BRASS Software Documentation

Version 1.0
April 13, 2019

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

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

BRASS
A C/C++ program for performing a permutation-based test with a binary response in structured samples.

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

==========

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
LAPACK, which is under the modified BSD license available at http://www.netlib.org/lapack/LICENSE.txt
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
CARAT, which is under GNU General Public License (version 3)

==========

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

Mbatchou, J., Abney, M., and McPeek, M. S. Permutation methods for assessing significance in binary trait association mapping with structured samples. *bioRxiv*, page 451377, 2019. doi: 10.1101/451377

==========

Contents

1 Overview of BRASS

BRASS is a C/C++ program that generates permutation-based replicates of a binary trait in correlated samples. It implements the BRASS (for "Binary trait Resampling method Adjusting for Sample Structure") method, a novel approach to generating replicates of a binary trait in the presence of population structure and/or cryptic relatedness.
It is an extension of the MVNpermute resampling method of Abney, Ober and McPeek from quantitative to binary traits [1]. BRASS is based on a quasi-likelihood framework that models covariate effects on a logit scale and accounts for the dependence of the variance on the mean, a natural feature of binary traits. The correlation present in the sample is incorporated in the variance structure as additive polygenic variance. A linear map is applied to the residuals to obtain a vector that is approximately second-order exchangeable (i.e. uncorrelated entries with same mean and variance), and permutation is then applied. The permuted vector is then mapped back to the original space to obtain a permutation-based replicate. The resampled data generated from BRASS approximately preserves the estimated mean and variance structure of the original data.
The trait replicates generated are quantitative and can be used with any association test that allows for either quantitative or binary traits (e.g. score tests). BRASS is applicable to general combinations of unrelated and related individuals, and if the latter are present, does not require knowledge of the pedigree structure. It allows for the option of fitting the trait model under the null of no association without generating trait replicates in order to formulate the null model for preliminary analyses of the trait/covariate data.
The main features of BRASS are:

- Allows for covariates in the phenotypic null model
- Incorporates the binary nature of the response
- Adjusts for population structure, cryptic relatedness and/or related individuals
- Computationally efficient

2 Installing BRASS

2.1 Instructions

1. Download the BRASS package. This package contains the documentation, source code, pre-compiled binaries, 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 BRASS_v1.0.tar.gz`
4. Switch to the newly created directory: `cd BRASS`
5. This directory contains the GNU GPL license in file `gpl.txt` and four subdirectories:
    - `bin` contains the pre-compiled binary executable file (x86 64bit Linux);
    - `src` contains the source code;

- `doc` contains this document `BRASS_v1.0_doc.pdf`;
- `examples` contains example input and output files.

6. To test whether the pre-compiled binary executable works on your system, you can switch to the `examples` directory and run the following command (see README.txt for more details):

   `../bin/BRASS -p pheno_ex -e eigen_ex -n 0`

   In some cases, you may need to run `chmod u+x ../bin/BRASS` prior to using the binary executable.

7. If the pre-compiled binary does not work on your system, make sure the compilation prerequisite explained in Section 2.2 is met and switch to the `src` directory. Type `make`. This will build an executable program called BRASS.

2.2 Compilation Prerequisites

To compile BRASS on your own machine, 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.

3 Running BRASS

1. To run the executable program (see Section 2), first, prepare the input files (see Section 4). Then BRASS can be run from the command line via the command `./BRASS` with the corresponding command line options. For example, the command might look like:

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

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

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

   - **-p phenofile**: Allows the user to specify the name of the phenotype data file. The filename defaults to `pheno.txt`. To specify another filename, replace `pheno.txt` with the appropriate filename.
   - **-e eigfile**: Allows the user to specify the name of the file containing the eigendecomposition results for the genetic relationship matrix, if available. There is no default filename.
   - **-r 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`. There is no default filename. When the `-e` option is used, the `-r` option will be ignored.
   - **-o prefix**: Allows the user to specify the prefix string added to the default output filenames. If skipped, the output files will be `BRASS_param.txt` and optionally `BRASS_perms.txt`. For example, if `-o PREFIX` is used, the filenames will be `PREFIX_param.txt` and `PREFIX_perms.txt`.

- **-w**: This option can be used in conjunction with `-r`, 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 `-o` option. If `-o` is not used, the filename defaults to `BRASS_eig`.
- **-n 1000**: Allows the user to specify the number of trait replicates to generate. If skipped, the default of 1,000 replicates will be generated. To specify another value, replace "`1000`" by the desired value.
- **-s 12345**: Allows the user to specify the random seed used for sampling the trait replicates. If skipped, a randomly selected value will be generated based on the execution time and will be printed in the output. To specify another value, replace "`12345`" by the desired value.

4 Input

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

The phenotype data file contains data on the the binary phenotype and any covariates. This file should have the format of a plink PED file. The columns in the file are:

```
1 (this is the family ID)
individual ID (numeric or alphanumeric)
father ID (numeric or alphanumeric)
mother ID (numeric or alphanumeric)
sex (1/2)
phenotype (0/1)
covariate1 (numeric)
covariate2 (numeric)
    ⋮
```

The requirements are the following:

- Tab or space delimited.
- The family ID must be 1 for all individuals. If that is not the case, the program will report an error.
- The individual ID is assumed to be unique across individuals. Numeric and alphanumeric IDs are allowed.
- Information on father ID, mother ID and sex is not used by the program. So one could simply have arbitrary values (numeric/alphanumeric for father and mother IDs, integer for sex) for them.
- Phenotype should be coded as either 0 or 1. No missing phenotype is allowed. Non-integer values or integers other than 0 and 1 will result in an error.
- Intercept should NOT be included in this file as a covariate. When the program fits the null model, it will automatically add an intercept to the phenotype model. If an intercept is included in the file, the program will report an error.

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

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

4.2 Optional eigendecomposition file (specified by flag `-e`)

This file (optional) allows the program 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.

4.3 Optional GRM file (specified by flag `-r` when `-e` is not used)

When the flag `-e` 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 columns are

```
1 indiv1 indiv2 entry_in_matrix
```

The requirements are:

- The order of the rows does not matter.
- The first column in the file must be 1 (corresponds to family ID).
- 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 ignore the information given the first time it appears.
- 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, zero will be used for the corresponding entry in the matrix. If it corresponds to a diagonal entry, one 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:

```
1 IND1 IND1 1.00
1 3 3 1.01
1 SAMP345 SAMP345 0.98
1 IND1 SAMP345 0.02
1 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
```

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 `qlbin.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 trait replicates to simulated cannot be above 10^6 , otherwise an error will be reported. To increase this amount, one may open the file `qlbin.h` in the directory `src`, locate the line starting with "`#define NPERM_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 three files: a text file that contains the parameter estimation results, another text file that contains the trait replicates simulated under the null, and an optional binary file that records the eigenvalues and the eigenvectors of the genetic relationship matrix. The default filenames are `BRASS_param.txt`, `BRASS_perms.txt`, and `BRASS_eig`, where the user may choose to replace "`BRASS`" by another prefix using the `-o` flag. The file containing the simulated trait replicates will not be part of the output if flag "`-n 0`" is used. Below we explain each of the output files (with the default filenames.)

1. **BRASS_perms.txt** is a text file containing the phenotypic replicates. The first line lists the subject IDs in the order they are included in the phenotype data file. Each subsequent row corresponds to a simulated trait replicate.
2. **BRASS_param.txt** is a text file containing variance component and covariate parameter estimates (and standard errors) under the null hypothesis of no association. With 2 covariates, the file might look like:

```
Parameter              Null_Estimate SE
Variance_Parameter_xi  0.001123      NA
Intercept              0.345134      0.541001
Covariate_1            1.340991      0.320202
Covariate_2            0.049667      0.100312
```

3. **BRASS_eig** is a binary file that stores the eigenvalues and eigenvectors of the genetic relationship matrix, as computed by the BRASS program. It will be part of the output only when both `-r` and `-w` are used. The formatting of this output file is the same as that of the binary eigendecomposition input file described in Section 4.2.

6 Examples

The directory `BRASS/examples` provides example input files: `pheno_ex`, `eigen_ex`, and `GRM_ex`. Below, we list several example commands that can be run on these files. Make sure that the example input files are in the same directory as the binary executable `BRASS`.

1. `./BRASS -p pheno_ex -e eigen_ex -o Yout -s 123`

   This command instructs the program to perform the `BRASS` method using the phenotype and covariate information in `pheno_ex` as well as the known eigendecomposition results given in `eigen_ex`. The default of 1,000 trait replicates will be simulated using as random seed 123. The `-o` option specifies the prefix of the output files to be `Yout`. The program will generate two output files: `Yout_param.txt` and `Yout_perms.txt`.

2. `./BRASS -p pheno_ex -r GRM_ex -w`

   With this command, the program will first read the genetic relationship matrix from `GRM_ex` and compute the eigendecomposition, whose result will be stored in an output file with the default filename (as `-o` is not used) `BRASS_eig`. There will be two additional output files with default filenames: `BRASS_param.txt` and `BRASS_perms.txt`.

3. `./BRASS -p pheno_ex -e eigen_ex -n 0`

   This command instructs the program to fit the null model without simulating trait replicates. The program will generate one file: `BRASS_param.txt`, which stores the parameter estimation results.

7 Technical Details

7.1 Model

Let $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ denote the $n \times 1$ phenotype vector, where Y_i is the binary trait value for the i-th individual. Let $\mathbf{X}$ denote the $n \times k$ covariate matrix (including an intercept term) and let $\mathbf{G} = (G_1, \ldots, G_n)^T$ denote the marker of interest on which association testing with the trait is performed, where G_i represents the minor allele count (i.e. coded as 0,1 or 2) for the i-th individual.

We assume the following quasi-likelihood model,

$$\boldsymbol{\mu} := \mathrm{E}(\mathbf{Y}|\mathbf{X},\mathbf{G}) = \mathrm{logit}^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{G}\gamma), \text{ and} \tag{1}$$

$$\boldsymbol{\Omega} := \mathrm{Var}(\mathbf{Y}|\mathbf{X},\mathbf{G}) = \boldsymbol{\Gamma}^{1/2}\,\boldsymbol{\Sigma}\,\boldsymbol{\Gamma}^{1/2}. \tag{2}$$

where in (1) $\boldsymbol{\beta}$ is a $k \times 1$ vector representing the unknown effects of covariates on the phenotypic mean, and γ represents the unknown effect of the genetic marker of interest; in (2), $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}(\boldsymbol{\mu}) = \mathrm{diag}\{\mu_1(1-\mu_1), \ldots, \mu_n(1-\mu_n)\}, \boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\xi) = \xi\boldsymbol{\Phi} + (1-\xi)\mathbf{I}$, where the unknown scalar parameter $\xi \in [0, 1]$ allows for the inclusion of additional correlation though the elements of $\boldsymbol{\Phi}$. In practice, an empirical genetic relationship matrix (GRM) from genome-wide data can be used as an estimate for $\boldsymbol{\Phi}$ to capture the genetic similarity between individuals and adjust for the effects of potential population structure and/or cryptic relatedness.

7.2 Fitting the null model

Trait replicates are generated under the null hypothesis of no association ($H_0 : \gamma = 0$). Under this constraint, the null estimate $(\widehat{\boldsymbol{\beta}}, \hat{\xi})$ is obtained by iteratively solving the following system of estimating equations [2],

$$U(\boldsymbol{\beta}) := \boldsymbol{D}_{\boldsymbol{\beta}}^T \boldsymbol{\Omega}^{-1}(\mathbf{Y} - \boldsymbol{\mu}) = \mathbf{0}, \tag{3}$$

$$(\mathbf{Y} - \boldsymbol{\mu})^T \boldsymbol{\Gamma}^{-1/2}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\Phi} - \mathbf{I})\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}^{-1/2}(\mathbf{Y} - \boldsymbol{\mu}) = \mathrm{trace}(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\Phi} - \mathbf{I})), \tag{4}$$

where $\boldsymbol{D}_{\boldsymbol{\beta}} = \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\beta}}$. In our model with a logit link function, $\boldsymbol{D}_{\boldsymbol{\beta}} = \boldsymbol{\Gamma}\mathbf{X}$.

7.3 The BRASS resampling method

We first consider ξ fixed. Assuming that H_0 is true, let $\boldsymbol{\beta}_0$ denote the true value of $\boldsymbol{\beta}$, and let $\boldsymbol{\mu}_0, \boldsymbol{\Gamma}_0$, $\mathbf{D}_0$ and $\boldsymbol{\Omega}_0$ correspond to $\boldsymbol{\mu}, \boldsymbol{\Gamma}, \mathbf{D}_\beta$ and $\boldsymbol{\Omega}$, respectively, evaluated at $\boldsymbol{\beta}_0$ and $\gamma = 0$. Similarly, let $\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Gamma}}$, $\widehat{\mathbf{D}}$ and $\widehat{\boldsymbol{\Omega}}$ be the same quantities evaluated at $\widehat{\boldsymbol{\beta}}$ and $\gamma = 0$.
By applying a Taylor series expansion to $U(\boldsymbol{\beta})$ and $\boldsymbol{\mu}$ around $\boldsymbol{\beta}_0$ and evaluated at $\widehat{\boldsymbol{\beta}}$, we can express the residual vector as,

$$\mathbf{Y} - \widehat{\boldsymbol{\mu}} \approx \left[\mathbf{I} - \mathbf{D}_0\,(\mathbf{D}_0^T\,\boldsymbol{\Omega}_0^{-1}\,\mathbf{D}_0)^{-1}\mathbf{D}_0^T\,\boldsymbol{\Omega}_0^{-1}\right]\,(\mathbf{Y} - \boldsymbol{\mu}_0). \tag{5}$$

We use a factorization $\mathbf{C}_0$ of $\boldsymbol{\Omega}_0$, with $\boldsymbol{\Omega}_0 = \mathbf{C}_0^T\mathbf{C}_0$, to remove the correlation due to $\boldsymbol{\Phi}$ and obtain,

$$\mathrm{Var}[\mathbf{C}_0^{-T}(\mathbf{Y} - \widehat{\boldsymbol{\mu}})] \approx \mathbf{I} - \mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T = \boldsymbol{\Psi}_0, \tag{6}$$

where $\mathbf{W} = \mathbf{C}_0^{-T}\mathbf{D}_0$. The matrix $\boldsymbol{\Psi}_0$ in (6) is symmetric and idempotent and thus can be expressed as $\boldsymbol{\Psi}_0 = \mathbf{V}\mathbf{V}^T$, where the columns of $\mathbf{V}$ contain the eigenvectors of $\boldsymbol{\Psi}_0$ corresponding to the eigenvalue 1 with $\mathbf{V}^T\mathbf{V} = \mathbf{I}_{n-k}$ (assuming $\mathbf{X}$ is full rank).
The linear transformation $\hat{\boldsymbol{\zeta}} = \mathbf{V}^T\mathbf{C}_0^{-T}(\mathbf{Y} - \widehat{\boldsymbol{\mu}})$ has for covariance matrix,

$$\mathrm{Var}(\hat{\boldsymbol{\zeta}}) \approx \mathbf{V}^T\boldsymbol{\Psi}_0\mathbf{V} = \mathbf{I}_{n-k}. \tag{7}$$

Hence, we obtain a transformation of the residuals with approximately uncorrelated entries, same mean and same variance. With $\boldsymbol{\Pi}$ denoting a random permutation matrix, we generate a new trait replicate $\mathbf{Y}_\pi$ as,

$$\mathbf{Y}_\pi = \widehat{\boldsymbol{\mu}} + \mathbf{C}_0^T\mathbf{V}\boldsymbol{\Pi}\mathbf{V}^T\mathbf{C}_0^{-T}(\mathbf{Y} - \widehat{\boldsymbol{\mu}}) \tag{8}$$

The derivation in (8) relies on ξ being known in addition to the vector $\boldsymbol{\beta}_0$. In practice, these are both estimated from the data; hence, we evaluate this equation at $(\widehat{\boldsymbol{\beta}}, \hat{\xi})$ and replace $\mathbf{C}_0$ and $\mathbf{V}$ by their estimated quantities $\widehat{\mathbf{C}}$ and $\widehat{\mathbf{V}}$, respectively. A trait replicate is obtained as,

$$\mathbf{Y}_\pi = \widehat{\boldsymbol{\mu}} + \widehat{\mathbf{C}}^T \widehat{\mathbf{V}} \boldsymbol{\Pi} \widehat{\mathbf{V}}^T \widehat{\mathbf{C}}^{-T} (\mathbf{Y} - \widehat{\boldsymbol{\mu}}) \tag{9}$$

7.4 Obtaining $\widehat{\mathbf{C}}$

Instead of computing the cholesky decomposition of $\widehat{\boldsymbol{\Omega}}$, which involves a time complexity of $O(n^3)$, we use the the eigendecomposition of $\boldsymbol{\Phi}$, which is computed when fitting the null model in Section 7.2. This involves an orthogonal $n \times n$ matrix $\mathbf{U}$ and a diagonal $n \times n$ matrix $\boldsymbol{\Lambda}$ with $\boldsymbol{\Phi} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$. The columns of $\mathbf{U}$ contain the eigenvectors of $\boldsymbol{\Phi}$, and the diagonal elements of $\boldsymbol{\Lambda}$ contain the eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\boldsymbol{\Phi}$. Hence, we may write:

$$\begin{aligned}
\widehat{\boldsymbol{\Omega}} &= \widehat{\boldsymbol{\Gamma}}^{1/2} \widehat{\boldsymbol{\Sigma}} \widehat{\boldsymbol{\Gamma}}^{1/2} \\
&= \widehat{\boldsymbol{\Gamma}}^{1/2} [\hat{\xi}\, \boldsymbol{\Phi} + (1 - \hat{\xi})\, \mathbf{I}] \widehat{\boldsymbol{\Gamma}}^{1/2} \\
&= \widehat{\boldsymbol{\Gamma}}^{1/2} [\hat{\xi}\, \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T + (1 - \hat{\xi})\, \mathbf{U}\mathbf{U}^T] \widehat{\boldsymbol{\Gamma}}^{1/2} \\
&= \widehat{\boldsymbol{\Gamma}}^{1/2} \mathbf{U} [\hat{\xi}\, \boldsymbol{\Lambda} + (1 - \hat{\xi})\, \mathbf{I}] \mathbf{U}^T \widehat{\boldsymbol{\Gamma}}^{1/2} \\
&= \widehat{\boldsymbol{\Gamma}}^{1/2} \mathbf{U} \mathbf{Z}^{1/2} \mathbf{Z}^{1/2} \mathbf{U}^T \widehat{\boldsymbol{\Gamma}}^{1/2}
\end{aligned}$$

where $\mathbf{Z}$ is a diagonal matrix with elements $(\hat{\xi}\lambda_1 + 1 - \hat{\xi}, \ldots, \hat{\xi}\lambda_n + 1 - \hat{\xi})$ in its diagonal. Therefore, we use $\widehat{\mathbf{C}} = \mathbf{Z}^{1/2} \mathbf{U}^T \widehat{\boldsymbol{\Gamma}}^{1/2}$ as a factorization of $\widehat{\boldsymbol{\Omega}}$.

7.5 Obtaining $\widehat{\mathbf{V}}$

Directly computing $\widehat{\Psi}$, which corresponds to $\boldsymbol{\Psi}_0$ in (6) evaluated at $(\widehat{\boldsymbol{\beta}}, \hat{\xi})$, and obtaining its eigendecomposition to get $\widehat{\mathbf{V}}$ would involve a time complexity of $O(n^3)$. Instead, assuming the number of covariates k is smaller than the sample size n, we consider the singular value decomposition (SVD) of $\widehat{\mathbf{W}} = \widehat{\mathbf{C}}^{-T} \widehat{\mathbf{D}} \in \mathbb{R}_{n \times k}$, and use it to obtain the eigenvectors of $\widehat{\Psi}$ that are in $\widehat{\mathbf{V}}$.
More precisely, we have $\widehat{\mathbf{W}} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ as the SVD of $\widehat{\mathbf{W}}$ with the columns of the orthogonal matrices $\mathbf{U} \in \mathbb{R}_{n \times n}$ and $\mathbf{V} \in \mathbb{R}_{k \times k}$ containing the left and right singular vectors of $\widehat{\mathbf{W}}$, respectively, and $\mathbf{S} \in \mathbb{R}_{n \times k}$ containing in its diagonal elements the singular values of $\widehat{\mathbf{W}}$. Hence, (6) can be rewritten as,

$$\begin{aligned}
\widehat{\Psi} &= \mathbf{I} - \mathbf{U}\mathbf{S}\mathbf{V}^T (\mathbf{V}\mathbf{S}^T\mathbf{U}^T \cdot \mathbf{U}\mathbf{S}\mathbf{V}^T)^{-1} \mathbf{V}\mathbf{S}^T\mathbf{U}^T \\
&= \mathbf{I} - \mathbf{U}\mathbf{S}\mathbf{V}^T (\mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T)^{-1} \mathbf{V}\mathbf{S}^T\mathbf{U}^T \\
&= \mathbf{I} - \mathbf{U}\mathbf{S}\mathbf{V}^T \mathbf{V} (\mathbf{S}^T\mathbf{S})^{-1} \mathbf{V}^T \mathbf{V}\mathbf{S}^T\mathbf{U}^T \\
&= \mathbf{I} - \mathbf{U}\mathbf{S} (\mathbf{S}^T\mathbf{S})^{-1} \mathbf{S}^T\mathbf{U}^T \\
&= \mathbf{U}\mathbf{U}^T - \mathbf{U}\mathbf{S} (\mathbf{S}^T\mathbf{S})^{-1} \mathbf{S}^T\mathbf{U}^T \\
&= \mathbf{U} [\mathbf{I} - \mathbf{S} (\mathbf{S}^T\mathbf{S})^{-1} \mathbf{S}^T] \mathbf{U}^T
\end{aligned}$$

The matrix $[\mathbf{I} - \mathbf{S}(\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T]$ is a $n \times n$ diagonal matrix, and the diagonal entries are $(0, \ldots, 0, 1, \ldots, 1)$ with the first k entries being 0 and the last $(n - k)$ entries being 1. Hence, we can obtain $\widehat{\mathbf{V}}$ by extracting the last $(n - k)$ columns of $\mathbf{U}$. This reduces the time complexity from $O(n^3)$ to $O(nk^2)$.

8 Bug reports and feedback

We appreciate comments and suggestions and if you do encounter a bug in the BRASS 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.

9 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. CARAT. Our implementation of parameter estimation is based on code from CARAT.

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

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

References

[1] Abney, M., Ober, C., and McPeek, M. S. Quantitative-Trait Homozygosity and Association Mapping and Empirical Genomewide Significance in Large, Complex Pedigrees: Fasting Serum-Insulin Level in the Hutterites. *The American Journal of Human Genetics*, 70(4):920–934, 2002.

[2] Jiang, D., Zhong, S., and McPeek, M. S. Retrospective Binary-Trait Association Test Elucidates Genetic Architecture of Crohn Disease. *American Journal of Human Genetics*, 98(2):243–255, 2016.

[3] Mbatchou, J., Abney, M., and McPeek, M. S. Permutation methods for assessing significance in binary trait association mapping with structured samples. *bioRxiv*, page 451377, 2019. doi: 10.1101/451377.