JASPER Software Documentation

Version 1.0
December 14, 2023

Joelle Mbatchou[1] and Mary Sara McPeek[1,2]

Department of Statistics[1] and Human Genetics[2]
The University of Chicago, Chicago IL 60637, USA.

JASPER
A C/C++ program to assess significance for a general class of test statistics in genetic association analyses with structured samples.

Homepage: http://www.stat.uchicago.edu/~mcpeek/software/index.html
Release 1.0

==========

License

==========

This program includes code provided by others under licenses compatible with GNU GPL as free software:
Eigen, which is under Mozilla Public License (version 2.0)
Numerical recipes utility functions, which are public domain
GNU Scientific Library, which is under GNU General Public License (version 3)
Hash table library by Attractive Cahos, which is under an Open Source MIT License

==========

We request that use of this software be cited in publications as follows:

Mbatchou, J. and McPeek, M. S. JASPER: fast, powerful, multitrait association testing in structured samples gives insight on pleiotropy in gene expression. *Manuscript in preparation*, 2023

==========

Contents

1 Overview of JASPER **4**

2 Installing JASPER **4**
- 2.1 Instructions . . . 4
- 2.2 Compilation Prerequisites . . . 4
- 2.3 Compiling the JASPER binary . . . 5

3 Running JASPER **5**

4 Input **6**
- 4.1 Phenotype data file (specified by flag `--pheno`) . . . 6
- 4.2 Genotype data file (specified by flag `--geno`) . . . 6
- 4.3 Ancestry-informative covariates file (specified by flag `--covG`) . . . 7
- 4.4 GRM file (specified by flag `--grm` when `--eigen` is not used) . . . 8
- 4.5 Optional eigendecomposition file (specified by flag `--eigen`) . . . 8

5 Output **9**

6 Examples **9**

7 Bug reports and feedback **10**

8 Acknowledgements **10**

1 Overview of JASPER

JASPER is a C/C++ program that assesses significance for a reasonably broad class of association tests that involve two matrices. JASPER (for "Joint Association analysis in Structured samples based on approximating a PERmutation distribution") a fast, powerful, robust method for assessing significance of multi-trait association with a set of genetic variants, in samples that have population sub-structure, admixture and/or relatedness. It allows for covariates, ascertainment and rare variants and is robust to phenotype model misspecification.

The main features of JASPER are:

- Applicable to a wide range of association test statistics, including kernel-based ("KAT"-type) tests with essentially arbitrary phenotype kernels
- Well-suited to handle high-dimensional traits.
- Allows for covariates in the model
- Adjusts for population structure, cryptic relatedness and/or related individuals
- Computationally efficient

2 Installing JASPER

2.1 Instructions

1. Download the JASPER package. This package contains the documentation, source code, example files, and the GNU GPL license.
2. Read the entire documentation (this document) carefully to understand the purpose of this program and how it works.
3. Decompress the archive with GNU software `gzip`: `tar xvfz JASPER_v1.0.tar.gz`
4. Switch to the newly created directory: `cd JASPER`
5. This directory contains the GNU GPL license in file `gpl.txt` and four subdirectories:
   - `bin` is where the compiled binary executable file will be generated;
   - `src` contains the source code;
   - `doc` contains this document `JASPER_v1.0_doc.pdf`;
   - `examples` contains example input and output files.

2.2 Compilation Prerequisites

You will need GCC including the standard C++ and Fortran libraries. If GCC is not available on your system, it can be obtained from: https://gcc.gnu.org. You will also need to install the Boost C++ libraries (https://www.boost.org/).

2.3 Compiling the JASPER binary

1. Set the variable `BOOST_LIB_PATH` in the Makefile to point to the installation folder for the installed boost library
2. Type `make`

. This will build an executable program in the `bin/` directory called JASPER.

Note: alternatively, you can specify the `BOOST_LIB_PATH` *variable directly in the* `make` *command. For example:* `BOOST_LIB_PATH=/opt/lib/boost/1.83.0/ make`

3 Running JASPER

1. To run the executable program, first, prepare the input files (see Section 4). Then JASPER can be run from the command line via the command `./bin/JASPER` with the desired options. For example, the command might look like:

   ```
   ./JASPER -p phenofile -e eigfile -n 1000 -o prefix
   ```

   ```
   ./JASPER -p phenofile -r GRMfile -s 12345
   ```

   We briefly summarize the usage of the available command line options below. Note that they are case-sensitive.

   - **--pheno phenofile**: Allows the user to specify the name of the phenotype data file.
   - **--geno genofile**: Allows the user to specify the name of the genotype data file.
   - **--covG covarfile**: Allows the user to specify the name of the ancestry-informative covariates file. (optional)
   - **--grm GRMfile**: When an eigendecomposition file is not available, the program offers the option to make use of a known genetic relationship matrix. The flag `-r` indicates the availability of the genetic relationship matrix and instructs the program to read in the matrix from the text file under the name `GRMfile`. When the `--eigen` option is used, the `--grm` option will be ignored.
   - **--eigen eigfile**: Allows the user to specify the name of the file containing the eigendecomposition results for the genetic relationship matrix, if available. (optional)
   - **--out prefix**: Allows the user to specify the prefix string added to the default output filenames.
   - **-s**: This option can be used in conjunction with `--grm`, to instruct the program to output the eigendecomposition results of the genetic relationship matrix. The name of the output file will be `prefix_eig`, where `prefix` is specified by the `--out` option.
   - **--mc 1000**: This option specifies the number of Monte Carlo iterations to perform if the Pearson approximation need for JASPER fails. If unused, no Monte Carlo iterations are performed. (optional)

4 Input

4.1 Phenotype data file (specified by flag `--pheno`)

The phenotype data file contains data on the 1+ phenotypes. This file should have the format of a plink PED file. The columns in the file are:

```
family ID  (numeric or alphanumeric)
individual ID (numeric or alphanumeric)
father ID (numeric or alphanumeric)
mother ID (numeric or alphanumeric)
sex (1/2)
phenotype  #1
phenotype  #2
     ⋮
```

The requirements are the following:

- Tab or space delimited. No header row.
- The individual ID is assumed to be unique across individuals. Numeric and alphanumeric IDs are allowed.
- Information on family ID, father ID, mother ID and sex is not used by the program. So one could simply have arbitrary values for them.
- Phenotypes should be numerical. No missing phenotype is allowed. Phenotypes could be residuals from a mixed model.

Example
A file with 4 individuals, and three phenotypes might look like:

```
FAM1 IND1    0 0 1 0 2.54  47
FAM1 SAMP345 0 0 2 1 -2.88 25
FAM1 3       0 0 2 1 4.37  29
FAM2 SUB4    0 0 1 0 -4.35 37
```

4.2 Genotype data file (specified by flag `--geno`)

The genotype data file contains data on the genotypes across the phenotyped individuals. The columns in the file correspond to the individuals and the rows to the variants:

```
variant ID
individual 1 ID
individual 2 ID
     ⋮
```

The requirements are the following:

- Tab or space delimited.

- The header row must contain the individual IDs (first column can have any value [e.g. ID]).
- The order of individuals must match that of the phenotype file. If that is not the case, the program will report an error.
- Genotype should be in [0,2]. No missing genotype is allowed (if missingness is present, impute genotypes prior to running JASPER, e.g. with average).

Example
A file with 3 variants across 4 individuals might look like:

```
ID IND1 SAMP345 3 SUB4
rs12 1 0 0 1
rs98 0 0 0 1
rs45 0 1 0 1
```

4.3 Ancestry-informative covariates file (specified by flag `--covG`)

The covariate file contains data on ancestry-informative covariates across the phenotyped individuals. The columns in the file are:

```
family ID
individual ID
covariate  #1
covariate  #2
     ⋮
```

The requirements are the following:

- Tab or space delimited. No header row.
- The order of individuals must match that of the phenotype file. If that is not the case, the program will report an error.
- Intercept should NOT be included in this file as a covariate. The program will automatically add an intercept column.

Example
A file with 4 individuals, and 2 covariates might look like:

```
FAM1  IND1    2.54  47
FAM1  SAMP345 -2.88 25
FAM10 3       4.37  29
FAM11 SUB4    -4.35 37
```

4.4 GRM file (specified by flag `--grm` when `--eigen` is not used)

When the flag `--eigen` is not used, the program offers the option to perform eigendecomposition on a known genetic relationship matrix. The GRM file is a text file listing one entry of the matrix per row. The three columns are

```
individual_1 ID
individual_2 ID
entry_in_matrix
```

The requirements are:

- The order of the rows does not matter.
- Individuals IDs should agree with those in the phenotype file.
- Any pair of individuals should be included in the file at most once, regardless of the order they appear. If a pair is included more than once with different values, the program will produce a warning and will discard the new value.
- When `indiv1` and `indiv2` are the same, the row in the file corresponds to a diagonal element in the matrix.
- If a pair is not found in the file, 0 will be used for the corresponding entry in the matrix. If it corresponds to a diagonal entry, 1 will be used for that entry in the matrix.
- If the resulting matrix is not positive semi-definite, the program will return an error.
- For a sample with individuals `IND1`, `SAMP345`, and `3`, the GRM file may look like:

  ```
  3 3 1.01
  SAMP345 SAMP345 0.98
  IND1 SAMP345 0.02
  SAMP345 3 0.06
  ```

  So that the corresponding genetic relationship matrix is:

  ```
  1.00 0.02 0.00
  0.02 0.98 0.06
  0.00 0.06 1.01
  ```

4.5 Optional eigendecomposition file (specified by flag `--eigen`)

This file is used to improve computational efficiency by making use of existing eigendecomposition results of the genetic relationship matrix. It is a binary file that encodes the eigenvalues and eigenvectors of the genetic relationship matrix in `double` (double precision floating-point type). Let $\mathbf{\Phi}$ be a $n \times n$ genetic relationship matrix, and $\mathbf{\Phi} = VDV^{-1}$ be an eigendecomposition of $\mathbf{\Phi}$, where D is a diagonal matrix containing the eigenvalues and V is an orthogonal matrix containing the corresponding eigenvectors. The eigendecomposition input file should encode the diagonal elements of D followed by a row-wise list of the elements of V in binary format as double precision floating-point numbers.

- The genetic relationship matrix $\mathbf{\Phi}$ underlying this file should correspond exactly to the set and ordering of individuals in the phenotype data file. In particular, the dimension $\mathbf{\Phi}$ should equal the number of individuals in the phenotype data file. Mismatched dimension will likely result in a segmentation fault. While mismatched ordering may not result in an error, it will lead to incorrect results.

Remarks: In the current version of the program,

- A user-specified filename should not exceed 2000 characters in length. Otherwise, an error will be reported. To increase this maximum length, one may open the file `jasper.h` in the directory `src`, locate the line starting with "`#define MAXFILELEN 2001`", replace "`2001`" with the number which equals the desired maximum length plus 1, and re-compile the program as instructed in Section 2.
- The number of Monte Carlo iterations cannot be greater than 10^6 , otherwise an error will be reported. To increase this amount, one may open the file `jasper.h` in the directory `src`, locate the line starting with "`#define NITER_MAX 1e6`", replace "`1e6`" with the desired maximum amount, and re-compile the program as instructed in Section 2.

5 Output

The program will output up to two files: a text file that contains the p-value result, and an optional binary file that records the eigenvalues and the eigenvectors of the genetic relationship matrix. Assuming `prefix` has been specified to `--out`, the filenames are `prefix_out.txt`, and `prefix_eig`. Below we explain each of the output files.

1. **prefix_out.txt** is a text file containing the p-value testing for association between the variants with the phenotypes specified:

   ```
   pvalue 0.91212602132517717
   ```

2. **prefix_eigen** is a binary file that stores the eigenvalues and eigenvectors of the genetic relationship matrix, as computed by the JASPER program. It will be part of the output only when both `--grm` and `-s` are used. The formatting of this output file is the same as that of the binary eigen-decomposition input file described in Section 4.5.

6 Examples

The directory `JASPER/examples` provides example input files: `pheno_ex, geno_ex`, `ai_cov_ex`, and `grm_ex`. Below, we list several example commands that can be run on these files.

1. `./bin/JASPER --pheno examples/pheno_ex --geno examples/geno_ex --grm examples/grm_ex --out run1`

   This command instructs the program to perform the `JASPER` method, testing for association between the phenotypes in `pheno_ex` with the genotypes in `geno_ex`. The relationship matrix is specified from `grm_ex`. The `--out` option specifies the prefix of the output files to be `run1`. The program will generate a single output file: `run1_out.txt`.

2. `./bin/JASPER --pheno examples/pheno_ex --geno examples/geno_ex --grm examples/grm_ex --out run2 -s`

   With this command, the program will first read the genetic relationship matrix from `grm_ex` and compute its eigendecomposition, whose result will be written to file `run2_eigen`.

7 Bug reports and feedback

We appreciate comments and suggestions and if you do encounter a bug in the JASPER software please send us a message. Please include in your message the program version (printed out when the program is run), platform (windows, mac, linux, etc.), description of your problem, and if possible example files (in a zip folder) that caused the problem.

8 Acknowledgements

1. Numerical Recipes in C. We use the utility functions in nrutil.h Eigen. We use the package for some of the matrix computations.

2. LAPACK. We use the `dsyevr` routine and its dependencies as one of the options for performing eigen-decomposition.

3. khash.h a fast and light-weighted hash table library in C.

4. Valgrind and gdb, for software debugging and profiling.

References

[1] Mbatchou, J. and McPeek, M. S. JASPER: fast, powerful, multitrait association testing in structured samples gives insight on pleiotropy in gene expression. *Manuscript in preparation*, 2023.