK testing: Manhattan plot for - PHE, test assoc - PHE.assoc

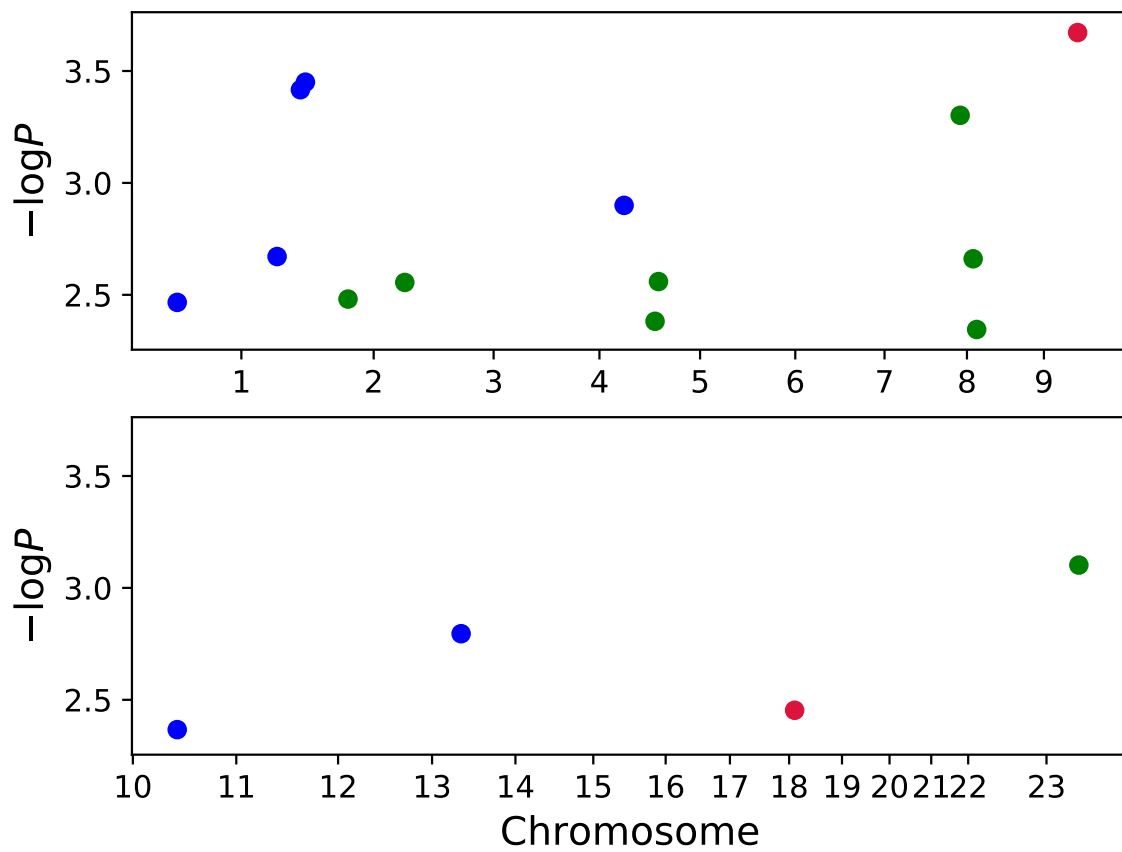


Table 3 Top SNPs found by plink using permutation testing for EMP2 test PHE

SNP	P
-----	---

3 PLINK Results for phenotype PHE, test logistic

The raw results can be found in the **logistic** directory. The QQ plot can be found in Figure 3, and the Manhattan plot in Figure 4.

- The top ranking SNPs for EMP2 according to permutation testing are found in Table 3

Figure 3 QQ plot for PLINK testing – PHE, test logistic – PHE.assoc.logistic

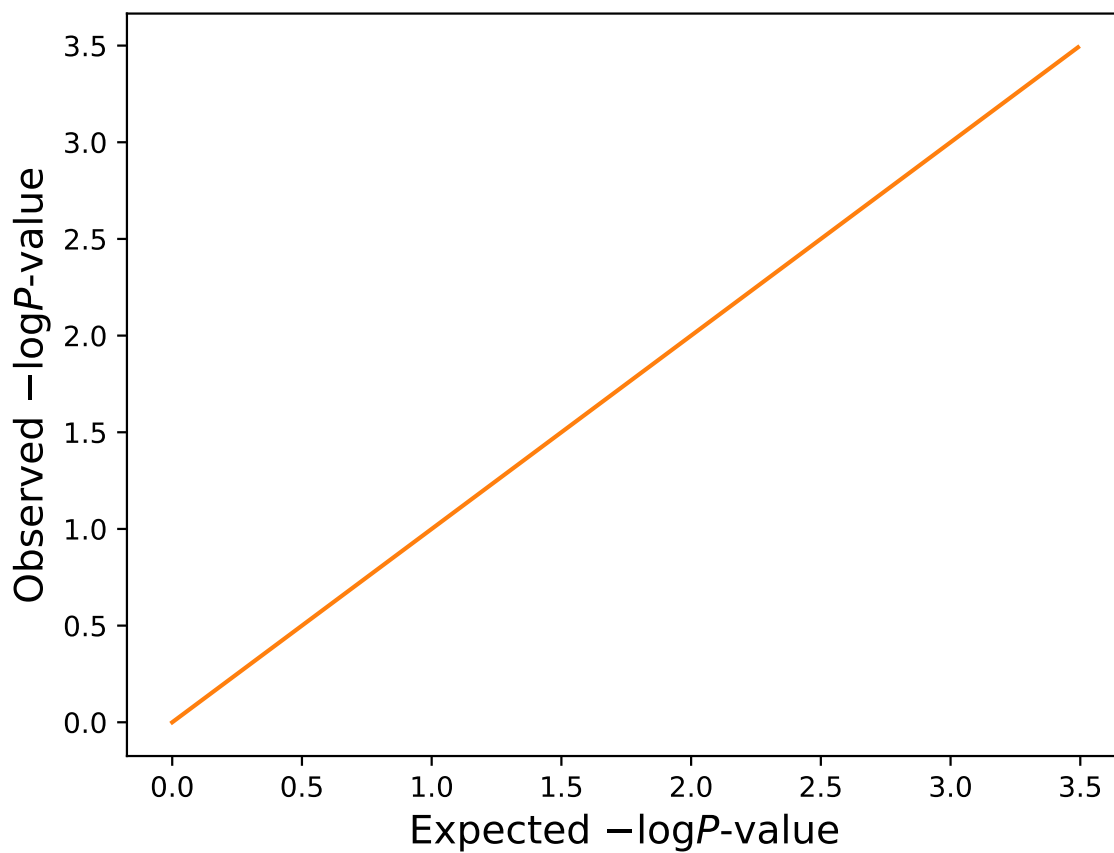


Figure 4 PLINK testing: Manhattan plot for - PHE, test logistic - PHE.assoc.logistic

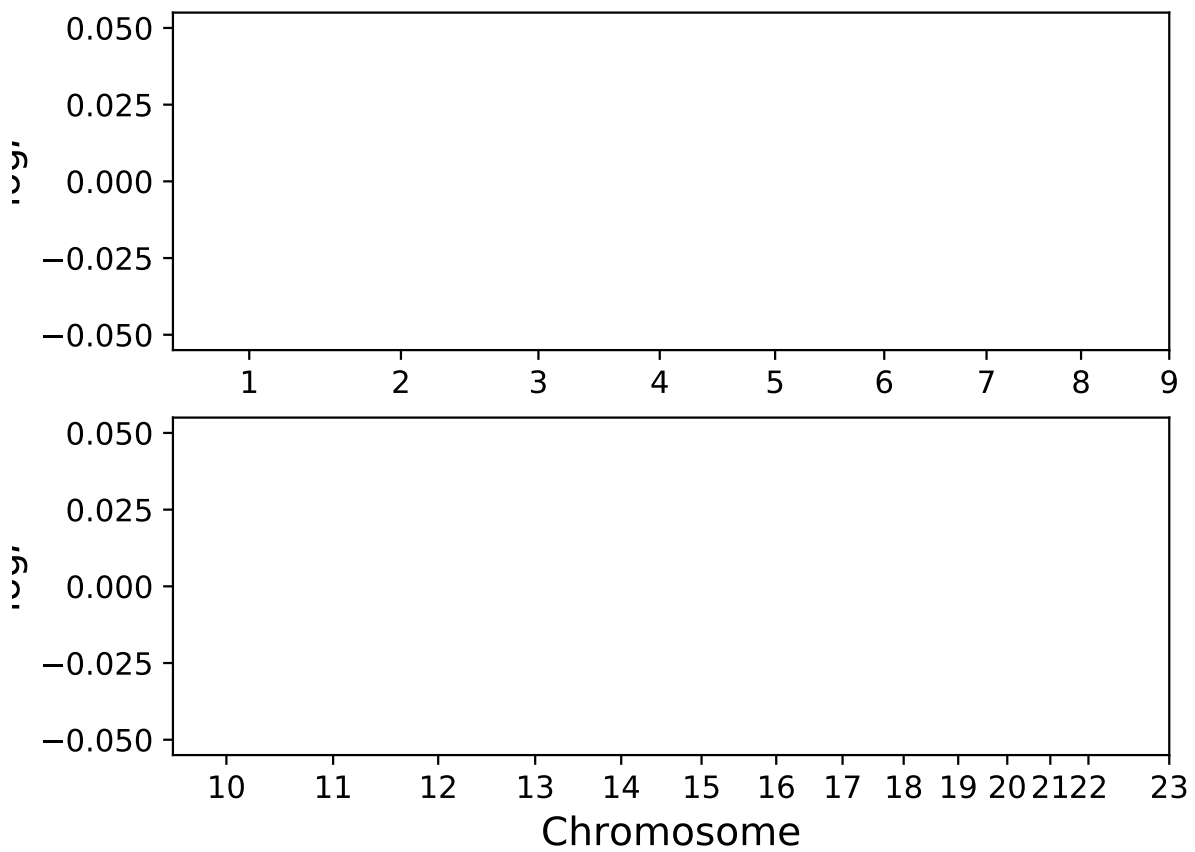


Table 4 Top SNPs found by plink using permutation testing for EMP2 test SEX

SNP	P
rs981155	0.0070

Table 5 Top SNPs found by plink using Bonferroni correction for BONF test SEX

SNP	P
rs981155	0.0055

4 PLINK Results for phenotype SEX, test assoc

The raw results can be found in the `assoc` directory. The QQ plot can be found in Figure 5, and the Manhattan plot in Figure 6.

Genomic inflation: : The estimate of genomic inflation λ (based on median chisq) = 1.

- The top ranking SNPs for EMP2 according to permutation testing are found in Table 6
- The top ranking SNPs for BONF according to Bonferroni correction are found in Table 5

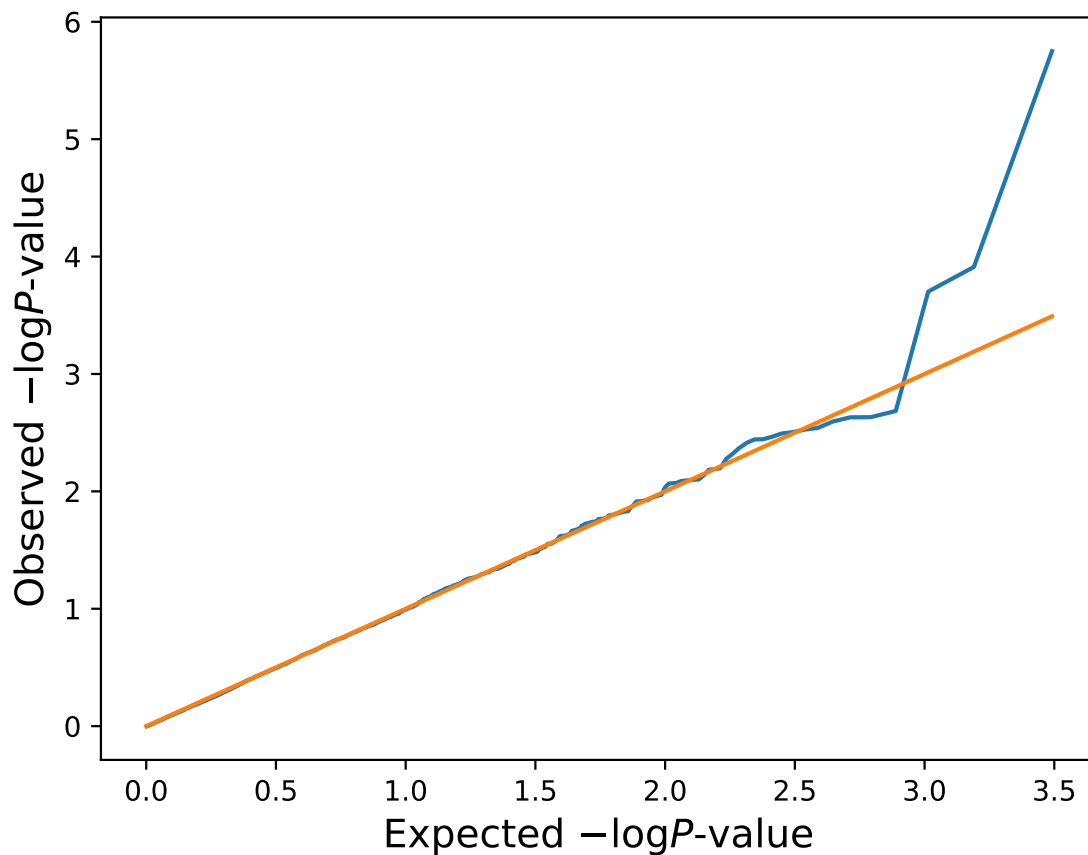
Figure 5 QQ plot for PLINK testing – SEX, test assoc – SEX.assoc

Figure 6 PLINK testing: Manhattan plot for `-SEX`, test assoc `-SEX.assoc`

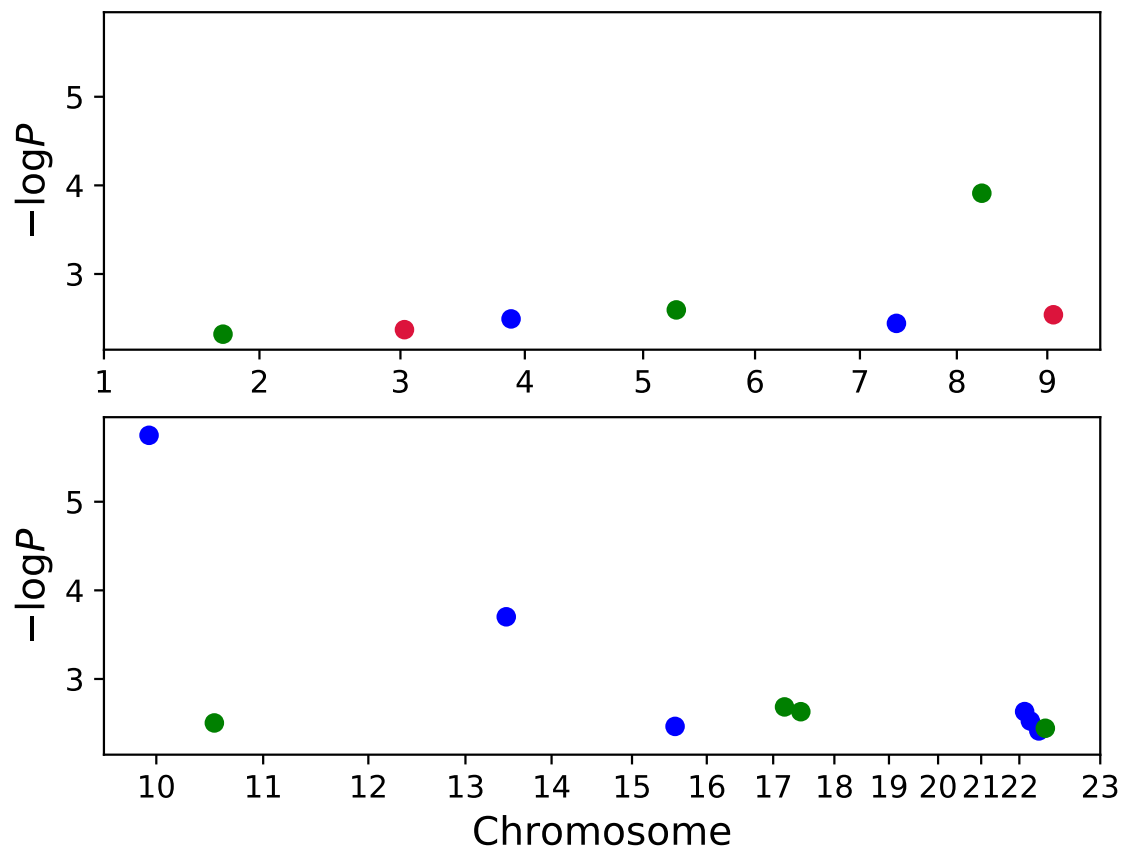


Table 6 Top SNPs found by plink using permutation testing for EMP2 test SEX

SNP	P
-----	---

5 PLINK Results for phenotype SEX, test logistic

The raw results can be found in the **logistic** directory. The QQ plot can be found in Figure 7, and the Manhattan plot in Figure 8.

- The top ranking SNPs for EMP2 according to permutation testing are found in Table 6

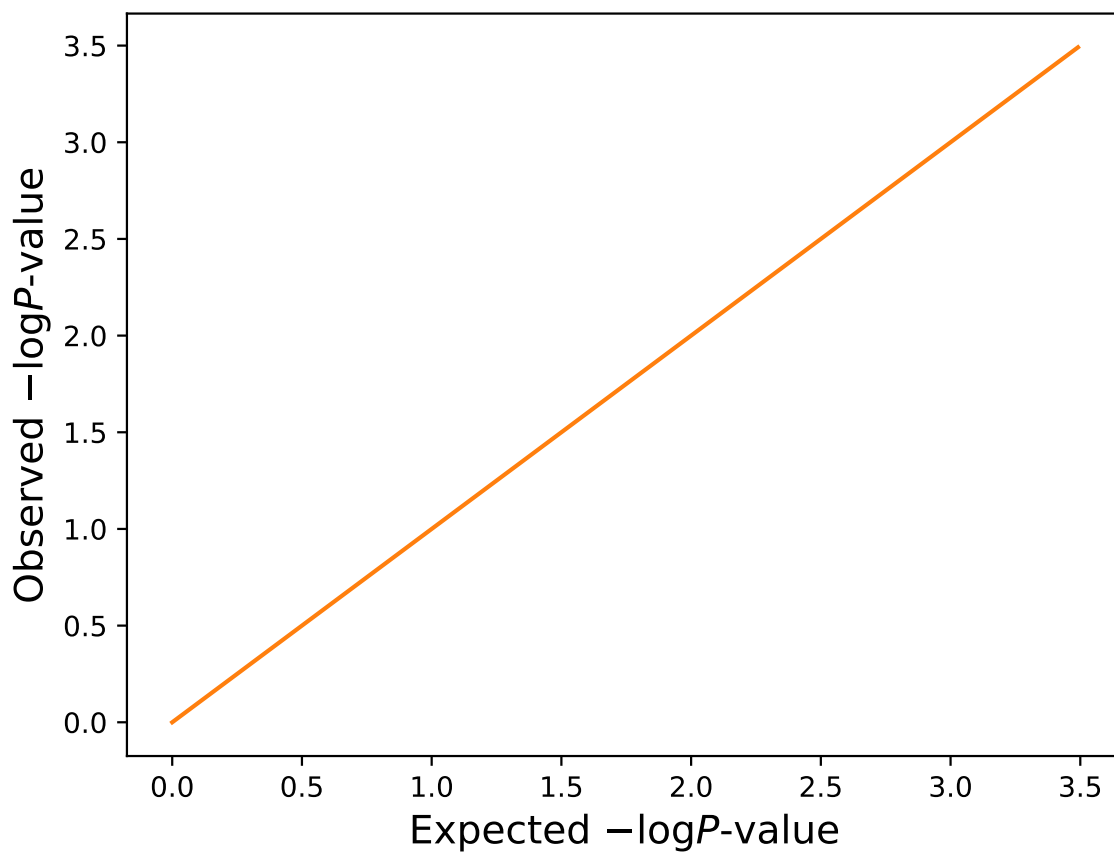
Figure 7 QQ plot for PLINK testing – SEX, test logistic – SEX.assoc.logistic

Figure 8 PLINK testing: Manhattan plot for - SEX, test logistic - SEX.assoc.logistic

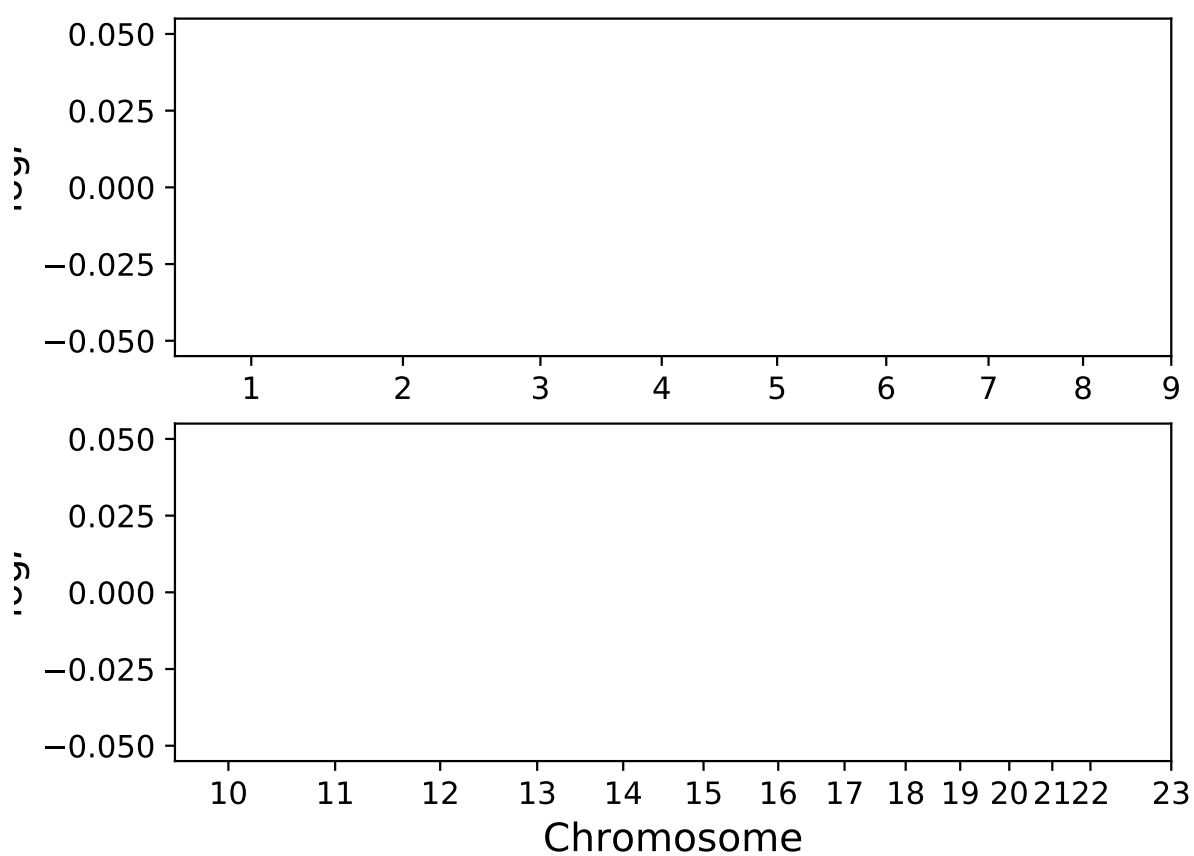


Table 7 Docker Images Used

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

6 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:

- nextflow version 18.10.1.5003 [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 plink-assoc.nf --input_dir sample --input_pat sampleA --chi2 1 --logistic
  1 --adjust 1 --data sample/sample.phe --pheno SEX,PHE --covariates PAT,MAT -
profile docker
```

- The profile docker was used: the docker images used are found in Table 7
- The full configuration can be found in the appendix.

A Configuration

The Nextflow configuration files are shown below.

A.1 nextflow

```

py3Image = "quay.io/h3abionet_org/py3plink"
gemmaImage="quay.io/h3abionet_org/h3agwas-gemma"
latexImage="quay.io/h3abionet_org/h3agwas-texlive"
swarmPort = '2376'
queue = 'batch'

manifest {
    homePage = 'http://github.com/h3abionet/h3agwas'
    description = 'GWAS Pipeline for H3Africa'
    mainScript = 'plink-qc.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 = "$PWD"
    input_dir = "${params.work_dir}/input"
    output_dir = "${params.work_dir}/output"
    scripts = "${params.work_dir}/scripts"
    output = "out"

    max_forks = 95
    // Data
    input_pat = "sampleA"

    high_ld_regions_fname = ""
    sexinfo_available = true
    cut_het_high = 0.343
    cut_het_low = 0.15
    cut_diff_miss = "0.05"
    cut_maf = "0.01"
}

```

```

cut_mind           = "0.02"
cut_geno           = 0.01
cut_hwe           = 0.008
pi_hat            = 0.11
super_pi_hat      = 0.7
f_lo_male         = 0.8 // default for F-sex check -- >= means male
f_hi_female       = 0.2 // <= means female
case_control       = "${params.input_dir}/sample.phe"
case_control_col   = "PHE"

phenotype = 0
pheno_col = "all"
batch = 0
batch_col = 0

samplesize        = 0 // How many individuals in each genotype report 0=ALL
strandreport      = ""
manifest           = ""
idpat              = 0 // or "(\\w+)-DNA_(\\w+)_.*" or ".*_(.*)"

// Needed for topbottom.nf -- uncomment and put in details
// reference       = ""
// output_align    = ""
// samplesheet     = ""
// chipdescription = ""

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         = '12h'
sharedStorageMount = "/mnt/shared"
gemma_num_cores  = 8
max_plink_cores  = 4

}
profiles {

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

// 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.cacheDir = "${HOME}/.singularity"

singularity {
    singularity.autoMounts = true
    singularity.enabled = true
    process.executor = 'local'
}

// For execution on a SLURM scheduler with singularity
slurmSingularity {

    process.executor = 'slurm'
    singularity.autoMounts = true
    singularity.enabled = true
    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
}

}

process {
    container = py3Image

    withLabel: latex {
        container = latexImage
    }

    withLabel: gemma {
        container = gemmaImage
    }
}

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

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