**Analysis plan - Long COVID**

**COVID-19 Host Genetics Initiative**

**October 2021 - version 1**

Modified for long COVID analyses from the main [COVID-19 Host Genetics Analysis Plan v1.1](https://docs.google.com/document/d/16ethjgi4MzlQeO0KAW_yDYyUHdB9kKbtfuGW4XYVKQg/edit)

**Table of contents**

[**Initial analytic considerations**](#_ph1gwckwigtp) **2**

[**Genetic data types**](#_boylaquxasqd) **2**

[**Checklist**](#_7asujft85e9s) **3**

[**Phenotype definitions**](#_tyvhtow2s6tk) **4**

[**Recommended pipeline and QC parameters**](#_vuoszfacmm0p) **5**

[**GWAS imputation**](#_2nxj9qcyth93) **6**

[**Association analysis**](#_erwxuhdvcegj) **10**

[**Results Quality Checks**](#_940mrpl23cuh) **11**

[**Results format**](#_zcv02hklx4ia) **14**

[Results naming](#_h8vqucuo9xe5) 15

[**Sample summary information**](#_9jyc3fycronl) **17**

[**Results upload instructions**](#_5qjedejw6g70) **18**

[Access to Google Cloud](#_vt7ta78v85ja) 18

[Bucket for uploading data](#_pvg1q09rgmap) 18

[Bucket for downloading data](#_6hn7355xhddp) 19

[**Appendix**](#_eq5fv7hvlfok) **20**

[Analysis instructions for performing GWAS with REGENIE](#_8n481721h7qq) 20

[Analysis instructions for performing GWAS with SAIGE](#_nzwt8m62gcz8) 25

[Generating PC plots for your cohort](#_uvx80nxd7vjl) 30

[Testing male X-chromosome coding in imputed VCFs](#_laz0uhnu4jtl) 32

[Correcting male non-PAR X-chromosome coding in pre-imputed VCFs](#_25h5um4jxhth) 36

## Initial analytic considerations

We would like to promote as much data and results sharing as possible, and so have developed an initial proposal to facilitate these activities. Again, nothing is definitive, but this should serve as a starting point. We are focused on primary association analyses, rather than potential secondary analysis projects.

We recognize that a diverse range of study designs, recruitment and data generation strategies will be pursued to learn more about COVID-19 related outcomes. This diversity of approach is a real strength of this effort, as it will enable a more thorough characterization of all aspects of COVID-19 infection.

The plan is that this is a long-term effort that will develop and be refined as time goes on and as more data becomes available. We understand that low sample sizes and/or case numbers may limit each contributor’s involvement in the initial stages, but hope through continuing data collection and regular data freezes that all contributing studies will be directly involved.

## Genetic data types

This analysis plan includes instructions for performing analysis of common/uncommon variants imputed from array data. As the study grows, we will extend this analysis plan to also include gene-burden tests.

When this project is well-established, we will consider further extending genetic analyses to use whole exome sequencing (WES) or whole genome sequencing (WGS) data, for those studies that (will) have this data available. The COVID-19 HGI has already put together an excellent protocol ([Study Protocol WGS 6.0](https://docs.google.com/document/d/1QrwktjejSH7A1Srzdkyfg4gJi8u4HAFrXPn8XrdX7wo/edit?usp=sharing)) with content more tailored to sequencing data.

## Checklist

* I have a (work-related) Google account. Otherwise check instructions for [creating a Google account](https://support.google.com/accounts/answer/27441) or [linking other email addresses to an existing Google account](https://support.google.com/accounts/answer/176347?co=GENIE.Platform%3DDesktop&hl=en).
* I have registered my study for the COVID-19 HGI Long COVID project by filling in the [COVID\_HG\_LongCOVID\_StudyInfoForm](https://docs.google.com/spreadsheets/d/1XfarBojPob6J9fwJpmGODn3cnX8evDrUeZmT-RI5suk/edit?usp=sharing) (at least the basic information of the study and contact persons, sheet Form, columns A-H).
* I have listed my Google account email in the Study Info Form and been given access to the [Google Cloud bucket for uploading data](#_pvg1q09rgmap).
* I have checked the [phenotype definition file](https://docs.google.com/document/d/1-o5VX83WiWZfBuVi1jkneH3Bz7p9p5Eg/edit).
* I have performed quality control, for example following what is described in “[Recommended pipeline and QC parameters](#3d1fds8nbh71)”.
* I have imputed my genotypes as described in “[GWAS imputation](#goq9en6uxpqn)”.
* I have run association analyses for phenotypes with N.cases > 50 according to the instructions in “[Association analysis](#db7qlymf1abs)” (detailed instructions for running GWAS with REGENIE and SAIGE in the [Appendix](#kykf9ynzisv3)).
* I have run analyses separately for each major ancestry group.
* I have run the checks described in “[Results Quality Checks](#xdn4ltm4wf4y)” to assess my results’ quality
* I have generated the additional descriptive data for my GWAS samples as described in “[Sample summary information](#_9jyc3fycronl)”.
* I have formatted the summary statistics according to the instructions in “[Results format](#h01bfxc978az)” and have named the files according to the naming convention.
* I have filled out information regarding the chromosome build, X chromosome coding and imputation in the [COVID\_HG\_LongCOVID\_StudyInfoForm](https://docs.google.com/spreadsheets/d/1XfarBojPob6J9fwJpmGODn3cnX8evDrUeZmT-RI5suk/edit?usp=sharing), end of sheet “Form”.
* I have uploaded the data according to the instructions in “[Results Upload Instructions](#bfcagfj12ojf)”.
* I have uploaded [Sample summary information](#_9jyc3fycronl) (including PC plots; see instructions in “[Generating PC plots for your cohort](#_uvx80nxd7vjl)”).

## Phenotype definitions

*For any question regarding this section, please contact Vilma Lammi (vilma.lammi@helsinki.fi)*

The phenotypes for analysis are described in the [Phenotype Definitions document](https://docs.google.com/document/d/1-o5VX83WiWZfBuVi1jkneH3Bz7p9p5Eg/edit).

**At least 50 cases are required to participate in the analysis. We also recommend a minimum of 200 controls.**

Sex-specific and/or age-stratified analyses are currently not recommended due to the small sample sizes available in most studies. As the Long COVID effort continues, additional stratified analyses and phenotypes will be introduced. If you have suggestions for additional phenotypes, please raise these as points of discussion at the regular consortium meetings.

### Concerning patients who died during follow-up

## 1: Questionnarie based studies should include all patients who were alive to finish the questionnaries during the follow-up after SARS-CoV-2 infection.

## 2: Electronic health record based studies should include all patients who received a Long-COVID diagnosis, irrespective of whether they died.

## 3: Each study should report the number of dead among cases and controls for each analysis.

## 4: A sensitivity analysis should be performed with patients who died excluded before the final analysis before publication. The time for this will be communicated separately to reduce the total burden of analysis work.

## Recommended pipeline and QC parameters

*(for any question regarding this section, please contact Samuel Jones (*[*samuel.jones@helsinki.fi*](mailto:samuel.jones@helsinki.fi) *- @Samuel Jones (Slack))*

For processing of **array genotype data**, we would highly recommend using the [Ricopili pipeline](https://sites.google.com/a/broadinstitute.org/ricopili/). Recommendations for QC parameter specifics can be found [here](https://sites.google.com/a/broadinstitute.org/ricopili/preimputation-qc#TOC-Technical-Details) and reported below for simplicity. However, every genotyping dataset is slightly different from the other: some tweaking may be required from dataset to dataset.

## 

Our recommended quality control thresholds for **pre-imputation genotypes** are:

* SNP QC: call rate ≥ 0.95 (this criterion is useful when merging case and control datasets from different studies)
* Sample QC: call rate in cases or controls ≥ 0.98
* Sample QC: FHET within +/- 0.20 in cases or controls
* Sample QC: Sex violations (excluded) - genetic sex does not match pedigree sex
* SNP QC: call rate ≥ 0.98
* SNP QC: missing difference ≤ 0.02
* SNP QC: Hardy-Weinberg equilibrium (HWE) in controls p value ≥ 1e-06 (i.e., ≥ log 10(p) of -6)
* SNP QC: Hardy-Weinberg equilibrium (HWE) in cases p value ≥ 1e-10 (i.e., ≥ log 10(p) of -10)
* (HWE step should be done only in females when applying to chromosome X.

Note that **no filter for allele frequency is imposed**: we suggest submitting without using an allele frequency filter.

The QC for chromosome X should be done similarly as the autosomal chromosomes. The only QC that should be differentiated is for Hardy-Weinberg Equilibrium (HWE) test: this test should only be done on females.

## GWAS imputation

*(for any question regarding this section, please contact Samuel Jones (*[*samuel.jones@helsinki.fi*](mailto:samuel.jones@helsinki.fi) *- @Samuel Jones (Slack))*

Input genotypes

Please use imputed genotypes for GWA analyses, both hg19 and hg38 genome versions are permitted. We advise imputing from genotypes captured using a SNP array that has a good imputation “back-bone” (i.e. is designed to contain genetic markers useful for imputation). The analysis plan does not currently include procedures for sequence data, though we hope to add this in the near future. If using the imputation servers, you must input [bgzipped](http://www.htslib.org/doc/bgzip.html) (not gzipped) [VCF files](https://www.ebi.ac.uk/training/online/courses/human-genetic-variation-introduction/variant-identification-and-analysis/understanding-vcf-format/), with a separate file for each chromosome.

If you use [PLINK 1.9](https://www.cog-genomics.org/plink/1.9/) to convert your array genotypes from [PLINK binary format (.bed/.bim/.fam)](https://www.cog-genomics.org/plink/1.9/input#bed) to VCF format for uploading to the imputation servers, we would recommend that you use the flag --recode vcf-iid instead of --recode vcf. This will save effort later when converting the VCFs (output from the imputation pipelines) to other formats, particularly if your sample IDs contain underscores. PLINK is also capable of bgzipping the output VCF, performed by using the modifier bgz with the above recode flag, which is a necessary step before uploading to the imputation server.

**Please ensure that male genotypes in the non-PAR regions of chromosome X (see** [**here**](https://en.wikipedia.org/wiki/Pseudoautosomal_region#Location) **for PAR region boundaries) are coded correctly (either “0” and “1” or “0/0” and “1/1”, not “0/0” and “0/1”) before imputing them**. Incorrect coding will affect the imputation process and will produce unreliable results for the GWAS.

Imputation servers

For genotype imputation, we would recommend the use of the [TopMed imputation server](https://imputation.biodatacatalyst.nhlbi.nih.gov/), which imputes to the latest genome reference (hg38) and uses a large reference panel. Alternatively, the [Michigan imputation server](https://imputationserver.sph.umich.edu/index.html) can be used, if GDPR compliance is critical (the Michigan imputation server has been certified GDPR compliant, whereas the TopMed server is not yet GDPR-certified). Where possible, imputation should be carried out using a reference panel that contains individuals of the same ancestry as your samples in addition to other ancestry groups. Therefore, if the reference panels [used by the TopMed](https://topmed.nhlbi.nih.gov/#Participant%20Diversity) and [Michigan](https://imputationserver.readthedocs.io/en/latest/reference-panels/#reference-panels) servers do not contain reference data from your cohort’s population and you have access to your own population-specific reference panel, it may be worth performing your own imputation using this panel.

If you do use the Michigan imputation server, you have the opportunity to get upgraded in the queue. To do this you will need to email [imputationserver@umich.edu](mailto:imputationserver@umich.edu) and specify the study is part of the COVID19-HGI initiative so that your imputation job will be given priority over others.

If your cohort already has imputed data, you may wish to consider reimputing with the latest reference panels using the TopMed or Michigan imputation servers, if your cohort’s population is represented in the reference panels, to ensure that the genotypes are as high-quality as possible.

Allele coding

If you are imputing your genotypes using your own reference panel, **please ensure that alleles for insertions and deletions are coded as per the** [**VCF 4.2 specification**](https://samtools.github.io/hts-specs/VCFv4.2.pdf#section.5): variants with reference or alternate alleles of “-”, “D” or “I” are ambiguous and are unlikely to be included in the meta-analysis.

X chromosome dosages

**If you impute the X chromosome, the coding at non-PAR regions must be 0/2 for males in order for the GWA software to calculate accurate association statistics.** The Michigan and TOPMED imputation servers output diploid genotypes for all samples in the non-PAR regions, but only if the input genotypes are correct. Therefore, please ensure that males are consistently coded “0” and “1” or “0/0” and “1/1” in the non-PAR regions of chromosome X. Both REGENIE and SAIGE consider only the unphased probabilities/dosages, so **the output from the imputation servers should be appropriate for use in these software, provided you convert the format correctly**, if not using VCF as input (as we recommend not to).

If you are not imputing your study’s genotypes through the imputation pipelines and are not sure if the non-PAR regions of the X chromosome are coded correctly in males, an explanation and tool are provided in the [Testing male X-chromosome coding in imputed VCFs](#_laz0uhnu4jtl) section of the [Appendix](#kykf9ynzisv3).

**If males are coded on the 0/1 scale in non-PAR regions, you will need to correct these genotypes manually.** We have provided a script to correct heterozygous pre-imputed hardcall genotypes in males in the non-PAR regions. Please see the [Correcting male non-PAR X-chromosome coding in pre-imputed VCFs](#_25h5um4jxhth) section in the Appendix.

Output formats

As we suggest the use of REGENIE to perform the GWA analyses, **we recommend that your imputed genotypes are stored in** [**PLINK2’s binary format .pgen/.psam/.pvar**](https://www.cog-genomics.org/plink/2.0/input#pgen), which supports genotype probabilities and is accepted by REGENIE, and that **each chromosome is stored in a separate fileset** for east. You will likely need to convert from VCF format, so it is important that you retain the dosage/probability information during the conversion. You may also convert to [Oxford .bgen/.sample](https://www.well.ox.ac.uk/~gav/bgen_format/spec/latest.html) (≥ v1.2) format. Instructions for converting VCF to PLINK2 .pgen/.psam/.pvar or to.bgen/.sample format are provided below.

If using SAIGE for GWA analysis, [Variant call format or VCF (.vcf(.gz))](https://samtools.github.io/hts-specs/VCFv4.2.pdf) are recommended, though SAIGE also accepts Oxford.bgen/.sample and [Savvy SAV format (.sav)](https://github.com/statgen/savvy/blob/master/sav_spec_v2.md). **If storing genotypes in VCF format, we recommend compressing files using bgzip (and not gzip) to agebe compatible with the tabix-indexing step and to save space** (if not already), as VCF is a plain-text format, unlike Bgen and SAV which are efficient binary formats. **If using Bgen or SAV formats please also make sure that probability/dosage information is retained** (i.e. that the “GP” field is present in the genotype format column).

Converting from VCF to PLINK2 format

For GWA analyses with REGENIE, we recommend using PLINK2’s .pgen/.psam/.pvar format. As the output of the imputation pipelines is in VCF format, you should use PLINK2 to convert your genotypes. First, if you have not done so already, please [download PLINK2](https://www.cog-genomics.org/plink/2.0/) ensuring that you have the correct version for your operating system and copy the pre-compiled binary to your working directory. To convert, you can use the following command:

./plink2 --vcf myfile.vcf.gz dosage=GP-force --make-pgen --out myfile

Repeat this process for each chromosome’s VCF file.

Converting from VCF to Bgen

The Michigan and TOPMED imputation servers output data in VCF format, but if you choose to use .bgen/.sample format genotypes as input to REGENIE or SAIGE, you will need to convert your VCFs to Bgen format. We recommend using [QCTOOL v2](https://www.well.ox.ac.uk/~gav/qctool_v2/) and converting with the following command:

./qctool -g myfile.vcf.gz -og myfile.bgen -vcf-genotype-field GP -ofiletype bgen\_v1.2 -bgen-bits 8

replacing ./qctool with the location of your QCTOOL v2 binary and myfile.vcf.gz and myfile.bgen with your input VCF and (desired) output Bgen file names.

Converting VCF to Bgen can be achieved using [PLINK2](https://www.cog-genomics.org/plink/2.0/)’s ability to handle non hardcall genotypes, **but PLINK lossily converts genotype probabilities to dosages and, as such, can generate less accurate genotypes than other methods**. We are also currently unable to ensure that PLINK correctly converts X chromosome genotypes in males and so **we would recommend avoiding using plink for the VCF to Bgen conversion**.

Index files for imputed genotypes

REGENIE can take advantage of index files to speed up GWA analyses when analysing .bgen format genotypes. Additionally, SAIGE requires fully indexed genotype files for quick access to binary or compressed genotype files. Therefore, **if you are using .bgen format imputed genotypes with REGENIE or any imputed format with SAIGE, you should create an index file for your imputed genotypes**. Instructions for each format are given below.

**Bgen (.bgen)**

* Download and compile the [BGEN software](https://enkre.net/cgi-bin/code/bgen/dir?ci=trunk) OR download the [pre-compiled binaries](https://www.well.ox.ac.uk/~gav/resources/) for [CentOS](https://www.well.ox.ac.uk/~gav/resources/bgen_v1.1.4-CentOS6.8-x86_64.tgz), [Ubuntu](https://www.well.ox.ac.uk/~gav/resources/bgen_v1.1.4-Ubuntu16.04-x86_64.tgz) or [OSX](https://www.well.ox.ac.uk/~gav/resources/bgen_v1.1.4-osx.tgz) (choose the appropriate one for your system). The pre-compiled binaries may not work if your system is missing their dependencies.
* Run the command bgenix -g myfile.bgen -index after replacing myfile.bgen with the name of your .bgen file.
* The created index file will be in the correct format for use with REGENIE or SAIGE and will be named as your .bgen file is named, but with an extra .bgi extension.

**VCF (.vcf.gz)**

* Ensure that your .vcf file is **bgzipped and not gzipped** (e.g. by using the command bgzip myfile.vcf).
* Download and install [samtools or bcftools](http://www.htslib.org/download/) (if not already installed).
* Run the command tabix -p vcf myfile.vcf.gz after replacing myfile.vcf.gz with the name of your bgzipped vcf.
* The created tabix index file will be in the correct format for use with SAIGE and will be named as your .vcf.gz file is named but with an extra .tbi extension.

**SAV (.sav)**

* Download the [latest release of Savvy](https://github.com/statgen/savvy/releases) and install from source or [install using cget or conda](https://github.com/statgen/savvy#installing). Alternatively, download the [Savvy shell install script](https://github.com/statgen/savvy/releases/download/v2.0.1/savvy-2.0.1-Linux-cli.sh), add executable permissions (e.g. chmod +x savvy-2.0.1-Linux-cli.sh) and run it (e.g. ./savvy-2.0.1-Linux-cli.sh).
* Run the sav executable to generate a Savvy index file:   
  ./sav index myfile.sav -o myfile.sav.s1r

and replace myfile.sav and the output .s1r filename with the name of your .sav file and desired index file name, ensuring that the index file keeps the .s1r extension.

* The created Savvy index file (ending in .s1r) will be in the required format for SAIGE.

## 

## Association analysis

*For any question regarding this section, please contact Samuel Jones (*[*samuel.jones@helsinki.fi*](mailto:samuel.jones@helsinki.fi) *- @Samuel Jones (Slack)), Vilma Lammi (*[*vilma.lammi@helsinki.fi*](mailto:vilma.lammi@helsinki.fi) *- @Vilma Lammi (Slack)) or Tomoko Nakanishi (*[*tomoko.nakanishi@mail.mcgill.ca*](mailto:tomoko.nakanishi@mail.mcgill.ca) *- @Tomoko Nakanishi (Slack))*

For all genetic studies, the following standard association model should be adopted if possible:

Phenotype ~ variant + age + age2 + sex + age\*sex + PCs + study\_specific\_covariates

GWA analyses will be run by each cohort and summary statistics uploaded to each study’s [Google Cloud bucket](#_pvg1q09rgmap), ready for the joint meta-analysis. We recommend using [REGENIE](https://rgcgithub.github.io/regenie/) for GWA analysis (see “[Appendix](#kykf9ynzisv3)” for installation instructions and example commands), as it accounts for relatedness and case-control unbalance. Use of the software package [SAIGE](https://github.com/weizhouUMICH/SAIGE) is also acceptable if you (or your analyst) is more familiar and has a working install. At present, we cannot accept GWA results generated by other software.

* Genotypes [should be imputed](#_2nxj9qcyth93) and can be either build hg19 or GRCh38/hg38.
* Phenotypes are described above in the “[Phenotype definitions](#kix.qd1gt19epgch)” section.
* Analysis should be run only in N.cases > 50. We recommend a minimum of 200 controls to ensure that we minimise the number of false-positive associations.
* Analyses should be run separately for each major ancestry group. Report results only for the ancestry groups with N.cases > 50.
* We suggest adjusting for the first 10 genetic principal components (PCs).
* *Study\_specific\_covariate* indicates covariates used to correct technical artifacts (e.g. batch number) and not risk factors or other comorbidities. Avoid adjusting for “heritable” covariates.
* MAF or INFO filtering are not necessary as both will be part of the uploaded summary statistics.
* Please see “[Appendix](#kykf9ynzisv3)” for more detailed [Analysis instructions for performing GWAS with REGENIE](#_8n481721h7qq) or [with SAIGE](#_nzwt8m62gcz8).

## Results Quality Checks

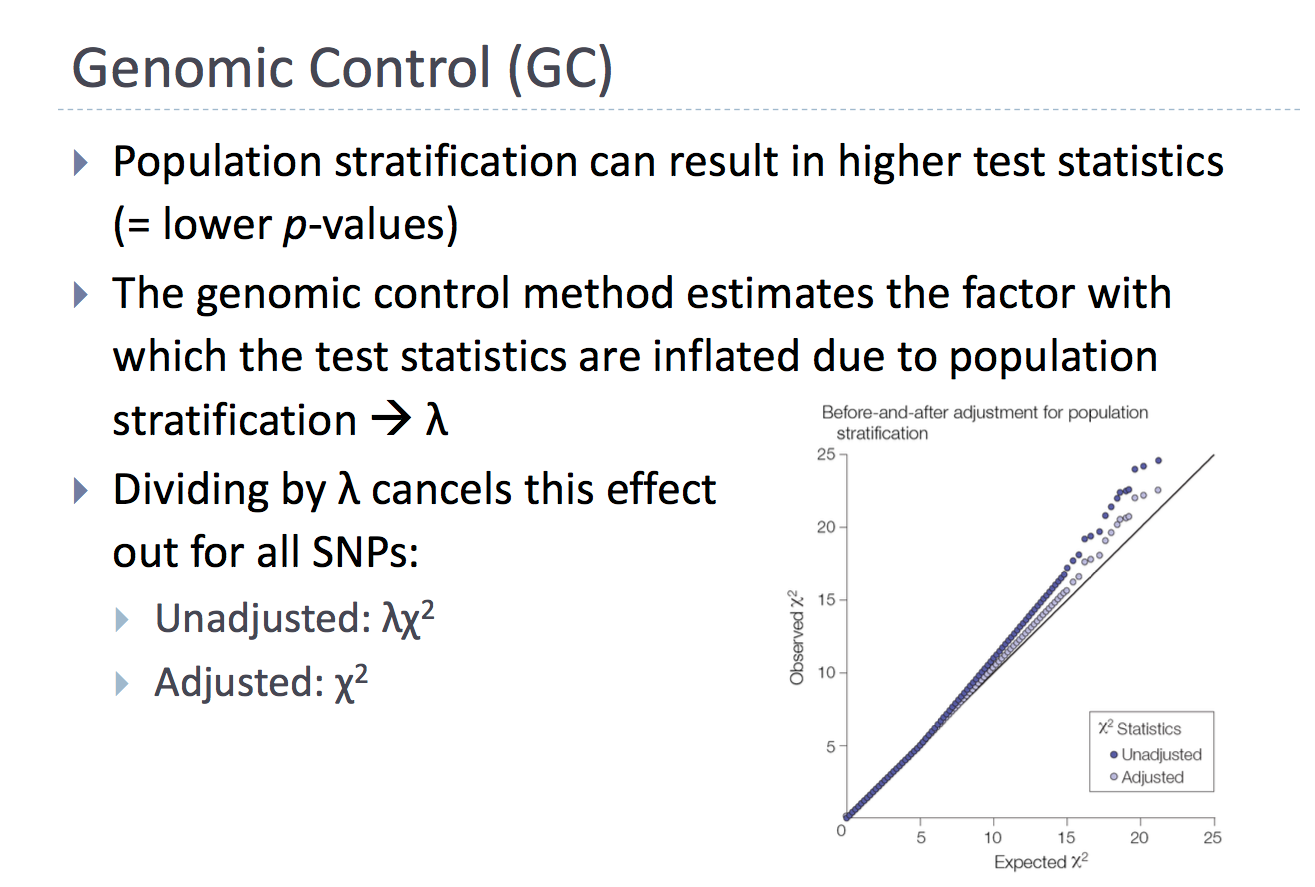
*(for any question regarding this section, please contact Samuel Jones (*[*samuel.jones@helsinki.fi*](mailto:samuel.jones@helsinki.fi) *- @Samuel Jones (Slack))*

*(Section originally written by Kumar Veerapen (*[*veerapen@broadinstitute.org*](mailto:veerapen@broadinstitute.org)*))*

To check the quality of your GWAS analysis, you can use [this R script](https://github.com/long-covid-hg/LongCovidTools/blob/main/QC_Checks/qq_man_longcovid.R) to generate a QQ plot and a Manhattan plot of the results. The script can take either REGENIE or SAIGE format summary statistics (see [Results format](#_zcv02hklx4ia)) as input, either plain-text or gzipped.

To run the script, simply download the file in the link (or copy and paste the code into a script called qq\_man\_longcovid.R) to your unix system and type

Rscript qq\_man\_longcovid.R --file=myresultsfile.txt(.gz)



where myresultsfile.txt(.gz) is the name of your GWAS results file. The script will then try to install the required packages (optparse, qqman, data.table, R.utils) if they’re missing and will create your plots. You can additionally specify the plotting output prefix in your Rscript command, using the --out myoutput flag. The script will automatically filter out low frequency (MAF≤0.1%) and low quality (INFO≤0.8) variants.

The information below has been provided to help you interpret these plots.

**Quantile-Quantile (QQ) plot**

The distribution of the analysed -log10(p-values) is linearly related to the expected values to some point. When there is a deviation from this, it would indicate that there are loci in your dataset which when higher than the expected normal distribution will hold significant (inflated QQ plot) or lower than the expected normal distribution (deflated QQ plot). Using the figure that was graciously adapted from the Analytic and Translational Genetics Unit Workshop 2020, we will describe the interpretation of a QQ plot.

When your QQ plot is highly inflated (unadjusted in Figure): the model may need further adjustment i.e. population stratification. Therefore, adjustment with the first 10 genetic principal components (PCs) should correct for stratification in your model. Another reason for inflation could be from the polygenic architecture in your model. As such, interpretation of the LD score regression intercept would shed light on this situation. Considering the LD score is directly estimating the polygenic effect, the LD score intercept should be approximately 1.0 (or not significant from 1.0). This would remain valid even in the presence of polygenicity. Any deviations from 1.0 would suggest uncontrolled population stratification. In the Figure, the unadjusted model is highly inflated with a genomic inflation (𝛌) of 3.2; upon adjustment, the inflation value reduced to 1.2. You can calculate the genomic inflation with the following formula (in R):

*median(qchisq(p\_value, df=1, lower.tail=FALSE)) / 0.456*

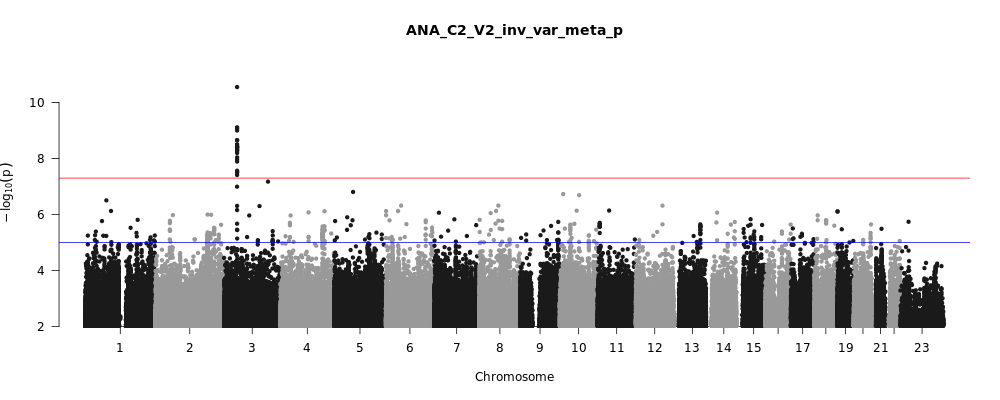
You would generally want a lambda of > 1.0 and less than ~ 1.5. This threshold would imply that you may have significantly associated loci to your trait of interest.

When your QQ plot is deflated (example at end of this section): the model may have some rare variants or that the variance of the tests are insufficient for the model to work well. For the latter, permutations may be needed to improve the quality of your tested model.

**Manhattan plot**

The easiest way to visualize the results from your analysis would be to create a Manhattan plot. An example of this would be the plot below where the x-axis are chromosomal positions and the y-axis is the -log10(p-value) from your association analysis. The highest associations will have the smallest p-values and therefore the highest -log10(p-value) in the plot.

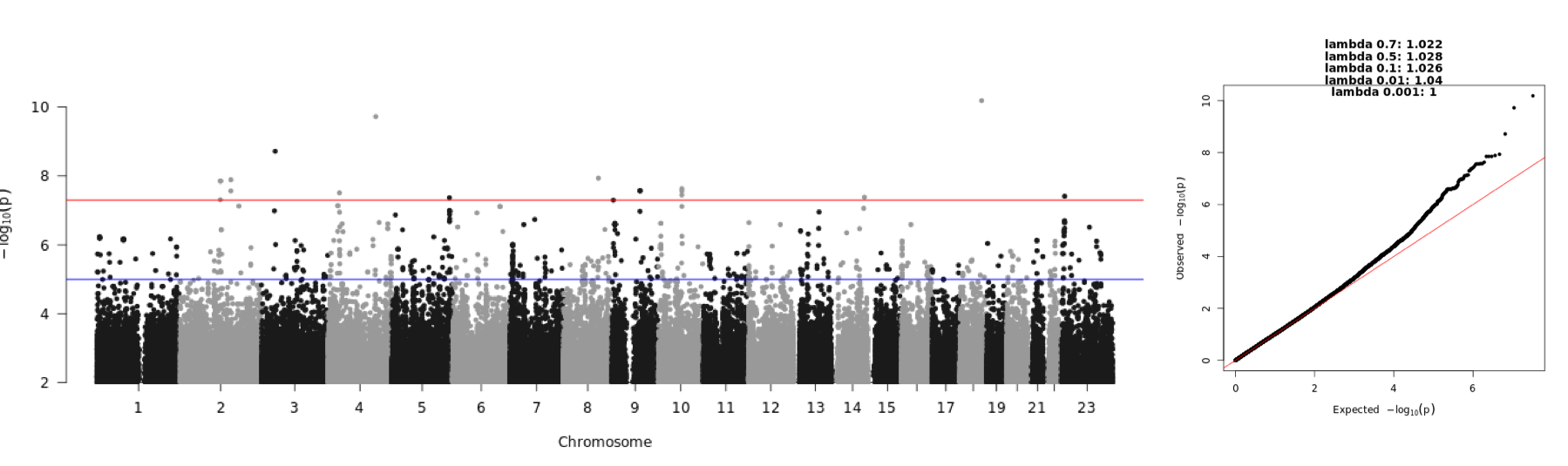
“Good” Manhattan Plot. A Manhattan plot that is considered “good” would have clear LD peaks with few sporadic points i.e. like the Manhattan skyline.



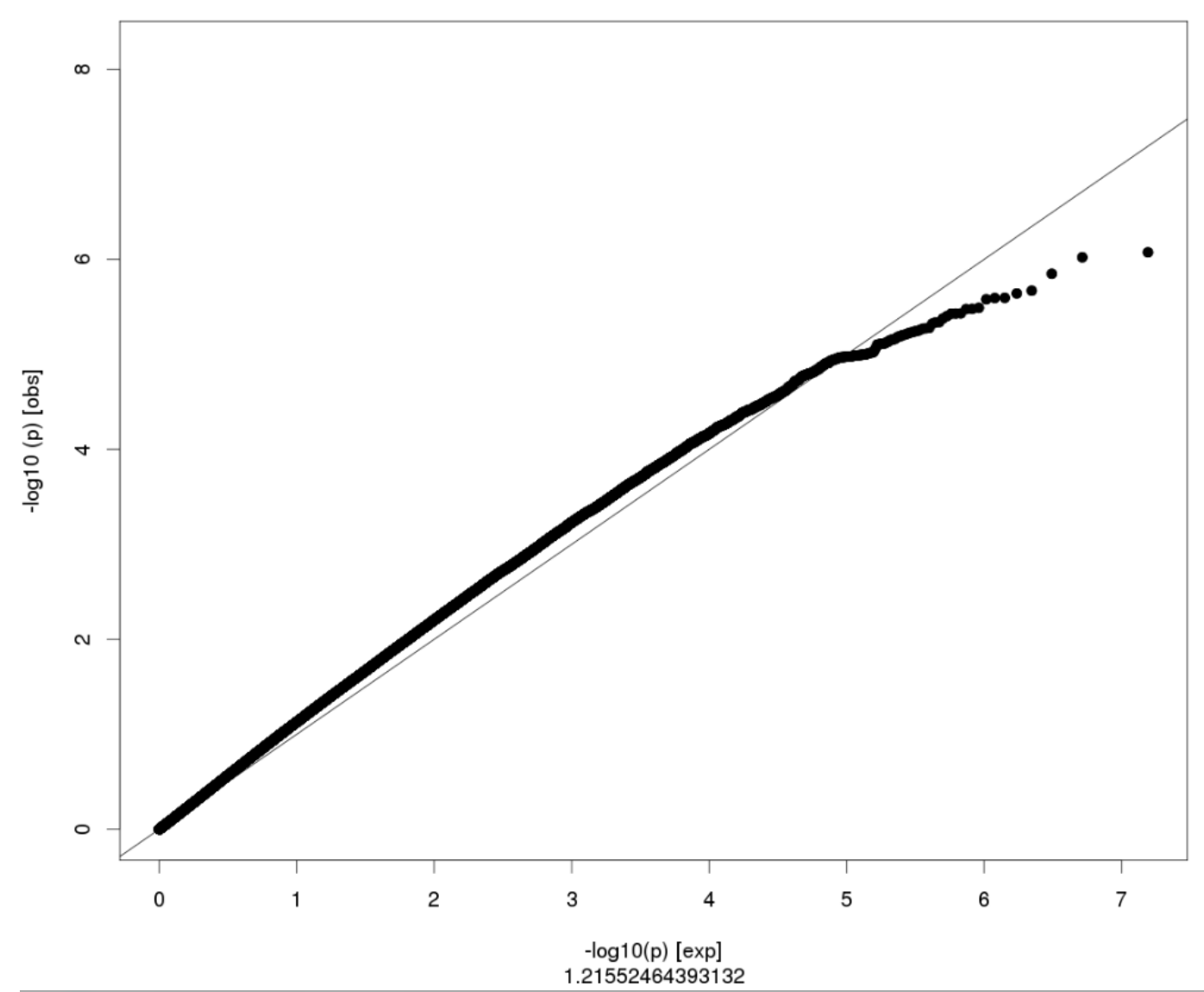
Other than observing significant loci from your association tests, Manhattan plots can also be used to determine if your statistical model needs further adjustments or a more strict quality control is needed. This could be determined by observing the “clarity” of the peaks.

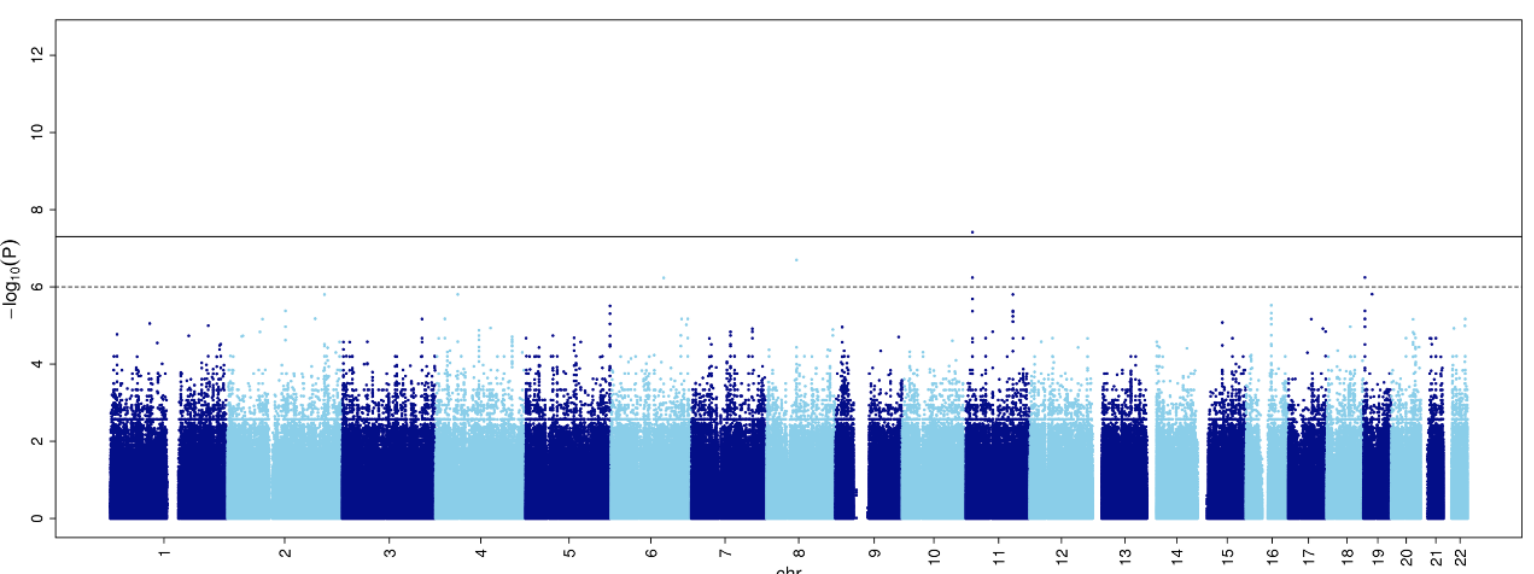
“Bad” Manhattan Plot. If the points are sporadic and many loci with single variants are seen (as in the plot below), this might indicate that a more strict quality control is needed. However, with small sample sizes, these sporadic points might simply be due to low allele frequencies variants. Inspect the frequency of those variants. **However**, If they are all < 1% (rare), then it is fine.

*Note*: Please share all results without filtering for allele frequencies and INFO score (as specified in the “[association analysis](#db7qlymf1abs)” section).

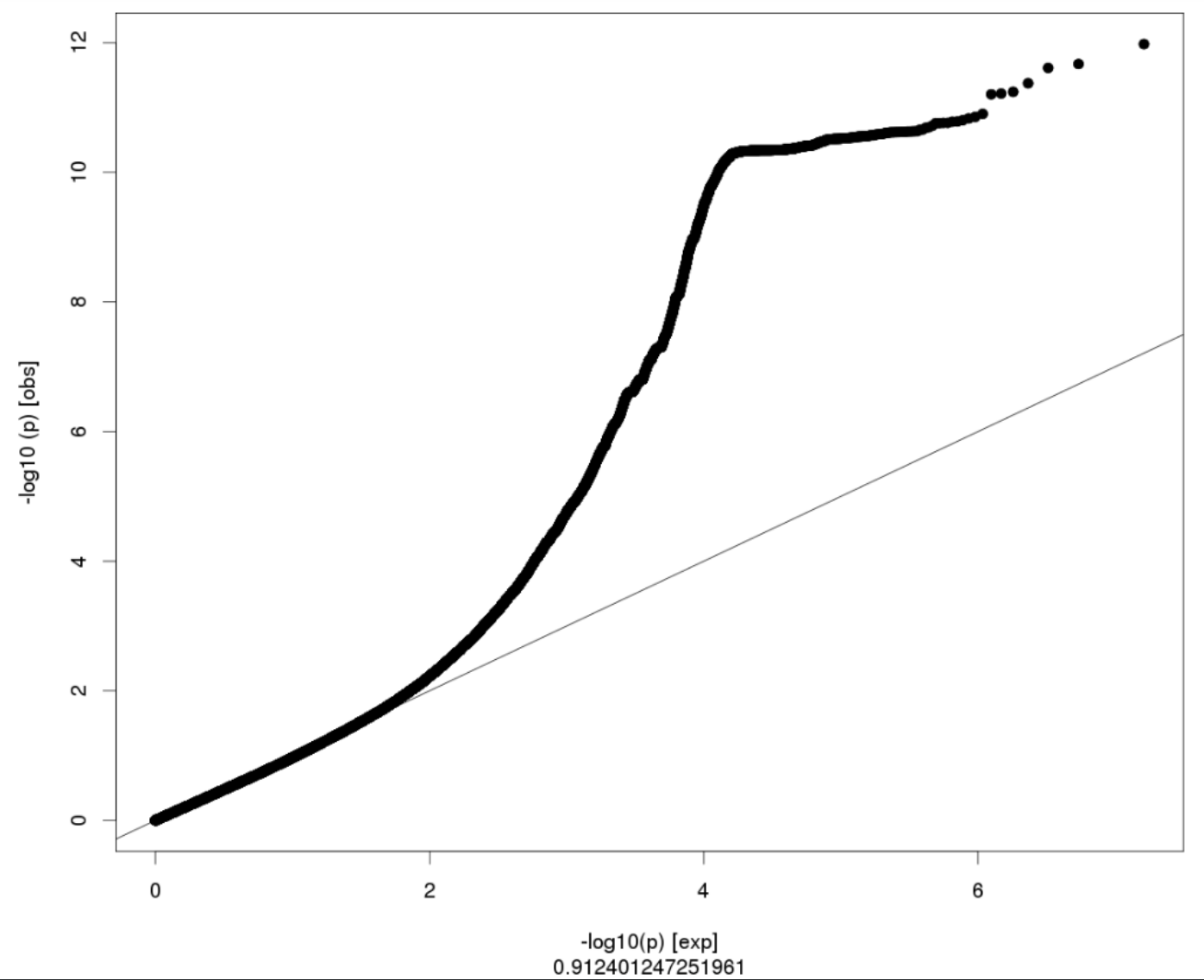
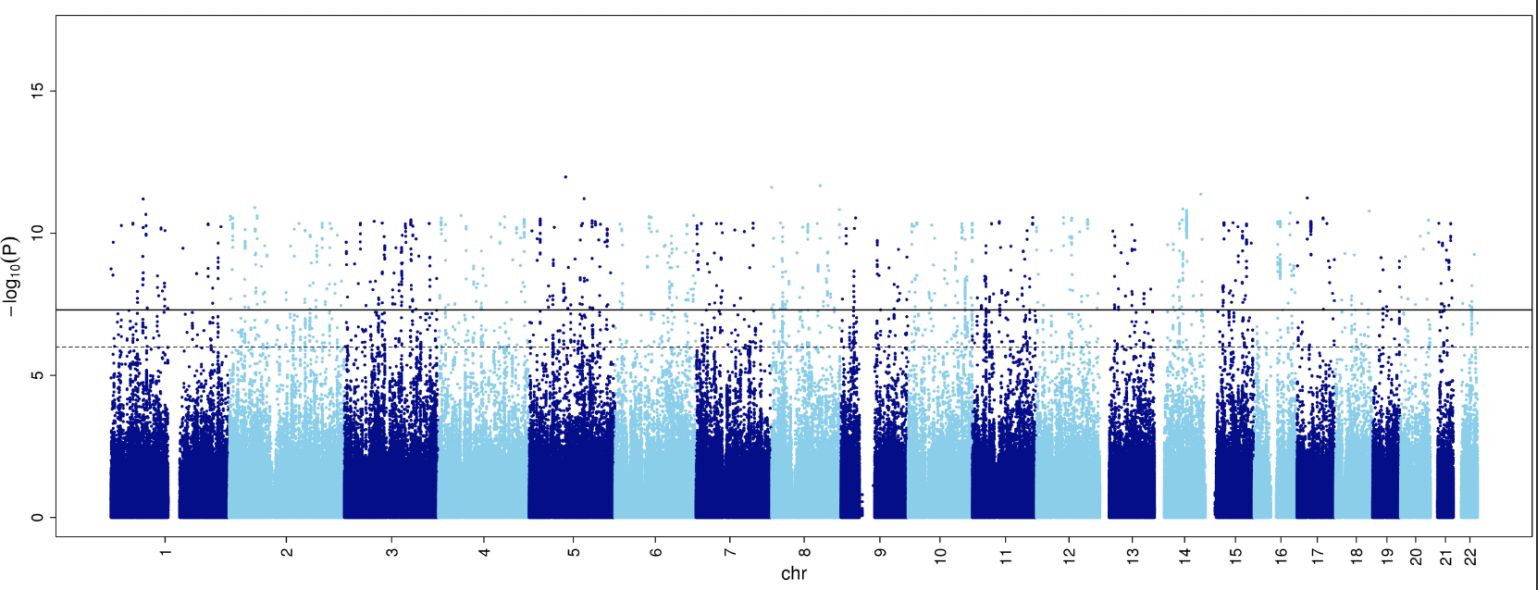


To show the relationship between QQ and Manhattan plots, we elaborate with further examples adapted from the Institute of Behavioral Science, University of Boulder, Colorado.

Deflated QQ plot that shows sporadic and poor associations throughout the genome:



Highly inflated QQ plot where everything is seemingly associated (but not) with the trait:



## Results format

*(for any question regarding this section, please contact Samuel Jones (*[*samuel.jones@helsinki.fi*](mailto:samuel.jones@helsinki.fi) *- @Samuel Jones (Slack))*

We welcome the association result files from both REGENIE and SAIGE. Please provide one file for each phenotype analysed. If, for a given phenotype, the analysis was run per chromosome, please combine all chromosomes into a single file before uploading to the Google Cloud bucket.

We recommend that you **retain all output fields/columns**. Please make sure that your files contain the **minimum set of fields** required for meta-analysis (including imputation info and sample size):

| **Column contents** | **REGENIE column name** | **SAIGE column name** |
| --- | --- | --- |
| Chromosome number | CHROM | CHR |
| Genome position | GENPOS | POS |
| Variant ID | ID | rsid |
| Non-effect allele | ALLELE0 | Allele1 |
| Effect allele | ALLELE1 | Allele2 |
| Effect allele frequency | A1FREQ | AF\_Allele2 |
| Imputation INFO score | INFO | imputationInfo |
| No. of samples with genotype | N | N |
| Beta | BETA | Beta |
| Standard Error (of Beta) | SE | SE |
| Association P-value (or log10P) | LOG10P | p.value |

Files should be **either space- or tab-delimited (not comma)** and please use these column names, keeping to one column naming convention (either REGENIE or SAIGE). Chromosome X can be represented by “X” or “23”. Chromosome numbers can be prefixed with “chr” or not (i.e. “chr1”, “1” and “01” are acceptable). The file should be sorted by chromosome and, within each chromosome, by base-pair position. Alternative contigs should not be included (only use autosomes and chromosome X) - contigs with any other name/label will be ignored in the meta-analysis.

### Results naming

**Analysis-specific information will be extracted from the filename**, so please keep to this naming scheme and be consistent in the way you name your files.

Upload the summary statistics file directly under your upload bucket (see [Results upload instructions](#_5qjedejw6g70)), without subfolders or zipped folders. Or if you prefer uploading several files at once, please name your combined zipped file as follows:

**[dataset]\_[freeze\_number]\_[YYYYMMDD].zip**

Please [**bgzip**](http://www.htslib.org/doc/bgzip.html) or gzip your **summary statistics**, and name the file using the following format:

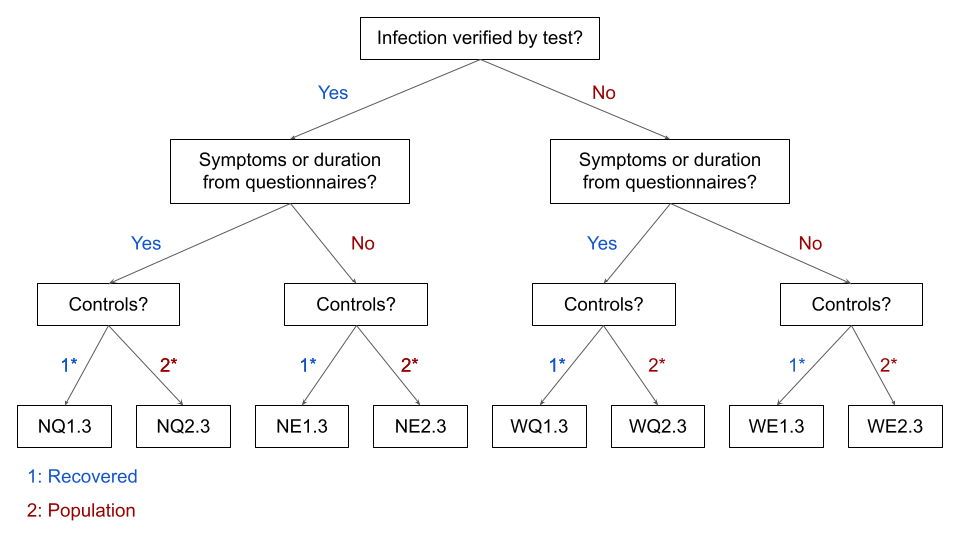
**[dataset]\_[surname]\_[phenotype]\_[freeze\_number]\_[age]\_[sex]\_[population]\_[n\_cases]\_[n\_controls]\_[gwas software]\_[YYYYMMDD].txt.gz**

(e.g., Finngen\_Jones\_WE2.3\_F1\_ALL\_ALL\_EUR\_200\_3500\_REGENIE\_20210915.txt.gz)

with these labels described below.

* [dataset] - study name or identifier, e.g. “UKBB”, “Finngen” or “ALSPAC”
* [surname] - surname (last name) of study’s PI or lead analyst, e.g. “Jones”
* [phenotype] - phenotype label, e.g. “WE2.3” (see reminder below on the phenotype nomenclature.
* [freeze\_number] - number of the data freeze these results are intended for, prefixed with “F” (e.g. (“F1”, “F2”, “F3”, …). The current freeze is ‘F4’.
* [age] - age group indicator for age-stratified analyses (e.g. “ALL”, “LE\_60”, “GT\_60”). Currently, “ALL” is the only group.
* [sex] - sex indicator for sex-stratified analyses (e.g. “ALL”, “FEMALE” or “MALE”). Currently, “ALL” is the only group (unless your cohort contains only a single sex).
* [population] - population label, as inferred from proximity to the the [1000 Genomes Phase 3 populations](https://www.internationalgenome.org/data-portal/population), e.g. by PCA analysis or self-identification:
  + African: AFR
  + Admixed American: AMR
  + European: EUR
  + East Asian: EAS
  + South Asian: SAS
  + (Other: **please indicate an appropriate label and let us know!**)
* [n\_cases] - final number of cases that were included in this phenotype’s GWA analysis, after all exclusions due to missing data have been made
* [n\_controls] - final number of controls that were included in this phenotype’s GWA analysis, after all exclusions due to missing data have been made
* [gwas\_software] - currently either “REGENIE” or “SAIGE”
* [YYYYMMDD] - analysis date in the YYYYMMDD format

***TIP: How to choose your phenotype label?***



See the [Phenotype Definitions file](https://docs.google.com/document/d/1-o5VX83WiWZfBuVi1jkneH3Bz7p9p5Eg) for details.

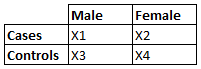
## Sample summary information

*(for any question regarding this section, please contact Samuel Jones (*[*samuel.jones@helsinki.fi*](mailto:samuel.jones@helsinki.fi) *- @Samuel Jones (Slack))*

As each study will have different recruitment strategies, data collection methods and demographics, we request that each study provide some basic information on the samples included and the diagnoses used.

Data required from all studies

From all studies, **for each distinct ethnicity** analysed separately, we require the following:

* Histograms of age distribution (in those included in your GWAS) for
  + all samples
  + females only
  + males only
* Cross-tabulation of sex with case-control status for each phenotype that you are submitting results for, e.g.

* NEW: Genetic principal component (PC) plots (PC1-PC2 and PC3-PC4) showing how your cohort is distributed amongst the 1000 Genomes populations. We provide [scripts and an explanation](#_uvx80nxd7vjl) on how to do this.

Further data required from questionnaire-based studies

* A document listing the questions (original language AND best possible English translation) used to determine:
  + COVID-19 diagnosis status
  + Long COVID status

Further data required from EHR-based studies

* A spreadsheet tabulation of the **frequency** of diagnosis codes used to determine:
  + COVID-19 diagnosis status
  + Long COVID status

Preparing and uploading these files

Please combine all requested files into a .zip archive with the following naming format

**[dataset]\_[surname]\_[freeze\_number].zip**

with these labels described below.

* [dataset] - study name or identifier, e.g. “UKBB”, “Finngen” or “ALSPAC”
* [surname] - surname (last name) of study’s PI or lead analyst, e.g. “Jones”
* [freeze\_number] - number of the data freeze these results are intended for, prefixed with “F” (e.g. (“F1”, “F2”, “F3”, …)

Upload this zip file along with your GWAS results to your bucket, as described in the [Results upload instructions](#_5qjedejw6g70) section.

## Results upload instructions

*(for any question regarding this section, please contact* [*sbryant@broadinstitute.org*](mailto:sbryant@broadinstitute.org) *- @samcbryant (Slack))*

### Access to Google Cloud

To upload your data or access data in the long-covid-hg buckets, you will first need a Google Account (Check if you have a Google account: [enter your email address](https://www.google.com/accounts/ForgotPasswd)). Instructions on the Google site: [Create a Google Account](https://support.google.com/accounts/answer/27441?hl=en&ref_topic=3382296) You can also use your existing email address and link it to a Google Account ([Google: Add an alternate email address](https://support.google.com/accounts/answer/176347?co=GENIE.Platform%3DDesktop&hl=en)).

Once you have a Google Account,check that you have the correct email connected to your Google account listed in the [COVID\_HG\_LongCOVID\_StudyInfoForm](https://docs.google.com/spreadsheets/d/1XfarBojPob6J9fwJpmGODn3cnX8evDrUeZmT-RI5suk/edit?usp=sharing) (PI/Lead or Contact/Analyst email column). We’ll set up a bucket and give you access.

### Bucket for uploading data

There will be an upload bucket created for each group. Use your upload bucket to upload your data. QC checks will be done with uploaded data and the data are then transferred to the analysis bucket.

The buckets are created based on the information each study has provided in the [COVID\_HG\_LongCOVID\_StudyInfoForm](https://docs.google.com/spreadsheets/d/1XfarBojPob6J9fwJpmGODn3cnX8evDrUeZmT-RI5suk/edit?usp=sharing). If a bucket has not been assigned to you and you are ready to provide data, please contact us, and we’ll sort this out with you.

For use with the browser:

<https://console.cloud.google.com/storage/browser/long-covid-hg-><cohort name>

To upload data, go to your upload bucket (need to be logged in with the given Google account) at the link provided to you, and use the Upload files or Upload folder buttons. Upload times vary depending on the network at your institution. If you prefer to use a command line utility, use [gsutil](https://cloud.google.com/storage/docs/gsutil_install). Instructions for gsutil use can be found [here](https://cloud.google.com/storage/docs/gsutil). Example command:

**gsutil cp Finngen\_Jones\_E2.3\_F1\_ALL\_ALL\_EUR\_200\_3500\_REGENIE\_20210915.txt.gz gs://long-covid-hg-your\_study/**

Analysis-specific information will be extracted from the filename, so please keep to the naming described in “[Results naming](#_h8vqucuo9xe5)”. Upload each summary statistics file directly under your upload bucket, without subfolders or zipped folders.

For each file that you upload, add the corresponding descriptive data as described in “[Sample summary information](#_9jyc3fycronl)”.

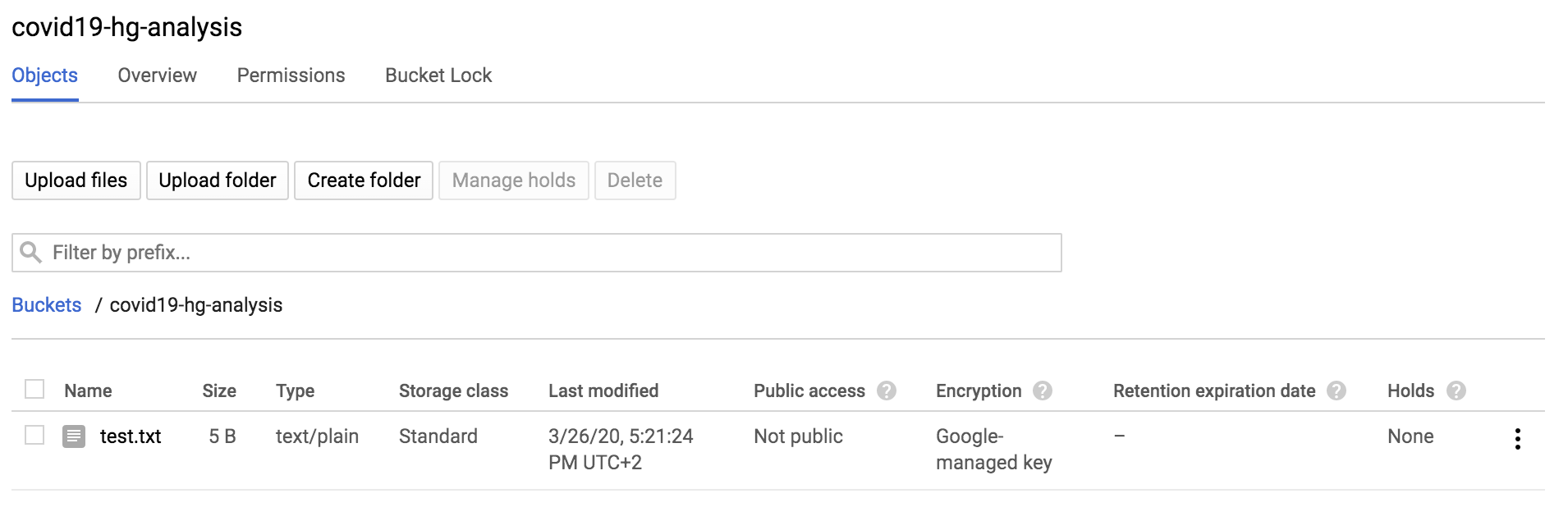
Please also check that you have correct information regarding the chromosome build, X chromosome coding and imputation in the [COVID\_HG\_LongCOVID\_StudyInfoForm (](https://docs.google.com/spreadsheets/d/1XfarBojPob6J9fwJpmGODn3cnX8evDrUeZmT-RI5suk/edit?usp=sharing)end of sheet “Form”.

### Bucket for downloading data

*[NOT YET AVAILABLE, will be created when we have data to share]*

long-covid-hg-analysis - use this bucket to read or download data provided by participants, as well as analysis results.

To download data from the long-covid-hg-analysis bucket, check the checkboxes next to the files or folders, click the 3 vertical periods (ellipses) at the far right and select “Download”.



## Appendix

*(for any question regarding this section, please contact* [*samuel.jones@helsinki.fi*](mailto:samuel.jones@helsinki.fi) *- @samuel jones(Slack))*

### Analysis instructions for performing GWAS with REGENIE

**REGENIE** is a C++ program created to efficiently perform genome-wide single-variant association analyses and gene-burden tests. The software corrects for population structure and cryptic relatedness while accounting for over-inflated statistics for rare variants and in highly unbalanced case-control ratios. **We recommend its use, as we have found it easier to install and quicker to run than SAIGE**, with no perceivable loss of accuracy.

Installation

There are multiple options for installing REGENIE, depending on your experience level or access rights and availability of dependencies on the machine or cluster that you will be performing the GWAS with. Installation instructions are provided [here](https://rgcgithub.github.io/regenie/install/) and we recommend, if possible, that you compile the code from source by cloning the latest build from the [REGENIE Git repository](https://github.com/rgcgithub/regenie), or by creating a [Docker image](https://github.com/rgcgithub/regenie/wiki/Using-docker) or [conda environment](https://anaconda.org/bioconda/regenie). This will ensure that your installation is up-to-date with the latest bug fixes and that it makes best use of your computing architecture.

[Pre-compiled binaries](https://github.com/rgcgithub/regenie/releases) are also provided by the REGENIE team. The advantage of these is that no installation is required, just download the relevant binary, unpack it and run the executable. However, if your machine is missing any of the REGENIE dependencies, you may have to install REGENIE from source or contact your system administrator to either install REGENIE or the missing dependencies. Pre-compiled binaries are not optimised to take advantage of your system’s architecture and so analyses may be slower and less efficient than using an installation compiled from source.

Input files

The REGENIE documentation has detailed information on the [input file format](https://rgcgithub.github.io/regenie/options/#input). The required files are:

* **Genotype data for building the whole genome model (step 1)**

To build the association model that is used to correct for relatedness and population structure, we need to input a set of high-quality, common genotypes from the same samples that will be analysed in the GWAS (step 2). We recommend using your study’s pre-imputed array genotypes in [PLINK’s .bed/.bim/.fam format](https://www.cog-genomics.org/plink/2.0/input#bed) with the following variant inclusion criteria:

* + Minor allele frequency (MAF) ≥ 0.01 (i.e. 1%)
  + Variant missingness ≤ 0.01 (1%)
  + Hardy-Weinberg Equilibrium P-value ≥ 1x10-6
  + Bi-allelic SNPs (no indels)
  + Outside of regions with [extended LD](https://www.cell.com/ajhg/fulltext/S0002-9297(08)00353-4) - regions are available [here](https://github.com/long-covid-hg/LongCovidTools/blob/main/Resources/longrange_LD_regions_hg37_GRCh37.txt) (b37) and [here](https://github.com/long-covid-hg/LongCovidTools/blob/main/Resources/longrange_LD_regions_hg38_GRCh38.txt) (b38)
  + Autosomal (chromosome 1-22) variants only
* **Genotype data for genetic association analysis (step 2)**

In step 2, REGENIE will perform the association analysis on your imputed genotypes, using the model built in step 1. The software accepts genotypes in three formats:

* + [PLINK .pgen/.pvar/.psam](https://www.cog-genomics.org/plink/2.0/input#pgen)
  + [Oxford .bgen/.sample](https://www.well.ox.ac.uk/~gav/bgen_format/spec/latest.html)
  + [PLINK .bed/.bim/.fam](https://www.cog-genomics.org/plink/2.0/input#bed)

**We advise that your imputed data be input as .pgen/.pvar/.psam format OR (if necessary) .bgen/.sample format**, as these two formats retain the uncertainties in genotype probabilities from the imputation process, unlike the .bed/.bim/.fam format. For ease of parallelisation, **it is also necessary that imputed genotypes are stored in one fileset per chromosome**, with each chromosome analysed individually (in parallel) in step 2.

* **Phenotype file**

This file contains a list of the sample identifiers (IDs) and their corresponding phenotype. The phenotype file format for REGENIE is [described here](https://rgcgithub.github.io/regenie/options/#phenotype-file-format). As we are analysing case-control phenotypes, you should use the coding **0=control, 1=case and NA=missing**; the software will not understand any other coding. The phenotype file must be **space- or tab-delimited** and **columns 1 and 2 must have a header of FID and IID, respectively**. These columns contain the family ID and the individual ID - if your study participants only have one identifier, these two columns should both contain the same ID. **Column 3 onwards should contain your phenotypes**, with the phenotype name or label as the header of their respective column.

* **Covariate file**

This **space- or tab-delimited file** contains a list of the sample identifiers (IDs) and their corresponding covariates (e.g. age, sex, PCs). As with the phenotype file, the **first two columns are family ID and individual ID, respectively, with column headers FID and IID. Column 3 onwards should contain all the covariates needed to perform your analysis**. See the [Association analysis](#db7qlymf1abs) section for information on which covariates to include. Each covariate in the additive model requires a column of its own, including those with interaction terms. For example age2 and age\*sex should each have their own columns and should be listed as covariates in the REGENIE command.

Note that the phenotypes and covariates can be included in a single phenotype-covariate file, with FID and IID as the first two columns, then phenotypes and covariates in subsequent columns. **It is also possible for these files to only have one ID column; column 1 should be IID with columns 2 onwards containing phenotypes and/or covariates and you must add the** --iid-only **flag to the REGENIE command.**

Step 1: Building the whole genome model

Once the necessary input files are prepared, you should be ready to run step 1 of REGENIE. An example command is given below. You will need to replace ./regenie with the location of your REGENIE executable.

./regenie \

--step 1 \

--bed GENOTYPES\_FOR\_MODEL \

--phenoFile PHENOTYPE\_FILE \

--phenoCol PHENOTYPE\_NAME \

--covarFile COVARIATE\_FILE \

--covarColList CONT\_COVARIATE\_NAMES \

--catCovarList CAT\_COVARIATE\_NAMES \

--bt \

--bsize 1000 \

--loocv \

--threads NUM\_THREADS \

--out REGENIE\_OUT\_STEP1

You should modify the command and replace the following arguments:

* GENOTYPES\_FOR\_MODEL = file prefix of plink-binary format (.bed/.bim/.fam) filtered array-based genotypes used to build the whole-genome model
* PHENOTYPE\_FILE = file path of space- or tab-delimited phenotype file
* PHENOTYPE\_NAME = column name(s) of phenotype column(s) in phenotype file (if multiple, separated by comma; e.g. PHENO1,PHENO2,...)
* COVARIATE\_FILE = file path of space- or tab-delimited covariate file
* CONT\_COVARIATE\_NAMES = list of column names for continuous covariate columns in covariate file, separated by comma (e.g. age,age2,PC1,PC2,PC3,...)
* CAT\_COVARIATE\_NAMES = list of column names for binary or categorical covariate columns in covariate file, separated by comma (e.g. sex,batch,...)
* NUM\_THREADS = number of CPU threads that REGENIE should utilise
* REGENIE\_OUT\_STEP1 = prefix for output files from step 1

A full list of options (some of which are specific to either step 1 or step 2) are available [here](https://rgcgithub.github.io/regenie/options/#options). Additional flags that we recommend considering for step 1 are:

* --lowmem - useful if performing the analysis on a system with limited memory. Writes intermediate files to the hard disk so increases compute time.
* --strict - exclude individuals who have missing phenotype or covariate data. **We recommend adding this flag.**
* --gz - compress output to gzip format. **We recommend adding this flag to save space, though it may only work for those who compile from source.**

Step 1 will output REGENIE\_OUT\_STEP1.log, REGENIE\_OUT\_STEP1\_pred.list and files REGENIE\_OUT\_STEP1\_{1:N}.loco (where N is the number of phenotypes given to the --phenoCol flag). These files are needed for step 2 of REGENIE.

**Note: REGENIE step 1 may take up to a few days to complete depending on your system specs, number of covariates included and, of course, sample size.**

Step 2: Running the association test

Once the necessary input files are prepared and step 1 has successfully run, you should be ready to run the GWA analysis using step 2 of REGENIE. An example command is given below. You will need to replace ./regenie with the location of your REGENIE executable.

./regenie \

--step 2 \

--pgen PGEN\_FOR\_GWAS \

--chr CHR \

--phenoFile PHENOTYPE\_FILE \

--phenoCol PHENOTYPE\_NAME \

--covarFile COVARIATE\_FILE \

--covarColList COVARIATE\_NAMES \

--bt \

--spa \

--pred REGENIE\_OUT\_STEP1\_pred.list \

--bsize 1000 \

--out REGENIE\_OUT\_STEP2

**If using .bgen/.sample format instead of PLINK2’s .pgen/.pvar/.psam instead of**, replace the --pgen PGEN\_FOR\_GWAS flag with --bgen BGEN\_FOR\_GWAS and --sample SAMPLE\_FILE\_FOR\_GWAS..

As for step 1, you should modify the command and replace the following arguments:

* PGEN\_FOR\_GWAS = fileset prefix of PLINK .pgen/.pvar/.psam files. All three files must have the same prefix. Needed if inputting .pgen/.pvar/.psam format imputed genotypes.
* BGEN\_FOR\_GWAS = file path (including extension) of .bgen format imputed genotypes to be analysed. Only required if inputting .bgen/.sample format imputed genotypes.
* SAMPLE\_FILE\_FOR\_GWAS = file path (including extension) of .sample file corresponding to .bgen file. Only required if inputting .bgen/.sample format imputed genotypes.
* CHR = chromosome number (or X) to be analysed
* PHENOTYPE\_FILE = file path of space- or tab-delimited phenotype file
* PHENOTYPE\_NAME = column name(s) of phenotype column(s) in phenotype file (if multiple, separated by comma; e.g. PHENO1,PHENO2,...)
* COVARIATE\_FILE = file path of space- or tab-delimited covariate file
* COVARIATE\_NAMES = list of column names for covariate columns in covariate file, separated by comma (e.g. sex,age,age2,PC1,PC2,PC3,...)
* REGENIE\_OUT\_STEP1 = prefix for output files from step 1
* REGENIE\_OUT\_STEP2 = prefix for output files for step 2 (can be the same as in step 1).

A full list of options (some of which are specific to either step 1 or step 2) are available [here](https://rgcgithub.github.io/regenie/options/#options). Additional flags that we recommend considering for step 2 are:

* --strict - exclude individuals who have missing phenotype or covariate data. **We recommend adding this flag.**
* --gz - compress output to gzip format. **We recommend adding this flag to save space, though it may only work for those who compile REGENIE from source.**
* --write-samples - write to file a list of samples included in the analysis. **We recommend doing this, as it helps to calculate numbers of cases and controls.**
* --threads NUM\_THREADS - number of CPU threads that REGENIE should utilise. This may speed up association tests, though we are unsure how much of an effect parallelisation has on step 2 of REGENIE.

It is important that the --spa flag is set to keep the results comparable between REGENIE and SAIGE. SAIGE, by default, uses a Saddle Point Approximation to more accurately estimate effect sizes and standard errors of variants with large P-values and so we must ensure that REGENIE approximates these in the same way.

We advise running step 1 in a multithreaded environment and setting NUM\_THREADS to a minimum of 8, to make best use of the parallelisation features of REGENIE. Of course, the number of threads available to you depends on your computing system.

Full instructions for running single-variant association analyses in REGENIE can be found at: <https://rgcgithub.github.io/regenie/options/#getting-started>

### Analysis instructions for performing GWAS with SAIGE

**SAIGE** is an R library designed to perform genome-wide single-variant and gene-burden association tests. This approach provides robust statistics for scenarios such as rare variants and highly skewed case:control ratios, while accounting for sample relatedness.

Installation

Because SAIGE is an R library and not stand-alone software, there are no pre-compiled binaries available and so SAIGE must be installed. Instructions for installing SAIGE can be found [here](https://github.com/weizhouUMICH/SAIGE/wiki/Genetic-association-tests-using-SAIGE#installing-saige). As with REGENIE, there are multiple options.

**We recommend installing SAIGE through the R packages “devtools” or “remotes”**. These can be installed by running R (≥ v3.6.1) and typing:

install.packages("devtools") or install.packages("remotes")

SAIGE is then installed using the commands (replace devtools with remotes if needed):

devtools::install\_github("leeshawn/MetaSKAT")

devtools::install\_github("weizhouUMICH/SAIGE")

SAIGE’s current minimum requirements are an R installation of v3.6.1, gcc v5.4.0, cmake v3.14.1,and cget. If your dependencies do not meet these requirements but your system has [conda installed](https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html), then you can attempt installation using the [provided SAIGE environment file](https://raw.githubusercontent.com/weizhouUMICH/SAIGE/master/conda_env/environment-RSAIGE.yml) by following the [SAIGE conda installation instructions](https://raw.githubusercontent.com/weizhouUMICH/SAIGE/master/conda_env/environment-RSAIGE.yml). The conda environment will need to be activated before running SAIGE analyses.

Alternatively, SAIGE can be installed using [Docker](https://docs.docker.com/get-started/overview/), if it is installed on your system. To do so, simply type:

docker pull wzhou88/saige:0.44.6.5

and run the docker image in interactive mode by typing (for example):

docker run -it -v $PWD:/tmp -w /tmp --env-file .envs --name SAIGE IMAGEID

where IMAGEID should be SAIGE’s docker image ID obtained from the docker ps command and .envs contains all the environment variables you wish to pass to the docker container (docker does not, by default, have access to your system’s environment variables).

Input files

The SAIGE documentation has information on [input file format](https://github.com/weizhouUMICH/SAIGE/wiki/Genetic-association-tests-using-SAIGE#input-files-1). The required files are:

* **Genotype data for building the null logistic mixed model (step 1)**

To build the null model that is used to correct for relatedness and population structure, we need to input a set of high-quality, common genotypes from the same samples that will be analysed in the GWAS (step 2). We recommend using your study’s pre-imputed array genotypes in [PLINK’s .bed/.bim/.fam format](https://www.cog-genomics.org/plink/2.0/input#bed) with the following variant inclusion criteria:

* + Minor allele frequency (MAF) ≥ 0.01 (i.e. 1%)
  + Variant missingness ≤ 0.01 (1%)
  + Hardy-Weinberg Equilibrium P-value ≥ 1x10-6
  + Bi-allelic SNPs (no indels)
  + Outside of regions with [extended LD](https://www.cell.com/ajhg/fulltext/S0002-9297(08)00353-4) - regions are available [here](https://github.com/long-covid-hg/LongCovidTools/blob/main/Resources/longrange_LD_regions_hg37_GRCh37.txt) (b37) and [here](https://github.com/long-covid-hg/LongCovidTools/blob/main/Resources/longrange_LD_regions_hg38_GRCh38.txt) (b38)
  + Autosomal (chromosome 1-22) variants only
* **Genotype data for genetic association analysis (step 2)**

In step 2, SAIGE will perform the association analysis on your imputed genotypes, using the model built in step 1. The software accepts genotypes in three formats:

* + [Variant call format or VCF (.vcf(.gz))](https://samtools.github.io/hts-specs/VCFv4.2.pdf)
  + [Oxford .bgen/.sample](https://www.well.ox.ac.uk/~gav/bgen_format/spec/latest.html)
  + [Savvy SAV format (.sav)](https://github.com/statgen/savvy)

**We advise that your imputed data be input in .vcf(.gz) format OR (if necessary) .bgen/.sample format**, as these two formats are widely compatible with other software. For ease of parallelisation, **it is also advised that imputed genotypes are stored in one fileset per chromosome**, with each chromosome analysed individually (in parallel) in step 2.

* **Sample list (if step 2 genotype data in .bgen format)**

SAIGE requires a list of individuals that corresponds exactly to the genotypes within the .bgen file, but does not accept the standard .sample file. This sample list corresponds to the first column of the .sample file with no headers (first two lines removed). You can create this file using the following code:

awk 'NR>2{print $1}' example.sample > example\_samples.txt

and by substituting in the names of your files.

* **Sample list (if step 2 genotypes in .vcf(.gz) format AND SAIGE version is <0.38)**

Earlier versions of SAIGE require a sample list to be included with a VCF file. You can run the following command to generated this list:

zcat myfile.vcf.gz | awk '$0~/^#CHR/{for(i=10;i<=NF;i++){print $i};exit}' > myfile\_samples.txt

If your vcf is not gzipped (doesn’t have the .gz extension), replace the zcat with cat in the above command. The resulting file myfile\_samples.txt should then be in the correct format for use with SAIGE.

* **Combined phenotype and covariate file**

This file contains a list of the sample identifiers (IDs) and their corresponding phenotype(s) and covariates. The file format for SAIGE is [described here](https://github.com/weizhouUMICH/SAIGE/wiki/Genetic-association-tests-using-SAIGE#input-files). As we are analysing case-control phenotypes, you should use the coding **0=control, 1=case and NA=missing**; the software will not understand any other coding. The phenotype/covariate file must be **space- or tab-delimited** and **column 1 must have a header of IID**.This column contains the individual ID. **Column 2 (onwards) should contain your phenotype(s)**, with the phenotype name or label as the header of their respective column and then the **covariates (e.g. age, sex, PCs) in subsequent columns** with appropriate column headers.

**Note: Categorical covariates (e.g. batch) with more than 2 categories must be recorded as multiple binary covariates.** For example, a single batch variable with N categories would need to be converted to N binary batch variables coded 1 if they are in the relevant batch and 0 otherwise.

Step 1: Building the null-model

After preparing the necessary input files, you should be ready to run step 1 of SAIGE. An example command is given below.

Rscript step1\_fitNULLGLMM.R \

--plinkFile=GENOTYPES\_FOR\_MODEL \

--phenoFile=PHENOTYPE\_COVARIATE\_FILE \

--phenoCol=PHENOTYPE\_NAME \

--covarColList=COVARIATE\_NAMES \

--sampleIDColinphenoFile=IID \

--traitType=binary \

--outputPrefix=SAIGE\_OUT\_STEP1 \

--LOCO=TRUE \

--nThreads=NUM\_THREADS

You should modify the command and replace the following arguments:

* GENOTYPES\_FOR\_MODEL = file prefix of plink-binary format (.bed/.bim/.fam) filtered array-based genotypes used to build the null model
* PHENOTYPE\_COVARIATE\_FILE = file path of space- or tab-delimited combined phenotype and covariate file
* PHENOTYPE\_NAME = column name of the phenotype column in the combined phenotype-covariate file
* COVARIATE\_NAMES = list of column names for covariate columns in covariate file, separated by comma (e.g. sex,age,age2,PC1,PC2,PC3,...)
* NUM\_THREADS = number of CPU threads that SAIGE should utilise
* SAIGE\_OUT\_STEP1 = prefix for output files from step 1

There are no additional options or flags that we recommend using. However, you can see a full list by typing Rscript step1\_fitNULLGLMM.R --help.

SAIGE will output SAIGE\_OUT\_STEP1.rda and SAIGE\_OUT\_STEP1.varianceRatio.txt after successful completion of step 1. These files are needed for step 2 of SAIGE.

**Note: SAIGE step 1 may take up to a week to complete depending on your system specs, number of covariates included and, of course, sample size.**

Step 2: Running the association test

Once the necessary input files are prepared and step 1 has successfully run, you should be ready to run the GWA analysis using step 2 of SAIGE. An example command is given below.

Rscript step2\_SPAtests.R \

--vcfFile=VCF\_FOR\_GWAS \

--vcfFileIndex=VCF\_INDEX\_FOR\_GWAS \

--vcfField=DS \

--chrom=CHR \

--GMMATmodelFile=SAIGE\_OUT\_STEP1.rda \

--varianceRatioFile=SAIGE\_OUT\_STEP1.varianceRatio.txt \

--SAIGEOutputFile=SAIGE\_OUT\_STEP2.bgen.txt \

--numLinesOutput=1000 \

--LOCO=TRUE \

--IsOutputNinCaseCtrl=TRUE \

--IsOutputHetHomCountsinCaseCtrl=TRUE \

--IsOutputAFinCaseCtrl=TRUE

**NOTE: If you are using SAIGE version 0.37 or earlier with VCF input, you will also need the flag --sampleFile=SAMPLE\_FILE\_FOR\_GWAS.** Please see the “Input files” section above for how to create this sample list.

**If using Bgen (.bgen) or SAV (.sav) formats instead of the recommended VCF (.vcf(.gz)) format**, replace the four VCF-specific options:

* --vcfFile=VCF\_FOR\_GWAS
* --vcfFileIndex=VCF\_INDEX\_FOR\_GWAS
* --vcfField=DS and
* --chrom=CHR

With the three .bgen-specific options:

* --bgenFile=BGEN\_FOR\_GWAS
* --bgenFileIndex=BGEN\_INDEX\_FOR\_GWAS and
* --sampleFile=SAMPLE\_FILE\_FOR\_GWAS

or the four SAV-specific options:

* --savFile=SAV\_FOR\_GWAS
* --savFileIndex=SAV\_INDEX\_FOR\_GWAS
* --vcfField=DS and
* --chrom=CHR.

As for step 1, you should modify the command and replace the following arguments:

* VCF\_FOR\_GWAS = file path (including extension) of .vcf.gz format imputed genotypes to be analysed. Only required if inputting .vcf(.gz) format imputed genotypes.
* VCF\_INDEX\_FOR\_GWAS = file path (including extension) of .vcf.gz.tbi format tabix index file corresponding to the .vcf.gz input. Only required if inputting .vcf.gz format genotypes.
* BGEN\_FOR\_GWAS = file path (including extension) of .bgen format imputed genotypes to be analysed. Only required if inputting .bgen format imputed genotypes.
* BGEN\_INDEX\_FOR\_GWAS = file path (including extension) of .bgen.bgi index file corresponding to the .bgen file. Only required if inputting .bgen format imputed genotypes.
* SAMPLE\_FILE\_FOR\_GWAS = file path (including extension) of text file containing nothing but a list of sample IDs in the same order as the input .vcf.gz or .bgen file (see “Input files” above). Only required if inputting vcf.gz genotypes and using SAIGE version 0.37 or earlier or if inputting .bgen format imputed genotypes with any version of SAIGE,
* SAV\_FOR\_GWAS = file path (including extension) of .sav format imputed genotypes to be analysed. Only required if inputting .sav format imputed genotypes.
* SAV\_INDEX\_FOR\_GWAS = file path (including extension) of .sav.s1r Savvy index file corresponding to the .sav file. Only required if inputting .sav format imputed genotypes.
* CHR = chromosome code of input .vcf.gz or .sav file, exactly as specific in the file. For instance, **if the chromosome code is “chr10”, this option should be specified as --chrom=chr10 and NOT --chrom=10**; the latter will cause the program to fail.
* SAIGE\_OUT\_STEP1 = prefix for output files from step 1.
* SAIGE\_OUT\_STEP2 = prefix for output files for step 2. If you are analysing each chromosome individually, **we recommend including the chromosome number in this prefix** so that each chromosome’s analysis does not overwrite the previous one!

One optional flag that you may consider running using is --nThreads=NUM\_THREADS, if you feel that the association testing will benefit from parallelisation. A full list of further options for SAIGE step 2 can be seen by running the command Rscript step2\_SPAtests.R --help.

Full instructions for running single-variant association analyses in SAIGE can be found at: <https://github.com/weizhouUMICH/SAIGE/wiki/Genetic-association-tests-using-SAIGE#single-variant-association-tests>ngGenerating PC plots for your cohort

This document provides instructions to create projected PCs of GWAS participants on the same PCA space using the pre-computed loadings and reference allele frequencies. The goal is to plot all studies participating in the Long COVID HGI together.

Step-by-step instructions

**1. Prerequisites**

* [PLINK 2 software](https://www.cog-genomics.org/plink/2.0/)
* Pre-computed PCA loadings and reference allele frequencies downloaded from [here](https://github.com/long-covid-hg/pca_projection/blob/master/docs/prerequisites.md#download-the-pre-computed-pca-loadings-and-reference-allele-frequencies).
* Imputed dosages, phenotype, and covariate files from your study used in Long COVID GWAS for data freeze 4 (or most recent freeze if <4)
* Analysis scripts downloaded from [our GitHub repository](https://github.com/long-covid-hg/pca_projection).

For detailed prerequisites, please refer to [this document](https://github.com/long-covid-hg/pca_projection/blob/master/docs/prerequisites.md).

**2. Project PCs**

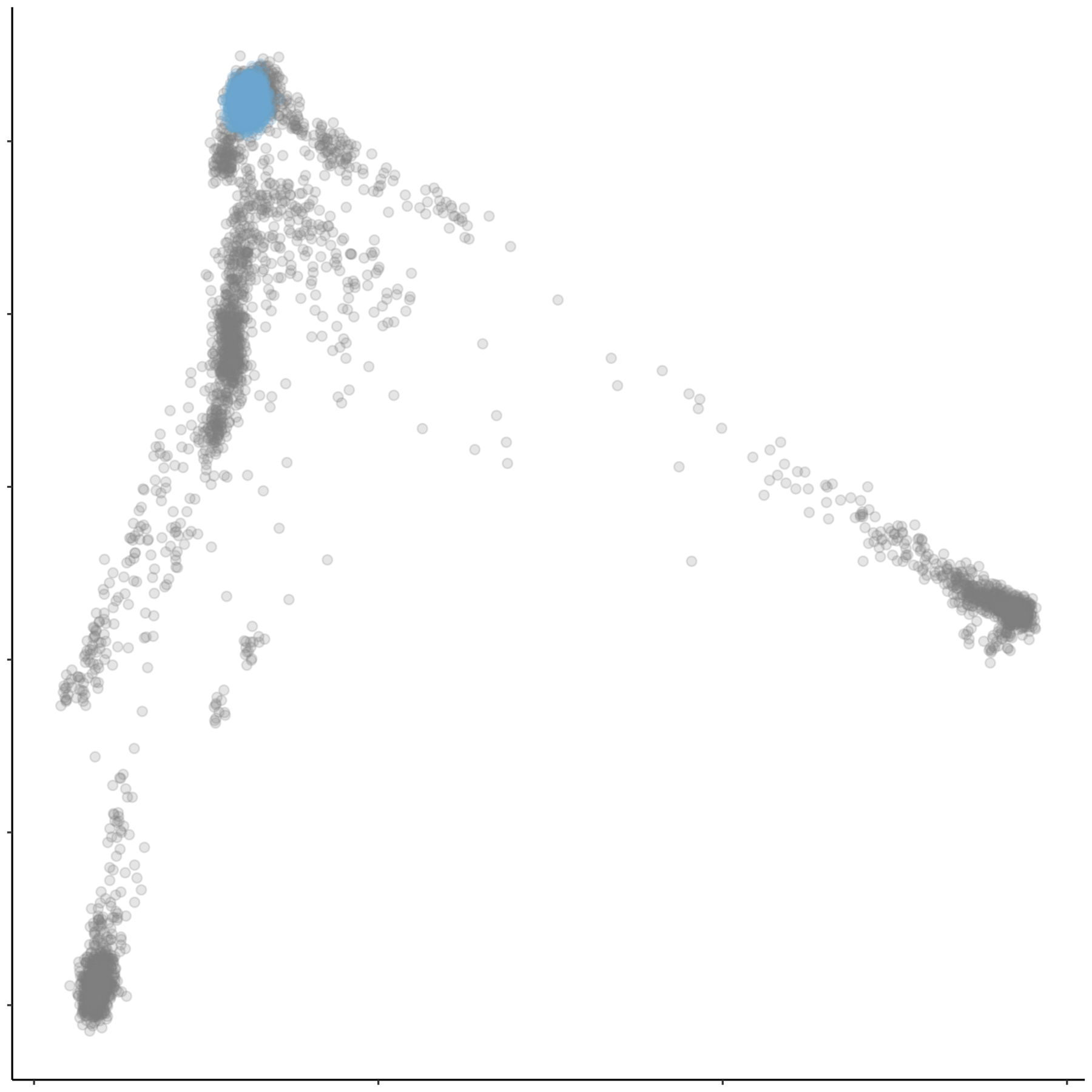
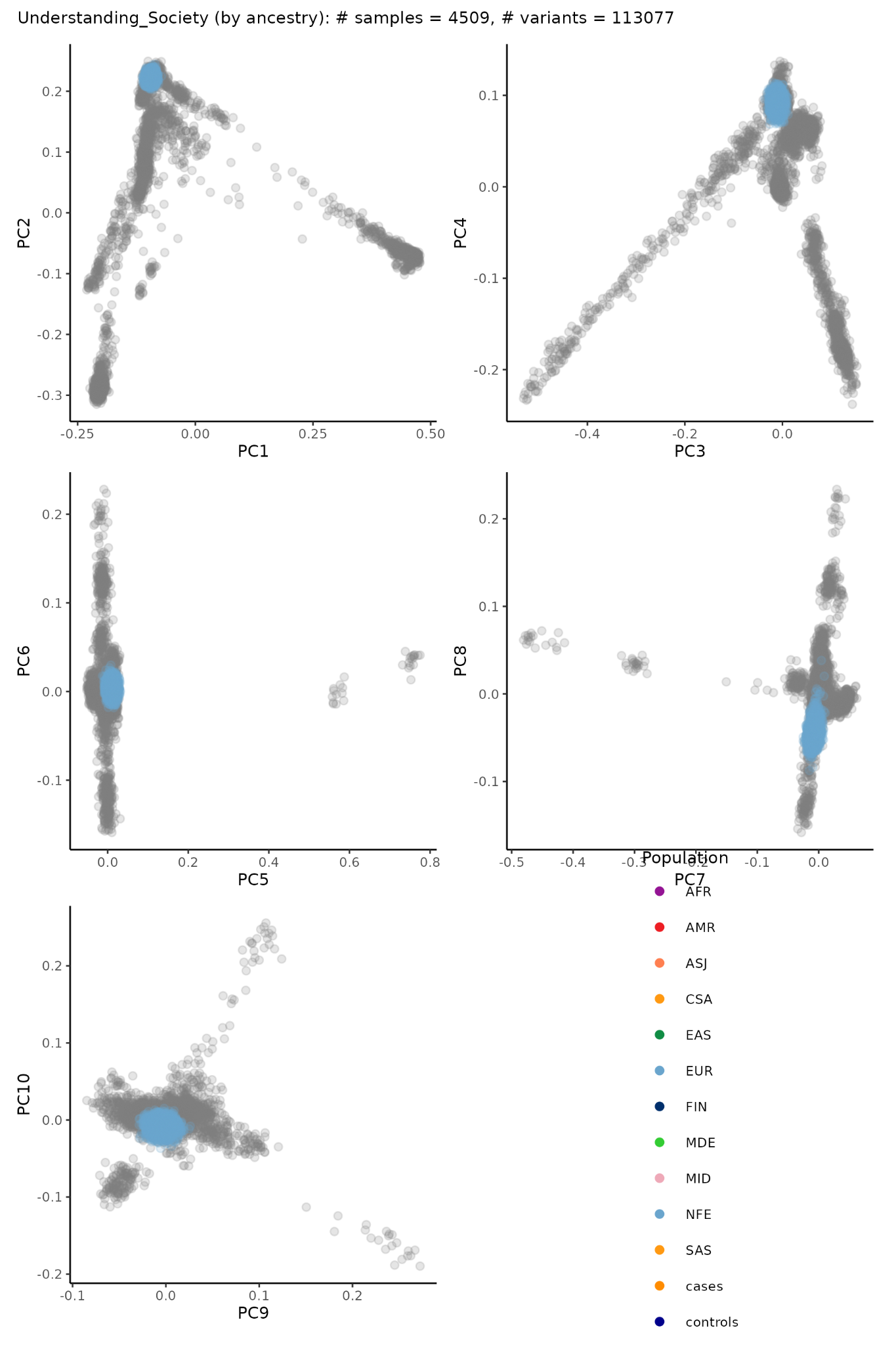
We use [PLINK 2](https://www.cog-genomics.org/plink/2.0/) to project each individual on the same PCA space using the Human Genome Diversity Project (HGDP) and the 1000 Genomes Project samples as reference. We provide an automated script available [here](https://github.com/long-covid-hg/pca_projection). Please refer to [this document](https://github.com/long-covid-hg/pca_projection/blob/master/docs/project_pc.md) for instructions.

**3. Plot projected PCs**

Once the projected PCs are computed, please use [this script](https://github.com/long-covid-hg/pca_projection/blob/master/plot_projected_pc.R) to plot all the projected PCs in the same PCA space. Documentation for the script is available [here](https://github.com/long-covid-hg/pca_projection/blob/master/docs/plot_projected_pc.md). Outputs will be:

* PNG images for top 10 PCs, all combined and transparent images for each
* A text file with per-sample projected PC values without including cohort-specific individual IDs

**Example: Understanding Society (EUR)**



Top 10 projected PCs Transparent

combined image for PC1-2

**4. Upload result files**

Please upload all the png images and a text file with projected PC values to the same bucket you uploaded GWAS results. For uploading, please refer to the [Results upload instructions](#bfcagfj12ojf) section. If you have concerns over sharing per-sample PC projection values, please feel free contact the Long COVID HGI leadership Vilma Lammi ([vilma.lammi@helsinki.fi](mailto:vilma.lammi@helsinki.fi)) or Hanna Ollila ([hanna.m.ollila@helsinki.fi](mailto:hanna.m.ollila@helsinki.fi)).

If you have any technical questions, please contact Samuel Jones ([samuel.jones@helsinki.fi](mailto:samuel.jones@helsinki.fi)), Tomoko Nakanishi ([tomoko.nakanishi@mail.mcgill.ca](mailto:tomoko.nakanishi@mail.mcgill.ca)), or Vilma Lammi ([vilma.lammi@helsinki.fi](mailto:vilma.lammi@helsinki.fi)).

### 

### Testing male X-chromosome coding in imputed VCFs

If you are unsure how males are coded in the non-PAR regions, we provide a software tool to test three common (MAF>10% in all populations) non-PAR X-chromosome variants.

Downloading the tool

The scripts can either be downloaded through the browser, or by using git (if installed).

**Browser**:

* Go to <https://github.com/long-covid-hg/LongCovidTools/tree/main/Xchr_Coding_Test>
* Download the three files and copy them to your desired directory on a Unix machine.
* Set the permissions for the shell script: chmod +x check\_X\_chr\_vcf\_ploidy.sh

**Git**:

* Go to your desired work folder on your Unix system (warning: ensure that no subfolder named “LongCovidTools” already exists in this location)
* Clone the repo:

git clone https://github.com/long-covid-hg/LongCovidTools.git

* Copy out the required files: cp LongCovidTools/Xchr\_Coding\_Test/\* ./
* Delete the repo’s folder: rm -rf LongCovidTools
* Set the permissions for the shell script: chmod +x check\_X\_chr\_vcf\_ploidy.sh

Using the tool

The tool uses R and requires specific packages to be installed. Please see the information in the “Requirements” section below.

You will need to provide the tool with the X chromosome .vcf(.gz) file you want to test and the [plink-format .fam file](https://www.cog-genomics.org/plink/1.9/formats#fam) from the pre-imputed genotypes of the same samples that are in your .vcf(.gz) file. If you don’t already have a .fam file, you can build one by creating a headerless six-column space-delimited file with sample ID in both columns 1 and 2, zeros in columns 3 and 4, sex (“1” for male, “2” for female, without the quotes) in column 5 and “-9” (without the quotes) in column 6. Please make sure that the sample IDs in column 1 match the format of those in the last VCF header line (starting “#CHROM”), otherwise this tool will fail to work.

To run the tool, simply type:

./check\_X\_chr\_vcf\_ploidy.sh myfile.vcf.gz myfile.fam

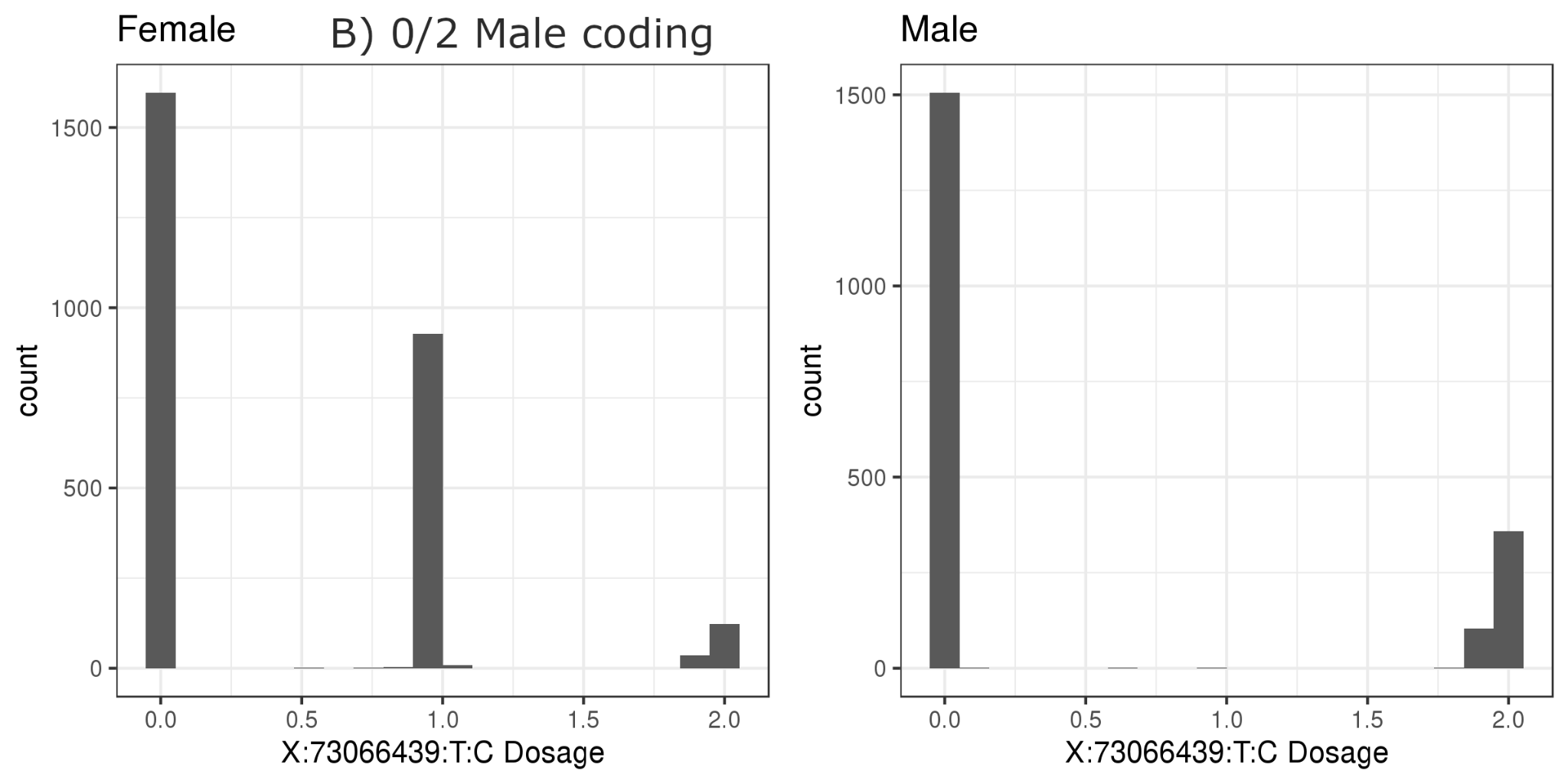
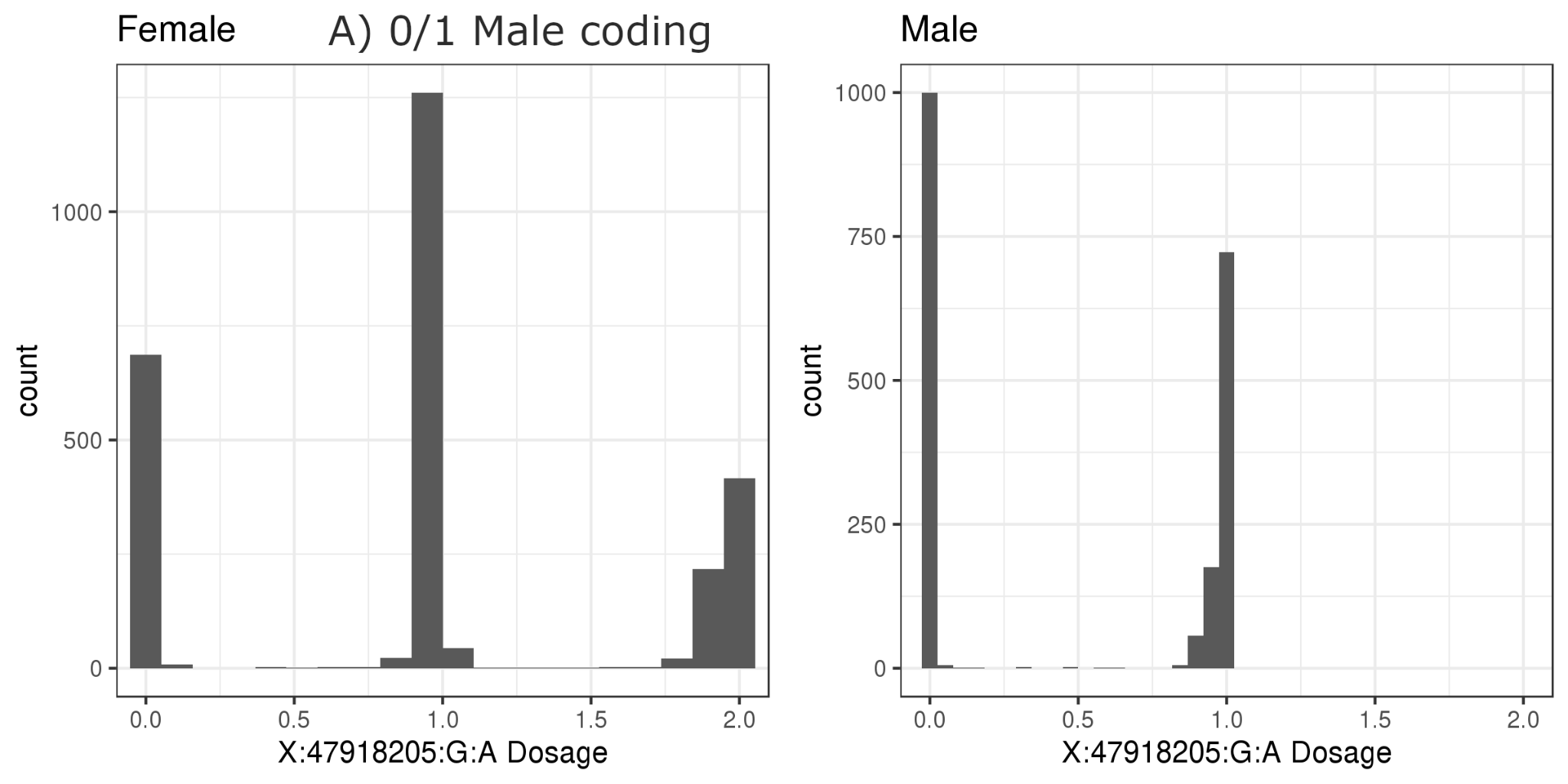
Assuming that the pre-selected variants are contained within your X chromosome VCF, three .png files should be output. These images should then be inspected (see “Interpreting output” section below) to see if the male coding is correct in these variants. The tool will output error messages if it fails and will provide warnings if it doesn’t find all three pre-selected variants.

Interpreting the output

The tool will first check for a “DS” (dosage) field and if it can’t find it, will use the “GP” (genotype probabilities) field from the VCF. The output will then be one image per variant of either dosages (files named “dosage\_distribution\*png”) or genotype probabilities (files named “genotype\_probabilities\*png”).

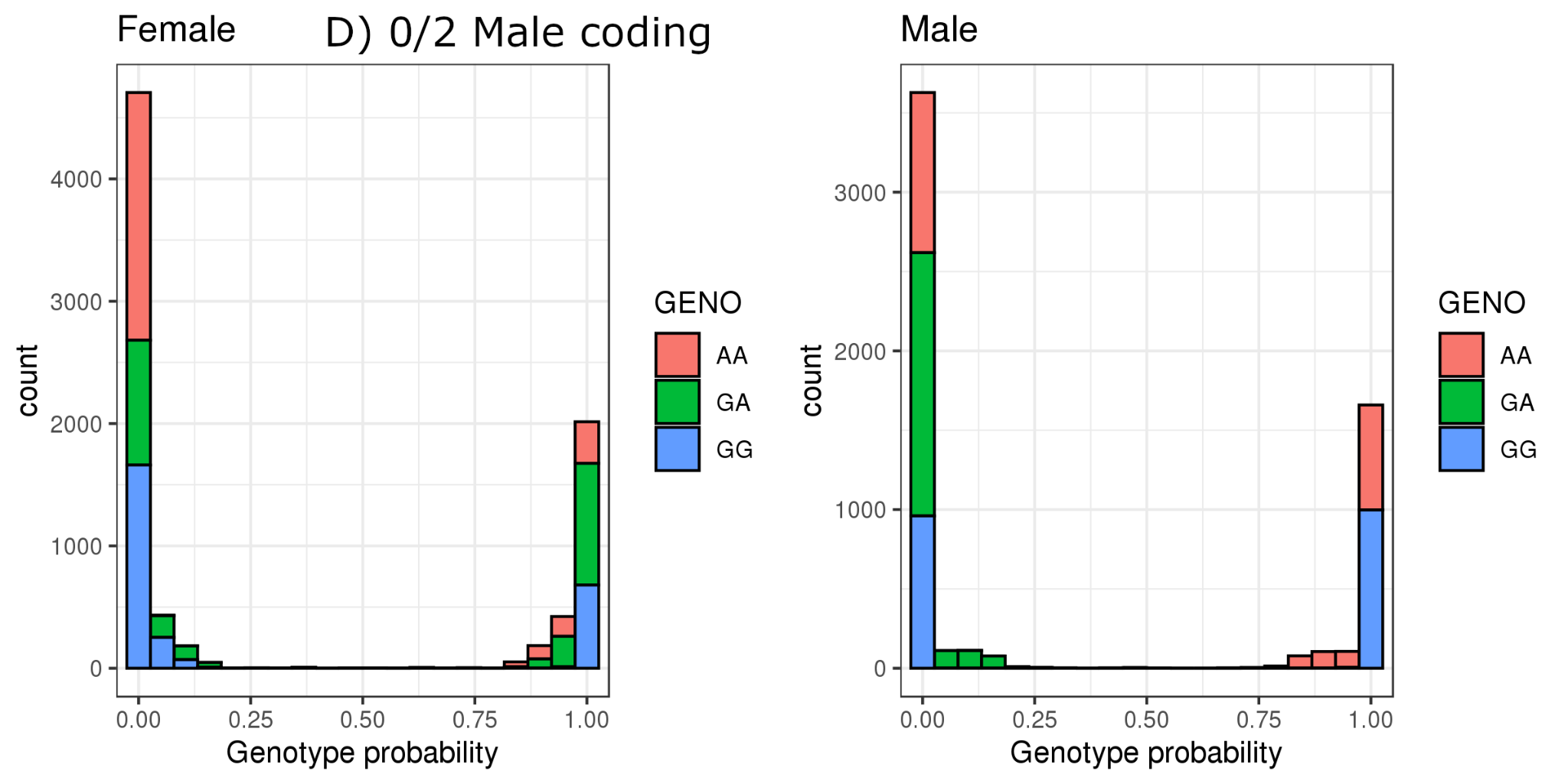
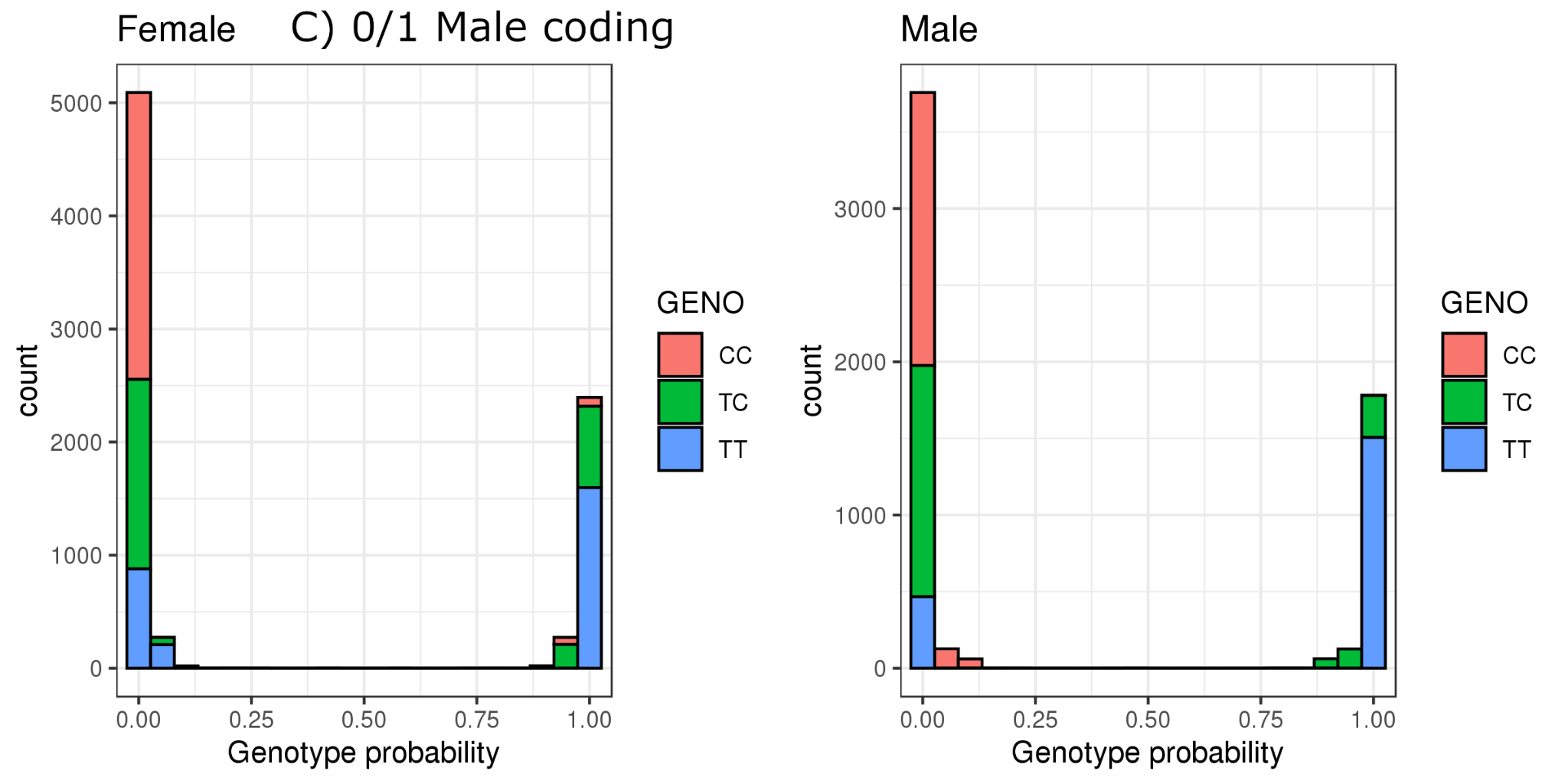
**Dosage histograms**

In these histograms, it should be easy to determine if the coding is 0/1 or 0/2 in males. Looking at the right-hand image (title “Male”), the coding will be 0/1 if there is a peak at 1.0 on the x-axis, but no peak at 2.0 (see figure A). Likewise, the coding will be 0/2 in males if there is a peak at 2.0, but no peak at 1.0 (see figure B).



**Genotype probability histograms**

In these histograms, you can determine the coding by looking at the Male plot (right-hand side) at the bar near genotype probability of 1.0. If the bar contains RefRef and RefAlt genotypes (see figure C), the coding is 0/1 and is incorrect. However, if the bar only contains RefRef and AltAlt genotypes (see figure D), the coding is 0/2 and is correct.



These checks should be made for a minimum of three high INFO (>0.8) common (MAF>10%) non-PAR variants to ensure that they all agree.

Choosing your own variants

If any of the pre-selected variants are not present in your cohort’s imputed X chromosome genotype VCF, or if you wish to test further variants, you can edit the check\_variants.txt file and add your own. The file requires RSID (if present) in column 1, hg19/GRCh37 CHR:POS:REF:ALT in column 2 and GRCh38 CHR:POS:REF:ALT in column 3. Please ensure that any variants you add are not in the [pseudoautosomal regions](https://en.wikipedia.org/wiki/Pseudoautosomal_region) of the X chromosome.

Requirements

This software requires a functioning R installation and several packages within that installation. The required packages are:

* VariantAnnotation
* ggplot2
* gridExtra

If VariantAnnotation is not installed, it can be installed through the “BiocManager” package. To do this, please install BiocManager using install.packages("BiocManager"), if not already installed, and then run BiocManager::install("VariantAnnotation") to install VariantAnnotation. The ggplot2 and gridExtra packages can be installed using the standard install.packages() command.

If the script fails to see your R installation (e.g. if Rscript: command not found error message is displayed), you can edit the check\_X\_chr\_vcf\_ploidy.sh script and replace the Rscript command with the full path of your Rscript executable (that has the required packages installed).

### Correcting male non-PAR X-chromosome coding in pre-imputed VCFs

To convert your X chromosome male non-PAR pre-imputed genotypes from “0/0” and “0/1” coding to “0/0” and “0/2”, we provide a software tool. Please only use this tool if you are sure that

1. your male genotypes are coded incorrectly (e.g. males only have “0/0” and “0/1” genotypes in the [non-PAR regions of chromosome X](https://en.wikipedia.org/wiki/Pseudoautosomal_region#Location)).
2. you know for certain the genome reference build of your genotypes (either hg19=GRCh37 or hg38=GRCh38)
3. you’ll be responsible for checking that the converted genotypes are correct in males.

Downloading the tool

The scripts can either be downloaded through the browser, or by using git (if installed).

**Browser**:

* Go to <https://github.com/long-covid-hg/LongCovidTools/tree/main/Xchr_Coding_Test>
* Download the files and copy them to your desired directory on a Unix machine.
* Set the permissions for the shell script:

chmod +x convert\_het\_males\_to\_althom.sh

**Git**:

* Go to your desired work folder on your Unix system (warning: ensure that no subfolder named “LongCovidTools” already exists in this location)
* Clone the repo:

git clone https://github.com/long-covid-hg/LongCovidTools.git

* Copy out the required files: cp LongCovidTools/Xchr\_Coding\_Test/\* ./
* Delete the repo’s folder: rm -rf LongCovidTools
* Set the permissions for the shell script:

chmod +x convert\_het\_males\_to\_althom.sh

Using the tool

The tool is written in bash and uses awk and gzip, which are typically pre-installed on most linux systems.

You will need to provide the tool with the pre-imputed X chromosome .vcf(.gz) file you want to test and the [plink-format .fam file](https://www.cog-genomics.org/plink/1.9/formats#fam) from the pre-imputed genotypes of the same samples that are in your .vcf(.gz) file. If you don’t already have a .fam file, you can build one by creating a headerless six-column space-delimited file with sample ID in both columns 1 and 2, zeros in columns 3 and 4, sex (“1” for male, “2” for female, without the quotes) in column 5 and “-9” (without the quotes) in column 6. Please make sure that the sample IDs in column 1 match the format of those in the last VCF header line (starting “#CHROM”), otherwise this tool will fail to work. The .fam file must contain all individuals in the VCF file - the script is designed to fail otherwise.

To run the tool, simply type:

./convert\_het\_males\_to\_althom.sh myfile.vcf.gz myfile.fam [hg19|hg38]

Where the last argument is either hg19 or hg38 depending on your genome build. It is important to get the correct build as the script uses this to determine the correct PAR boundaries, as it uses knowledge of these boundaries to avoid correcting male genotypes in the PAR regions.

The tool will then check the input files and either provide error messages for troubleshooting, or will create an output file with converted male non-PAR genotyped - the script will print the name of the newly created gzipped VCF.

The script will replace the following genotypes:

* 1/0 with 1/1
* 0/1 with 1/1
* 1|0 with 1|1
* 0|1 with 1|1

All other genotypes in males (e.g. “0”, “1” “0/0”, “1/1”, “0|0”, “1|1” and “./.”) and all genotypes in females will be left as they are. If you input a VCF that contains either multiple genotype fields (e.g. “GT:DS”-->“0/0:0.01”) or a single genotype field that is not of format “GT”, the script will ignore these fields and the output file will be identical to the input file. If you have additional genotype fields in your pre-imputed chromosome X file, please remove these before rerunning this script.