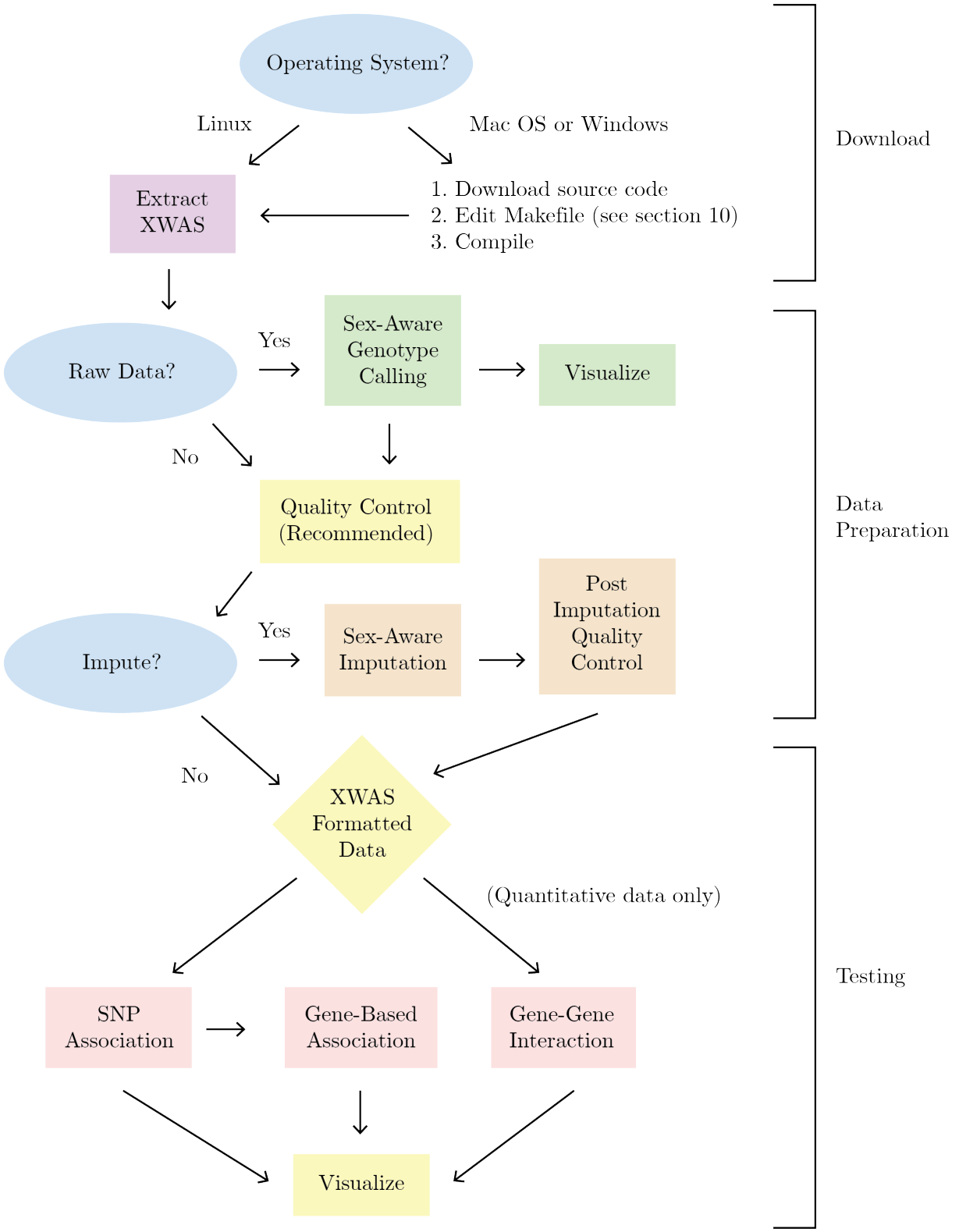
`XWAS` (version 3.0): a toolset for chromosome X-wide data analysis and association studies

Many members of Alon Keinan's lab have contributed between 2012 and 2018 to the design, development, and testing of different versions of this software package and the analytical methods involved.

They are listed here in alphabetical order:

Leonardo Arbiza, Arjun Biddanda, Diana Chang, Feng Gao,
Yingjie Guo, Lauren J. Lo, Alexandre Lussier, Li Ma, Sean McCoy,
Yuhuan Qiu, Kaixiong Ye, Liang Zhang, Zilu Zhou

May 1, 2018


Operating System?
Linux
Mac OS or Windows
Extract XWAS
1. Download source code
2. Edit Makefile (see section 10)
3. Compile
Download
Raw Data?
Yes
Sex-Aware Genotype Calling
Visualize
No
Quality Control (Recommended)
Impute?
Yes
Sex-Aware Imputation
Post Imputation Quality Control
Data Preparation
No
XWAS Formatted Data
(Quantitative data only)
SNP Association
Gene-Based Association
Gene-Gene Interaction
Visualize
Testing

Contents

1 Background

This manual describes the implementation and usage of the XWAS software package (chromosome X-Wide Analysis toolSet, v3.0). XWAS is designed to perform single-marker and gene-based association analyses of chromosome X. It also includes quality control, imputation, genotype calling, and visualization tools. For convenience, "X" is used to refer to chromosome X throughout the manual.

XWAS version 3.0 is based on version 2.0. In addition to various bug fixes, XWAS version 3.0 includes the following new features:

- Fixed- and random-effects meta-analyses of XWAS association results in females or males only.
- A suite of visualization tools for XWAS results, including QQ and Manhattan plots.
- X-wide genomic control for various XWAS association tests.
- An option to parallelize XWAS testing with male genotypes on X coded as 0/1 and as 0/2.
- Calculation and output of confidence intervals of effect sizes or odds ratios for various XWAS association tests.
- Improved powerful gene-gene interaction testing based on pairwise SNP P-values.
- Parallelized genotype calling and the ability to visualize genotyping clusters.

For more details about v1.0, please refer to:

Chang D, Gao F, Slavney A, Ma L, Waldman YY, Sams AJ, Billing-Ross P, Madar A, Spritz R, Keinan A. 2014. No eXceptions: Accounting for the X chromosome in GWAS reveals X-linked genes implicated in autoimmune diseases. *PLoS ONE* 9(12): e113684.

Gao F, Chang D, Biddanda A, Ma L, Guo Y, Zhou Z, Keinan A. 2015. XWAS: a toolset for chromosome X-wide data analysis and association studies. *Journal of Heredity*, 106(5):666-671.

The Keinan Lab continues to develop, support, and release updates of the XWAS package on a regular basis. To receive updates about future versions or any bugs, please sign up for our mailing list. Please report bugs or ask questions by contacting us at `keinanlab.xwas@gmail.com`. Our lab is also involved in many collaborative projects, including projects in which we apply our expertise and software for analysis of the X chromosome to existing data from genome-wide association studies (thus far, we have analyzed over 100 GWAS and found over 20 novel X-linked risk factors or QTLs). We welcome additional collaborations. Please email Alon Keinan (`ak735@cornell.edu`) to explore such opportunities.

2 Downloading and Extracting XWAS

XWAS can be freely downloaded from the Keinan lab website. Use the following command to extract the package locally:

```
tar -zxvf XWAS_v3.0.tar.gz
```

All necessary binary files and scripts are included. However, users can also compile XWAS on their system by downloading the source code and building the executable using the following commands:

```
tar -zxvf xwas_src.tar.gz
cd xwas_src
make
```

For the remainder of this manual, `$path` denotes the location of the XWAS package directory. The provided XWAS binary is optimized and compiled for LINUX. To compile for Windows or MAC OS, consult the troubleshooting section (section 10).

3 Genotype Calling

If you do not wish to perform sex-aware genotype calling or do not have access to raw intensity data, you may proceed to the next section.

XWAS can perform sex-aware genotype calling from raw intensity data based on an Affymetrix genotype array. This reports genotypes for each SNP-by-individual combination in an XWAS formatted dataset. XWAS also includes methods to summarize the differences from another set of genotype calls and visualize the genotyping intensity clusters. Our implementation is based on `BIRDSUITE` (Altshuler *et al.* 2008).

3.1 Prerequisites and Set Up

We currently support intensity data from Affymetrix Genome-Wide Human SNP Array 6.0 and 5.0. Genotype calling requires Java 1.5+, Python 2.5.2+, and R 2.4+; the Python library `numpy`; and the R library `mclust`.

Navigate to `$path/XWAS/genotyping_pipeline/bin`. If you are the admin of your system, run the following command:

```
sudo easy_install --script-dir=./ *py[VERSION].egg
```

where `VERSION` is your system's version of Python. Note that the egg packages are optimized for Python 2.5 and 2.7. Contact our team for compatibility with other version of Python. If you are not the admin of your system, run the following command:

```
easy_install --install-dir=INSTALL_DIR --script-dir=./ *py[VERSION].egg
```

where `VERSION` is your system's version of Python and `INSTALL_DIR` is your desired location for installing Python packages. Make sure `INSTALL_DIR` is in your `$PYTHONPATH` environment variable.

Next, install the three R packages by running the following commands:

```
R CMD INSTALL -l ./ broadgap.utils_1.0.tar.gz
R CMD INSTALL -l ./ broadgap.cnputils_1.0.tar.gz
R CMD INSTALL -l ./ broadgap.canary_1.0.tar.gz
```

Download the Matlab compiled runtime for your appropriate system (64-bit systems, 32-bit systems). Decompress and install the file by using the following commands:

```
gzip -d MCRInstaller.75.[VERSION].bin.gz
./MCRInstaller.75.[VERSION].bin -console
```

When prompted for a destination directory, reply with `MCR75_glnxa64` for 64-bit systems and `MCR75_glnx86` for 32-bit systems. `MCRInstaller.75.[VERSION].bin` may be deleted after installation.

Download the metadata directory. Place the compressed directory in `$path/XWAS/genotyping_pipeline`. Decompress the directory by using the following command:

```
tar -zxvf metadata.tar.gz
```

3.2 Usage

Navigate to `$path/XWAS/genotyping_pipeline`.

```
./call_genotypes.py -c FILE.params -i FILE.cels -o OUTPUT [--parallel]
```

Required arguments:

`--config, -c`	The path to the parameter file (see 3.2.1)
`--individuals, -i`	The path to the individuals file (see 3.2.2)
`--output, -o`	The path to your desired output directory

Optional arguments:

`--parallel, -p`	Sample plates can be genotyped in parallel. Invoking this flag causes `call_genotypes.py` to produces a series of scripts which can be run in parallel.
`--help, -h`	Show brief descriptions of all `call_genotypes.py` flags and exit.

The parallel call will produce one `run_PLATE_birdsuite.sh` script per batch and one `run_birdsuite_to_plink.sh` script. The batch scripts must be run first in parallel before the `run_birdsuite_to_plink.sh` is invoked. The scripts can be invoked with the simple command:

```
./run_PLATE_birdsuite.sh
```

The output of the genotyping procedure is an XWAS formatted dataset (`.tped`/`.tfam`) and other output files detailing the specifics of the genotype calling. The full description of each output file can be found here.

3.2.1 Parameter File

See `example.params` for an example parameter file. The following parameters are required:

`chipType`	Name of Affymetrix chip type used. Accepted arguments are `GenomeWideSNP_6` and `GenomeWideSNP_5`.
`famFile`	Path to `.fam` file for your data.

The following parameters are optional:

`genomeBuild`	hg18 is the default. Currently, hg17 and hg18 are supported.
`celMap`	Use a cel map file to convert complicated cel names to common names. Each line of the file should contain two columns: the cel name, and the new individual name. See `example.celMap` for an example.
`threshold`	Each genotype call is made with a certain confidence (0 indicates most confident, 1 indicates least confident). Define a confidence cutoff value for which to exclude data from your final dataset. Default is 0.1.
`outputName`	Define the file name root for the output files. Default is "Output".
`apt_probeset_summarize.force`	Use this option if one or more of the cel files were processed with a non-standard CDF.
`noLsf`	By default, Load-Sharing Facility (LSF) is used to parallelize parts of the genotyping procedure. Use this option if you do not have LSF installed on your system, or do not wish to use LSF.
`canary.priors`	Use a specific models file to aid in clustering common copy-number polymorphisms. For example, setting this option to `metadata/GenomeWideSNP_6.CEU.canary_priors` will result in better clustering for samples of European ancestry. The default is a models file built using all 270 HapMap phase I and II samples. Additional models files can be found in `metadata/*priors`.
`canary.allele_freq_weight`	How much to weight observed CNP frequency from HapMap to aid in clustering. The recommended value is 32 if samples are of European ancestry and the `GenomeWideSNP_6.CEU.canary_priors` models file is used.

3.2.2 Individuals File

See `example.cels` for an example individuals file. The file should contain one line per individual. Each line has three columns:

1. Path to the cel file.
2. Gender of the individual. 0 indicates female, 1 indicates male, 2 indicates unknown.
3. Batch of the cel file. Samples that were run on the same plate belong to the same batch. For accurate analysis, all individuals from a given batch must be run together.

WARNING: The root name of your individuals file cannot be the same as a batch name (for example, you cannot name your individuals file `ANNUL.cels` if you also have a batch called `ANNUL`).

3.3 Comparing Genotype Calls

After genotyping your dataset, you may wish to compare your new calls to a previous set of genotype calls. The script `$path/XWAS/genotyping_pipeline/compare_calls.py` summarizes the differences between two datasets for you.

```
python compare_calls.py OLD_ROOT NEW_ROOT
```

Required arguments:

`OLD_ROOT`	File root name of your old calls
`NEW_ROOT`	File root name of your new calls

Make sure both datasets are in the same directory as `compare_calls.py`. This script will produce four output files.

1. `included_snps.diff`: one row per differentially called SNP with the following columns:

   `[SNP_ID] [NUM_DIFF_CALLS]`

   First column is SNP identifier. Second column is number of individuals with a differential call for this SNP. If no individuals were differentially called for a given SNP, it will not be included in this file.

2. `excluded_snps.diff`: one row for each SNP that was excluded from one of the datasets. Columns are:

   `[SNP_ID] [OLD_ROOT] [NEW_ROOT]`

   First column is SNP identifier. `OLD_ROOT` is 1 if `SNP_ID` was called in the old dataset and 0 if it was not. `NEW_ROOT` follows the same format. If a SNP was included in both datasets, it will not be included in this file.

3. `included_individuals.diff`: one row per individual that was included in both datasets. Columns are:

```
[IID] [NUM_DIFF_CALLS]
```

First column is individual identifier. Second column is number of differentially called SNPs between the two datasets for this individual. If no SNPs were differentially called for a given individual, this individual will not be included in this file.

4. `excluded_individuals.diff`: one row for each individual that was excluded from one of the two datasets. Columns are:

```
[IID] [OLD_ROOT] [NEW_ROOT]
```

First column is individual identifier. `NEW_ROOT` is 1 if individual was included in the new calls and 0 if missing. `OLD_ROOT` follows the same format. If an individual was included in both datasets, it will not be included in this file.

3.4 Visualization

After genotyping your data and comparing its calls, you may wish to visualize the genotype intensity clusters created to call genotypes. For instance, this may help you determine the validity of a differential call. The script `$path/XWAS/genotyping_pipeline/visualize_clusters.py` uses output files from our genotyping procedure to visualize the clusters for any number of SNPs.

```
./visualize_clusters.py -a FILE.allele_summary -t FILE -s rsid
                [-c example.celMap] [-p] [-f] [-m]
```

Required arguments:

`--allele-summary, -a`	A `BATCH.allele_summary` file produced by our genotyping procedure. All individuals in this `BATCH` will be included in the visualization.
`--tfile, -t`	The prefix of an XWAS transposed fileset (`.tped`/`.tfam`) which contains all individuals in `BATCH` and their genotypes of the SNP(s) of interest.
`--snp, -s`	Either the name of a single SNP or a text file with a list of SNP indentifiers (one per line). These are the SNPs which will be visualized.

Optional arguments:

`--cel-map, -c`	If you used a cel map in the genotyping procedure, supply it here.

`--png, -p`	Use this flag to save the cluster visualizations as PNGs (without this flag, visuals are simply displayed). Plots will be saved as `SNPID.png` (or `SNPID_female.png` and `SNPID_male.png` for females only or males only respectively) where `SNPID` is the SNP identifier of the visualized SNP(s).
`--female, -f`	Use this flag to visualize female individuals only.
`--male, -m`	Use this flag to visualize male individuals only.

4 Quality-Control

XWAS performs quality control (QC) that applies standard autosomal GWAS quality control steps as implemented in PLINK (Purcell *et al.* 2007) and `SMARTPCA` (Price *et al.* 2006), as well as procedures that are specific to X.

The dataset is initially split into two datasets, one consisting only of males and the other of females. The general quality control steps are performed separately on the two datasets, which are then merged before the X-specific quality control is applied. See 4.2 and 4.3 for details.

For a full example of running quality control, navigate to `$path/XWAS/example/qc` and execute `run_example_qc.sh`

4.1 Usage

```
$path/XWAS/bin/run_QC.sh params_file.txt [-l] [-a] [-v] [-g] [-s]
```

Required arguments:

`params_file.txt`	The path to the parameter file (see 4.1.1)

Optional arguments:

`--help, -h`	Show brief descriptions of all QC flags and exit
`--save-logs, -l`	Save logs from QC to `./$filename_QC_logs`
`--save-all, -a`	Save all intermediate files from QC, including logs, to `./$filename_QC_intermediate_files`
`--verbose, -v`	Unsupress XWAS output
`--debug, -g`	Save logs and intermediate files, and unsupress XWAS output

`--skip-ibd, -s`	Skip the identity-by-descent relatedness filtering step in the quality control procedure. This analysis can be highly time and memory consuming if sample size is large. **WARNING:** we only recommend skipping this step if you are confident your sample does not contain relatedness or you have elected to use your own method of identity-by-descent filtering. **This flag is not to be used lightly and may bear the consequence of inaccurate association analysis.**

The output dataset called `$filename.preprocessed_final_x` contains the final data for X chromosome only. The output dataset called `$filename.preprocessed_final` contains the final data for all chromosomes. Quality control also outputs a file called `$filename.preprocessed.covar`, which contains the covariate information from population structure determined by `SMARTPCA`.

QC uses a compiled `SMARTPCA` executable in the `$path/XWAS/bin` folder. If you encounter errors running the executable, try downloading the source code for `EIGENSOFT` from the the Price Lab and compiling locally.

4.1.1 Parameter File

See `$path/XWAS/example/qc/example_params_qc.txt` for an example parameter file. Each parameter should be listed on a separate line in the format "`parameter_name` [value]". The parameters are outlined below:

`filename`	Name of dataset file (without the extension)
`xwasloc`	Location of XWAS executable, `$path/XWAS/bin`
`eigstratloc`	Location of the `SMARTPCA` and `convertf` executables, `$path/XWAS/bin`
`exclind` [0/1]	This parameter allows you to exclude a predefined set of individuals. To specify individuals to be removed, set the value to `1` and list individuals in a file named `$filename_exclind.remove`, where `$filename` is the same as the `filename` parameter. Otherwise, set the value to `0`.
`excludexchrPCA` [YES/NO]	Set value to `YES` to exclude X data when calculating PCA, or `NO` to include it.
`build` [17/18/19]	Specify the genome build of the dataset. QC supports `17`, `18` or `19` for hg17, hg18 and hg19 respectively.
`alpha` [α]	Sets Bonferroni-corrected P-value (divided by the number of SNPs in the dataset) for significance in exclusion criteria, recommended is 0.05
`plinkformat` [ped/bed]	Specify whether data is in `.ped/.map` or binary format
`maf` [α]	Variants with minor allele frequency less than α are removed, recommended is 0.005
`missindiv` [α]	Individuals with missing genotype rate greater than α are removed, recommended is 0.10

`missgeno` [α]	Variants with missing genotype rate greater than α are removed, recommended is 0.10
`numpc`	Number of PCA principal components to include as covariates, recommended is 10
`related` [α]	Individuals with a proportion of shared IBD segments above the cutoff proportion are considered for removal, recommended is 0.125 which corresponds to the relatedness between first-degree cousins
`quant` [0/1]	Specify whether the phenotype is quantitative (1) or binary (0)

4.2 General Quality Control

The general quality control steps are performed separately on males and females.

1. Removing interdependent individuals. For parent-child pairs, parents are retained and children are removed. For siblings, one is arbitrarily retained and the rest are removed. All individuals remaining are independent from the standpoint of known familial relationship (their relationships will be further identified by identity-by-descent-based analysis).

2. SNPs are filtered if they have a missingness rate above some threshold, a minor allele frequency (MAF) below some threshold, or if they are not in Hardy-Weinberg equilibrium in females. Thresholds are set in QC parameters.

3. Variants are filtered if their missingness is significantly correlated with phenotype. **Note:** this step is only applied to case-control studies. Significance is set in QC parameters.

4. Individual samples are removed if they are inferred to be related based on the proportion of shared identity-by-descent segments. Identity-by-descent is calculated using the `--genome` flag in PLINK. To avoid removing too many samples, only one individual from each pair of related individuals is removed. **Note: this assumes that the samples are from a homogeneous population; if they are from different populations, this analysis will be problematic.** **Note: There may be some issues with PLINK relatedness filtering method in removing more individuals than expected.*

5. Individuals are removed if they have a genotype missingness rate above some threshold or if their reported sex does not match the heterozygosity rates observed on X. Threshold is set in QC parameters.

6. Population structure is assessed with the software `SMARTPCA` from the package `EIGENSTRAT` (Price *et al.* 2006) and outlier individuals are removed.

4.3 X-Specific Quality Control

The X-specific quality control steps are performed on males and females together after general quality control.

1. Variants on chromosomes other than X are removed, as well as variants in the pseudoautosomal regions (PARs) on X.

2. Variants are filtered if they have significantly different MAF between male and female controls. Significance is set in QC parameters.

3. Variants are filtered if they have significantly different missingness between male and female controls. Significance is set in QC parameters.

5 Imputation

If your dataset is already imputed or if you do not wish to impute your data, you may skip this step.

XWAS performs sex-aware X imputation with `IMPUTE2` (Howie *et al.* 2009) using 1000 Genomes Phase III reference data. `IMPUTE2` handles X differently from the autosomes by reducing the effective size of the population (N_e) by 25% for X and setting the male heterozygous genotype probability to 0 for X.

5.1 Data Preparation

The imputation reference files (1000 Genomes Phase III) are in build human genome 19 (hg19) and `IMPUTE2` requires all data to be on the positive strand alignment. If you already know your dataset is build hg19 and on the positive strand alignment, you may skip this step.

Otherwise, check the build and strand alignment of your dataset with our script `$path/XWAS/bin/check_genome_build_and_strange_alignment.pl`. This script uses the X chromosome to perform position-based and SNP-identifier-based allele concordance checks.

```
perl check_genome_build_and_strange_alignment.pl FILE.bim
      1000GP_Phase3_chrX_NONPAR.legend OUTPUT.txt
```

Required arguments:

`FILE.bim`	The `.bim` file from your dataset
`1000GP_Phase3_chrX_NONPAR.legend`	The 1000 Genomes Phase III reference file (see 5.2.2 to obtain this file)
`OUTPUT.txt`	The desired output file

The script will produce two output files:

1. `OUTPUT.txt`: summarizes the results of the concordance checks. If position-based allele-concordance is below 95%, this indicates your dataset is likely not in build hg19. Our imputation procedure will identify this and convert your data to build hg19 automatically. If ID-based allele-concordance is below 95%, this indicates your dataset is likely not all on the positive strand. Will Rayner at the Wellcome Trust Centre for Human Genetics has provided a number of useful scripts which can fix strand alignment. These can be found

on his website.

2. `OUTPUT.txt.inconsistent`: one row for each SNP which was inconsistent between `FILE.bim` and the reference file. Each row has two columns: the position or SNP identifier of the inconsistent SNP, and a column which indicates whether the inconsistency was position-based or ID-based.

5.2 Usage

There are four steps to the imputation procedure.

1. Generate all the necessary shell scripts for imputation. Make sure your parameter file `FILE.par` (see 5.2.1) is in the same directory as `make_imputation_files.py`.

```
python make_imputation_files.py FILE.par
```

2. Pre-imputation step.

```
./FILE_preimpute.sh
```

3. There are two options for the imputation step. If you choose to run all of the jobs in sequence:

```
./FILE_impute2_run_all.sh
```

   If you choose to run the jobs in parallel, type each command from `FILE_impute2_run_all.sh` in a separate command window.

   WARNING: running parallele jobs on a local machine may be very memory-consuming. We recommend running the jobs in sequence if using a local machine.

   BENCHMARK: On an 8 core and 7.8 GB RAM Ubuntu 14.04LTS machine, running a sample of 198 individuals and 32,526 SNPs divided into 30 jobs took approximately 20 minutes and 1GB per job.

4. The final step combines the files from the different jobs in the previous step.

```
./FILE_impute2_cat.sh
```

5.2.1 Parameter File

See `$path/XWAS/imputation_pipeline/example.par` for an example parameter file. The example file starts with explanatory comments which have a `#` sign at the beginning of each line. These lines are ignored and not required. The parameters are outlined below:

`FILE`	Root name of pre-imputation dataset (without extension)

OUTPUT	Desired root name of output dataset (without extension)
NJOBS	Number of parts to break chromosome X into. These parts become jobs which can be run sequentially or in parallel. Default value is 31.
BUILD	Build of the data (17 for hg17 and 18 for hg18)
FILELOC*	Location of the pre-imputation dataset
REFLOC*	Location of the reference files (see 5.2.2), $path/XWAS/imputation_pipeline/imputation_reference_files/
TOOLSLOC*	Location of the imputation tools, $path/XWAS/imputation_pipeline/imputation_tools/
RESLOC1*	Location to store the output files after the pre-imputation step (make sure this folder exists)
RESLOC2*	Location to store the output files after the imputation step (make sure this folder exists)
FINALRESLOC*	Location to store the final results, which include the dataset after imputation and the imputation info files (make sure this folder exists)
MAFRULE	Population name and corresponding minor allele frequency for filtering out the sites in the 1000 Genomes reference file. The value must be one of: AFR.MAF (Africa), AMR.MAF (America), EAS.MAF (East Asia), EUR.MAF (Europe), SAS.MAF (South Asia), ALL.MAF (all above popuations together), combined with $\leq$ VAL. For example, if the sample is from European population and the minor allele frequency threshold is 0.005, the value will be EUR.MAF<=0.005.

***Note**: For file locations, make sure to include the whole path, including the ending "/".

5.2.2 Reference Files Preparation

Download the 1000 Genomes Phase III reference files for X. Additionally, download the file 1000GP_Phase3_chrX_NONPAR.legend.gz. Decompress the files:

```
tar -zxvf 1000GP_Phase3_chrX.tgz
gzip -d 1000GP_Phase3_chrX_NONPAR.hap.gz
gzip -d 1000GP_Phase3_chrX_NONPAR.legend.gz
```

The files needed for imputation are genetic_map_chrX_nonPAR_combined_b37.txt, 1000GP_Phase3_chrX_NONPAR.hap, and 1000GP_Phase3_chrX_NONPAR.legend. Move them into $path/XWAS/imputation_pipeline/imputation_reference_files. All other files will not be used and can be deleted.

5.3 Post Imputation Quality Control

After imputation, we recommend a post imputation QC step. This step resembles the QC from section 4.

```
$path/XWAS/bin/xwas_qc.post_imputation.sh params_qc.txt
```

Required arguments:

`params_qc.txt` The path to the parameter file (see 4.1.1)

Note that the same set of parameters from QC can be re-used, except the genome build must be hg19. The filename also changes after imputation.

6 Variant Association Testing

XWAS contains new functions for conducting X-wide association studies while keeping all previous functions that are provided by the commonly-used PLINK. The newly-added functionality is described below.

6.1 General Flags

```
--xhelp
--xwas
--run-all
```

`--xhelp` shows brief descriptions of all the XWAS-specific functions. In order to use any of the options below, the `--xwas` option needs to be called. The `--run-all` option conveniently runs the sex-stratified test (with both Fishers and Stouffers methods), sex-difference test, variance-heterogeneity test, and Clayton's test.

6.2 Allele Frequency

```
--freq-x
--freqdiff-x α
```

`--freq-x` outputs the allele frequencies for each polymorphic site in males and females separately.

`--freqdiff-x` tests for significantly different minor allele frequency between male and female controls, according to P-value threshold α. This option is used during QC to filter out SNPs that have significantly different minor allele frequencies between males and female controls.

6.3 Sex-Stratified Association Test

```
--strat-sex [--fishers] [--stouffers] [--ci α] [--xbeta] [--gc]
[--multi-xchr-model]
```

The sex-stratified test carries out an association test in males and females separately and then combines the two test results to produce a final sex-stratified significance value for each SNP. The `--fishers` modifier combines the P-values using Fisher's method (this is the default method for combining P-values). The `--stouffers` modifier combines the P-values using Stouffer's method. Our sample-size-based analysis of Stouffer's method within this software follows that of Willer *et al.*

Adding `--ci` outputs the lower and upper bound of the confidence interval where α is the desired coverage, e.g. 0.95 or 0.99. It also outputs standard error. The `--xbeta` option outputs regression coefficients (in place of the default odds ratios) for each sex. Adding `--gc` outputs genomic control adjusted P-values to a separate output file, `FILE.adjusted`.

Use `--multi-xchr-model` to output results when male genotypes are coded as 0/1 (males are considered equivalent to female heterozygotes) and 0/2 (males are considered equivalent to either female homozygote) in parallel. This produces two result files: `FILE.01` and `FILE.02` for 0/1 coded results and 0/2 coded results respectively.

6.4 Sex Difference in Effect Size

```
--sex-diff [--xbeta] [--multi-xchr-model]
```

This outputs the difference in effect size between males and females for each SNP. The `--xbeta` option outputs regression coefficients (in place of the default odds ratios) for each sex.

Use `--multi-xchr-model` to output results when male genotypes are coded as 0/1 (males are considered equivalent to female heterozygotes) and 0/2 (males are considered equivalent to either female homozygote) in parallel. This produces two result files: `FILE.01` and `FILE.02` for 0/1 coded results and 0/2 coded results respectively.

6.5 Meta Analysis

```
--meta-analysis [file1 file2 file3 ...]  [+ female] [+ male]
```

Meta-analysis function for males or females only, in which two or more XWAS result files (`file1`, `file2`, etc) can be combined in fixed-effects and random-effects meta-analysis. Use the `+ female` option for females only and the `+ male` option for males only. Each results file must contain the following columns:

`SNP`	SNP idenitifier
`OR`	Odds ratio (or `BETA`)
`SE`	Standard error. Output when `--ci` α is used.
`P`	P-value from test

If the `+ male` or `+ female` options are used, the function will search for `OR_M`, `SE_M`, `P_M` or `OR_F`, `SE_F`, `P_F` respectively. The output file, `xwas.meta`, has the following columns:

`N`	Number of valid studies for this SNP
`P`	Fixed-effects meta-analysis P-value
`P(R)`	Random-effects meta-analysis P-value
`OR`	Fixed-effects OR estimate
`OR(R)`	Random-effects OR estimate
`Q`	P-value for Cochrane's Q statistic
`I`	I^2 heterogeneity index (0-100)

6.6 Variance Heterogeneity Test

```
--var-het [--gc]
--var-het-weight
--var-het-comb
```

The variance-heterogeneity test tests for X-linked association by looking for higher phenotypic variance in heterozygous females than homozygous females at each SNP. Adding `--gc` also outputs genomic control adjusted P-values to a separate output file, `FILE.adjusted`.

The weighted-heterogeneity test tests for X-linked association by a weighted regression approach to account for the variance inflation. Finally, the combined test combines the variance-based test and the weighted association test into a single test statistic using the Stouffer's Z score method. All heterogeneity methods are described in further detail in Ma *et al.*

6.7 Clayton's Test

```
--clayton-x [--gc]
```

This tests for association on X using Clayton's method under the assumption that the allele frequency does not vary with sex. This method is described in further detail in Clayton. Adding `--gc` also outputs genomic control adjusted P-values to a separate output file, `FILE.adjusted`.

6.8 Epistasis Test

```
--xepi [--set set.file] [--set-by-all] [--covar covar.file] [--xepi1 α]
[--xepi2 α]
```

This tests SNP × SNP epistasis for qualitative/quantitative phenotypes. For qualitative phenotypes, `--xepi` is essentially the same as `--epistasis` in PLINK. Both use a logistic model and test the interaction term with a Wald test. For quantitative phenotypes, `--epistasis` in PLINK uses a linear model with a Wald test, while `--xepi` uses a linear model and then tests the interaction with a t-test. Adding `--covar` supports the inclusion of one or more covariates.

There are four different modes to assign which SNPs are tested:

1. ALL × ALL: `--xepi`
   Tests all X chromosome SNPs against each other.
2. SET1 × SET1: `--xepi --set set.file`
   Tests all SNPs in `set.file` against each other, where `set.file` contains one set of SNPs.
3. SET1 × ALL: `--xepi --set set.file --set-by-all`
   Tests all SNPs in `set.file` against all X chromosome SNPs, where `set.file` contains one set of SNPs.
4. SET1 × SET2: `--xepi --set set.file`
   Tests the two sets of SNPs in `set.file` against each other, where `set.file` contains two sets of SNPs.

There can only be one or two sets in the `set.file`, which should be in the following format:

```
SET_A
rs001
rs002
END

SET_B
rs101
rs102
rs103
END
```

The epistasis test outputs two files: the first contains the pairwise P-values for each SNP × SNP pair; the second records the summary information for each SNP. `--xepi1` sets the output threshold for SNP × SNP P-values and `--xepi2` sets the output threshold for P-values in summary file. The default for both values is 1e-4.

6.9 PLINK Functions

In addition to these new functions, all pre-existing PLINK functionality carries over to XWAS. In particular, the options below can be useful in carrying out X-wide association studies.

```
--logistic
--linear
```

The options to carry out logistic regression and linear regression for binary and quantitative traits respectively. Note that male genotypes on X will follow 0/1 coding by default, i.e. a male allele is considered as equivalent to a single female copy.

```
--xchr-model 2
```

This option can be used together with `--logistic` or `--linear` to code male genotypes on X as 0/2, i.e. males are always considered equivalent to either female homozygote. When relevant, the above newly-added tests can also consider either 0/1 or 0/2 coding using this option.

```
--sex
```

If the `--sex` flag is added, then sex will be entered as a covariate in the model (coded 1 for male, 0 for female). This can be used together with `--logistic` or `--linear` or any of the relevant newly-added tests.

6.10 Examples

For examples of running the functions above, we have included two samples datasets (one with quantitative phenotypes and one with qualitative phenotypes). Navigate to `$path/XWAS/example/sample_data` to find the datasets. Below are some commands you can use to run examples of XWAS tests using the sample datasets:

1. Get allele frequencies for males and females separately
   `../../bin/xwas --bfile dummy_case1 --xwas --freq-x`
2. Perform the sex-stratified test, using Fisher's method to combine P-values
   `../../bin/xwas --bfile dummy_case1 --xwas --strat-sex --fishers`
3. Perform the sex-stratified test, using Stouffer's method to combine P-values
   `../../bin/xwas --bfile dummy_case1 --xwas --strat-sex --stouffers`
4. Perform the sex-difference test
   `../../bin/xwas --bfile dummy_quant1 --xwas --sex-diff`
5. Perform the variance-heterogeneity test (with covariates)
   `../../bin/xwas --bfile dummy_quant1 --xwas --var-het --covar dummy_quant1.covar`
6. Perform linear regression using 0/2 coding for males
   `../../bin/xwas --bfile dummy_quant1 --xchr-model 2 --linear`

The output files will be column-delimited tables with the prefix `xwas`. To change the output file name, add the flag `--out filename`.

7 Gene-Based Association Testing

Gene-based testing builds upon SNP-level analyses by using the P-values obtained for each SNP. Our implementation is based on the VEGAS (Liu *et al.* 2010) framework. Here, the procedure is modified to utilize the truncated tail strength (Jiang *et al.* 2011) and truncated product (Zaykin *et al.* 2002) method to combine individual SNP-level P-values. Gene-based testing requires the R libraries `corpcor` and `mvtnorm`.

7.1 Automated Gene-Based Association Testing

For your convenience, we provide a script `$path/XWAS/bin/gene_based_test_automate.sh` that automatically searches for the temporary files XWAS creates when you run the SNP-level analyses, and generates the gene based results for you.

```
$path/XWAS/bin/gene_based_test_automate.sh params.txt
```

Required arguments:

`params.txt`	Parameter file for automated gene testing (see 7.1.1)

The output of the gene-based association test will have the following columns:

`[gene] [reps] [tail_p_value] [prod_p_value]`

The first column states the name of the gene. The second is the number of (adaptively determined) bootstrap replicates. The third column is the P-value for the gene using the truncated tail strength method. The last column shows the P-value calculated using the truncated product method.

For a full example of running the automated gene-based tests, navigate to the directory `$path/XWAS/example/gene_automated` and execute `./run_example_auto_test.sh`.

7.1.1 Parameter File

Parameters for the automated gene test should be provided in a file similar to `$path/XWAS/example/gene_automated/example_params_gene_test_auto.txt`.

`filename`	Root name of binary dataset without extension
`xwasloc`	Location of XWAS executable, `$path/XWAS/bin`
`genescriptloc`	Location of `truncgene.R`, `$path/XWAS/bin`
`genelistname`	File with list of genes to test (see 7.1.2)
`pvfolder`	Location of the SNP-association results files
`buffer`	Buffer region around genes (in base-pairs) to account for SNPs that may be slightly outside the defined gene region

The results will output to `pvfolder`.

7.1.2 Gene File

Genes to test should be provided in a file similar to `$path/XWAS/example/gene_automated/gene_list.txt`, with one line per gene and the following columns:

`[chromosome #] [gene start position] [gene end position] [gene name]`

Gene positions in this file should be in the same build as the dataset. We recommend using the "UCSC knownCanonical" transcript file for a comprehensive list of chromosome X genes. We have also provided correctly formatted chromosome X gene lists for hg19 and hg38 in the `$path/XWAS/genes` directory.

7.2 Gene-Based Association Testing

```
$path/XWAS/bin/gene_based_test.sh params.txt
```

Required arguments:

`params.txt`	Parameter file for gene association testing (see 7.2.1)

The output of the gene-based association test will have the following columns:

`[gene] [reps] [tail_p_value] [prod_p_value]`

The first column states the name of the gene. The second is the number of (adaptively determined) bootstrap replicates. The third column is the P-value for the gene using the truncated tail strength method. The last column shows the P-value calculated using the truncated product method.

For a full example of running the gene-based tests, navigate to the directory `$path/XWAS/example/gene_test` and execute `./run_example_gene_test.sh`

7.2.1 Parameter File

Parameters should be provided in a file with a similar structure to `example_params_gene_test.txt` in the folder `$path/XWAS/example/gene_test`.

`filename`	Root name of binary dataset without extension
`xwasloc`	Location of XWAS executable, `$path/XWAS/bin`
`genescriptloc`	Location of `truncgene.R`, `$path/XWAS/bin`
`genelistname`	File with list of genes to test (see 7.1.2)
`assocfile`	Association results file (see 7.2.2)

`buffer`	Buffer region around genes (in base-pairs) to account for SNPs that may be slightly outside the defined gene region
`numindiv`	Number of individuals with which to estimate linkage disequilibrium for estimating dependency between single-SNP tests. We recommend a minimum of 200.
`output`	Output filename

7.2.2 Association Results File

Gene based testing requires SNP-level P-values. These should be provided in a file with one line per SNP and the following columns:

```
[SNP name] [SNP P-value]
```

These P-values can be obtained from any of the SNP-based tests in section 6 (e.g. the sex-stratified test, `--strat-sex`).

8 Gene-Based Gene-Gene Interaction Testing

The gene-based gene-gene interaction (GGG) tests are based on a framework we previously developed (Ma *et al.* 2013). The tests combine SNP-based interaction tests between all pairs of SNPs in two genes to produce a gene-level test for interaction between the two genes. GGG tests are built upon the SNP-level interaction test function `--xepi` (see 6.8), and use the P-values obtained for each SNP pair.

There are four GGG tests that use the following P-value combining methods: minimum P-value, extended Simes procedure, truncated tail strength (Jiang *et al.* 2011), and truncated P-value product (Zaykin *et al.* 2002).

Note: currently, the GGG tests can only analyze quantitative phenotypes.

8.1 Usage

GGG testing requires the R libraries `corpcor` and `mvtnorm`.

```
$path/XWAS/bin/gene_based_interaction.sh params.txt
```

Required arguments:

`params.txt`	Parameter file for gene-gene interaction testing (see 8.1.1)

GGG will output two file formats:

1. `[OUTPUT].txt`, which has the following columns:

```
[gene1] [gene2] [min_P_value] [gate_P_value] [tail_p_value] [prod_p_value]
```

The first and second columns state the names of two genes. The remaining columns record P-values for each method: minimum P-value, extended Simes procedure, truncated tail strength, and truncated P-value product.

2. `gene1name_gene2name.qt` records the SNP × SNP pairwise P-values from the epistasis test for each pair of genes. The file format follows the output of the `--xepi` function (see 6.8).

For a full example of running the GGG tests, navigate to `$path/XWAS/example/gene_gene_inter` and execute `run_example_GGG_test.sh`

8.1.1 Parameter File

Parameters should be provided in a file with a similar structure to `example_params_gene_test.txt` in the folder `$path/XWAS/example/gene_gene_inter`.

`filename`	Root name of binary dataset without extension
`xwasloc`	Location of XWAS executable, `$path/XWAS/bin`
`genescriptloc`	Location of `gene_based_inter.R`, `$path/XWAS/bin`
`genelistname`	File with list of genes to test (see 7.1.2)
`genelistname2`	Optional second file with list of genes to test (see 7.1.2)
`buffer`	Buffer region around genes (in base-pairs) to account for SNPs that may be slightly outside the defined gene region
`covarfile`	Optional covariant file
`output`	Output filename

Similar to the `--xepi` SNP test, the gene-based interaction test can be run in a few different modes. By default, it performs interaction tests on all pairs of genes in `genelistname`. If given the optional argument `genelistname2`, it will perform SET1 × SET2. To run the GGG test on all X chromosome genes, use the list of the genes provided in `$path/XWAS/genes`.

9 Visualization

XWAS provides a suite of visualization scripts that conveniently take XWAS association result files as input. They are located in `$path/XWAS/visualization`.

9.1 QQ Plots

When using genomic control, it can be helpful to view the unadjusted and adjusted P-values in a QQ plot. QQ plots can be generated with our script `$path/XWAS/visualization/qqplot.r`. Using this script requires a working installation of R and the R libraries `ggplot2` and `optparse`.

```
Rscript qqplot.r -f results.file [-u pval] [-a gc] [-l λ]
                 [-n output] [-d dir]
```

Required arguments:

`--file, -f`	XWAS results file name (for genomic control, these files typically end with `.adjusted`)

Optional arguments:

`--unadjusted, -u`	Name of column containing unadjusted P-values. Default is `UNADJ`.
`--adjusted, -a`	Name of column containing adjusted P-values. Default is `GC`.
`--lambda, -l`	Genomic inflation factor (printed in XWAS log file).
`--name, -n`	Root name of output PNG files. Default is "qqplot".
`--destination, -d`	Destination directory of all three PNGs. Default is current working directory.
`--help, -h`	Show brief descriptions of all `qqplot.r` flags and exit.

This will create three files: `qqplot_1_unadjusted.png`, `qqplot_2_adjusted.png`, and `qqplot_3_compare.png`; which plot the unadjusted P-values, the genomic control adjusted P-values, and both P-value sets respectively.

9.2 Manhattan Plots

Manhattan plots are common visualizations to identify peaks of significance in GWAS studies. Manhattan plots can be generated with our script `$path/XWAS/visualization/manhattan.r`. Using this script requires a working installation of R and the R libraries `qqman` and `optparse`.

```
Rscript manhattan.r -f xwas.xstrat.logistic [-c chr] [-p pval] [-o
                    output]
```

Required arguments:

`--file, -f`	XWAS results file name. This file should contain four columns: chromosome (`CHR`), base pair (`BP`), SNP identifier (`SNP`), and a P-value column.

Optional arguments:

`--chromosome, -c`	Chromosome number to view. By default, all chromosomes included in the results file are plotted.
`--p-value, -p`	Name of column containing *P*-value. Default is `P`.
`--out, -o`	Name of output PNG. Default is "manhattan".
`--help, -h`	Show brief descriptions of all `manhattan.r` flags and exit.

10 Troubleshooting

Below are some helpful tips for troubleshooting if XWAS is not running correctly for you:

1. Our QC procedure uses a compiled `SMARTPCA` executable in the `$path/XWAS/bin` folder. If you encounter errors running the executable, try downloading the source code for `EIGENSOFT` from the the Price Lab and compiling locally.

2. Many procedures in XWAS require various R libraries and will not run correctly if the library is not installed. To install an R library, type the following into R's command line:

   ```
   install.packages("libraryname")
   ```

   where `libraryname` is the name of the library you wish to install.

3. XWAS requires the newer `liblapack.so.3` in place of the older `liblapack.so.3gf` library. Run `ldconfig -p | grep liblapack` to locate which version is currently installed. In Debian based Linux distros, `liblapack.so.3` can be installed from the `liblapack3` package, while the `lapack` package provides it in RPM based systems.

4. When combining or comparing datasets, it can be important to know which build your dataset is in and whether you have agreeing strand alignment for your datasets. You can check genome build and strand alignment by using a helpful script that our package includes called `check_genome_build_and_strange_alignment.pl`. See 5.1 for more details on how to use and interpret this script.

5. Our binary package is optimized for Linux. If you are attempting to run XWAS on Windows or Mac OS, please download our source code and alter the Makefile for your appropriate system. This can be done by changing the line that says:

   ```
   SYS = UNIX
   ```

   to say:

   ```
   SYS = MAC or SYS = WIN
   ```

   for Mac OS or Windows respectively

6. Because our software is built on PLINK, many PLINK features and flags carry over to XWAS. If you wish to run a PLINK test that you do not see detailed in this manual, first try running the test as detailed in PLINK's documentation. If this does not work as expected, contact us for more details.

For further questions, please contact `keinanlab.xwas@gmail.com`.

11 References

1. Chang D, Gao F, Slavney A, Ma L, Waldman YY, Sams AJ, Billing-Ross P, Madar A, Spritz R, Keinan A. 2014. No eXceptions: Accounting for the X chromosome in GWAS reveals X-linked genes implicated in autoimmune diseases. *PLoS ONE*, 9(12):e113684.

2. Clayton D. 2008. Testing for association on X chromosome. *Biostatistics*, 9(4):593-600.

3. Gao F, Chang D, Biddanda A, Ma L, Guo Y, Zhou Z, Keinan A. 2015. XWAS: a toolset for chromosome X-wide data analysis and association studies. *Journal of Heredity*, 106(5):666-671.

4. Howie BN, Donnelly P, Marchini J. 2009. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genetics*, 5:e1000529.

5. International HapMap Consortium. 2003. The International HapMap Project. *Nature*, 426(6968):789-96.

6. Jiang B, Zhang X, Zuo Y, Kang G. 2011. A powerful truncated tail strength method for testing multiple null hypotheses in one dataset. *Journal of Theoretical Biology*, 277:67-73.

7. Korn JM, Kuruvilla FG, McCarroll SA, Wysoker A, Nemesh J, Cawley S, Hubbell E, Veitch J, Collins PJ, Darvishi K, Lee C, Nizzari MM, Gabriel SB, Purcell S, Daly MJ, Altshuler D. 2008. Integrated genotype calling and association analysis of SNPs, common copy number polymorphisms and rare CNVs. *Nature Genetics*, 40:1253-1260.

8. Liu JZ, MaRae AF, Nyholt DR, Medland SE, Wray NR, Brown KM, AMFS Investigators, Hayward NK, Montgomery GW, Visscher PM, Martin NG, Macgregor S. 2010. A versatile gene-based test for genome-wide association studies. *American Journal of Human Genetics*, 87:139-145.

9. Ma L, Hoffman G, Keinan A. 2015. X-inactivation informs variance-based testing for X-linked association of a quantitative trait. *BMC Genomics*, 16:241.

10. Ma L, Clark AG, Keinan A. 2013. Gene-Based Testing of Interactions in Association Studies of Quantitative Traits. *PLoS Genetics*, 9(2):e1003321.

11. Price AL, Patterson NJ, Plenge RM, Weinblatt ME, Shadick NA, Reich D. 2006. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38:904-909.

12. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MA, Bender D, Maller J, Sklar P, de Bakker PI, Daly MJ, Sham PC. 2007. PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses. *American Journal of Human Genetics*, 81:559-575.

13. Willer CJ, Li Y, Abecasis GR. 2010. METAL : fast and efficient meta-analysis of genomewide association scans. *Bioinfomatics*, 26:2190-2191

14. Zaykin DV, Zhivotovsky LA, Westfall PH, Weir BS. 2002. Truncated product method for combining P-values. *Genetic Epidemiology*, 22:170-185.