CARAT Software Documentation

Version 1.3
July 23, 2020

Duo Jiang[1], Sheng Zhong[2], Mary Sara McPeek[2,3]

Department of Statistics[1]
Oregon State University, Corvallis, OR 97330, USA

Departments of Statistics[2] and Human Genetics[3]
The University of Chicago, Chicago, IL 60637, USA

CARAT
A C/C++ program for case-control association analysis accounting for populations structure and covariates

Homepage: http://www.stat.uchicago.edu/~mcpeek/software/index.html
Release 1.3 July 23, 2020

License

We request that use of this software be cited in publications as follows:
Jiang D., Zhong S., McPeek M. S. (2016). Retrospective Binary-Trait Association Test Elucidates Genetic Architecture of Crohn Disease. *The American Journal of Human Genetics* 98:243-255

Contents

1 Overview of CARAT

CARAT is a C/C++ program that performs genome-wide association analysis of a binary trait in samples with unknown population structure. It implements the CARAT (Case-control Retrospective Association Test) method, a novel approach to binary-trait association testing, which accounts for relevant covariate information and effectively controls for population structure[1]. The method can achieve substantial power improvement over linear mixed model methods. CARAT is based on a mixed-effects quasi-likelihood framework, which models covariate effects on the logit scale and accounts for the dependence of the variance on the mean. To capture population structure, a genetic relationship matrix calculated from genome-wide SNP data is incorporated in an additive polygenic variance component. CARAT takes an estimating equation and score test approach, which is computationally efficient for large-scale genome-wide studies. When assessing significance of the test statistic, CARAT takes a retrospective approach in which genotypes are viewed as random under the null, conditional on the phenotype and the covariates. This approach is robust to misspecification of the phenotype model. Compared to other association testing methods, CARAT features:

1. consistent power improvement over methods based on linear mixed models
2. the ability to account for covariate information
3. a model that specifically exploits the binary nature of the trait
4. no need for the prevalence of the disease to be known
5. computational efficiency and stability
6. robust control of type I error
7. adjustment for population stratification, admixture and/or cryptic relatedness

The program also allows the option of fitting the phenotypic model to the phenotype and covariate data under the null hypothesis without performing any association test. This can be useful for preliminary analysis of the phenotype and covariate data in order to formulate the null model.

2 Installing CARAT

2.1 Installation Instructions

To install CARAT, proceed with the following steps:

1. Download the CARAT package. The package contains the documentation, source code, example files, binary executable, and the GNU GPL license.
2. Read the documentation (this document) to understand the purpose of this software and how it works. It is highly recommended that users read the paper on the CARAT test [1].
3. Decompress the archive with GNU software `gzip`: `tar xvfz CARAT_v1.0.tar.gz`
4. Change working directory to the one (i.e. CARAT) newly created in Step 3: `cd CARAT`
5. This directory includes the GNU GPL license in file `gpl.txt` and four subdirectories:

- `bin` contains a pre-compiled binary executable file (x86 64bit Linux);
- `src` contains the source code;
- `doc` contains this document `CARAT_v1.3_doc.pdf`;
- `examples` contains example input and output files.

6. If the pre-compiled binary executable works on your system, no installation is necessary. Otherwise, follow Steps 7 and 8 to compile CARAT yourself. To test whether the pre-compiled binary works, you can switch to the `examples` directory and run the following command (see README.txt for more details):

   ```
   ../bin/CARAT -p pheno_ex.txt -e eigfile_ex -n
   ```

   In some cases, you may need to run `chmod a+x ../bin/CARAT` before using the binary executable.

   When optimal computational efficiency is important (for example, for large data sets), it is recommended that the user compile CARAT on his or her own computer system instead of using the pre-compiled executable.

7. If you want to compile CARAT yourself, make sure the compilation prerequisite explained in Section 2.1 is met.

8. Switch back to the `CARAT` directory. Type `make`. This will build an executable program called CARAT.

2.2 Compilation Prerequisite

To compile CARAT on your own machine, you will need GCC including the standard C++ and Fortran libraries. If GCC is not already available on your machine, it is freely available at https://gcc.gnu.org

If the user would like to use the pre-compiled binary executable in the `bin` directory, the prerequisite is not necessary (see Step 6 in Section 2.1).

3 Using CARAT

To run the executable program (see Section 2), first prepare the input files (see Section 4). Then, CARAT can be run from the command line via the command `./CARAT` with all information specified by command line options. The following examples are what the command might look like:

```
./CARAT -p pheno.txt -g geno.txt -e eig
./CARAT -p pheno.txt -g chr14.txt -G genomewide.txt -o prefix
./CARAT -p pheno.txt -R grm.txt -n
```

We briefly summarize the usage of the available options below. Note that the options are case-sensitive. More details about the input file speficications can be found in Section 4.

- **`-p pheno.txt`**: This option allows the user to specify the name of the phenotype input file. This file includes phenotype and covariate information. The filename defaults to `pheno.txt` if this option is not used. To specify another filename, replace `pheno.txt` with the appropriate filename. See Section 4 for format details.

- **-g geno.txt**: This option allows the user to specify the name of the genotype input file, which includes the genotypes of the SNP(s) to be tested for association. The filename defaults to `geno.txt` if this option is not used. To specify another filename, replace `geno.txt` with the appropriate filename. See Section 4 for format details.

- **-o prefix**: This option allows the user to specify the prefix of the output files. If this option is not used, the output files will be `CARAT_out.txt`, `CARAT_param.txt` and optionally `CARAT_eig` and `CARAT_GRM.txt`. If `-o chr1` is used, for example, the filenames will be `chr1_out.txt`, `chr1_param.txt`, `chr1_eig`, and `chr1_GRM.txt`. `chr1` can be replaced with the desired prefix.

- **-e eig_file**: This option allows the user to indicate the availability of a file containing eigen-decomposition results for the genetic relationship matrix and its filename. There is no default filename. This option instructs the program to improve computational efficiency by making use of existing eigen-decomposition results stored in the file under the name `eig_file` (can be replaced by another filename). See Section 4 for format details.

- **-G grm_geno_file**: When an eigen-decomposition file is not available, the program will first obtain the eigen-decomposition results for the genetic relationship matrix to be used in subsequent model fitting and association testing. To do this, the program offers the option to compute the genetic relationship matrix based on the genotypes of a large number of SNPs. The flag `-G` specifies the name of the file including these genotypes, which we will call "the GRM genotype file," to distinguish it from the genotype data file specified by the `-g` option. There is no default filename. The required format of the GRM genotype file is the same as that of the genotype file for the tested SNPs. See Section 4 for details. The two genotypes files may or may not be the same file. When the `-e` option is used, the `-G` option will be ignored. Alternatively, the program can obtain the eigen-decomposition using a known genetic relationship matrix. See flag `-R`.

- **-R grm_file**: When an eigen-decomposition file is not available, the program will first obtain the eigen-decomposition results to be used in subsequent model fitting and association testing. To do this, 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 `grm_file` (can be replaced another filename), which we will call "the GRM file". There is no default filename. See Section 4 for format details. When the `-e` option is used, the `-R` option will be ignored. Alternatively, the program can obtain the genetic relationship matrix based on genotypes of genom-wide SNPs (or a subset of them). See flag `-G`.

- **-r**: This option can be used in conjunction with `-G`, to instruct the program to output the genetic relationship matrix computed from the GRM genotype file. The name of the output file will be `prefix_GRM.txt`, where `prefix` is given by the `-o` option. If `-o` is not used, the filename defaults to `CARAT_GRM.txt`.

- **-n**: This option instructs the program to only fit the phenotypic model under the null hypothesis, without doing association testing. If it is used, no genotype data file will be read.

4 Input

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 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 performs the test, it will automatically add an intercept to the phenotype model. If an intercept is also input as a covariate, the program report an error.

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

```
1 IND1  0 0 2 0 3.9 -0.7
1 3     0 0 2 1 1.2 4.12
1 2     0 0 2 1 0.4 2
1 IND3  0 0 1 0 1.4 -2.0
```

   The optional flag `-p pheno.txt` allows the user to specify the name of the phenotype data file. The filename defaults to `pheno.txt` if this flag is not used. To specify another filename, replace `pheno.txt` with the appropriate filename.

2. **Genotype data file (specified by flag -g)**

   The genotype data file stores each individual's genotypes at the SNPs to be tested. Its rows refer to the SNPs and its columns refer to the individuals. The requirements are the following:

- Tab or space delimited.
- The first row is a header line that starts with 'Chr SNP cm bp ', followed by the individual IDs. Every person in this file has to be present in the phenotype file and vice versa. In the current version of the software, individuals must be listed in the same order as in the phenotype data file, and if not, an error will be reported.
- The first four columnns are annotation information about each SNP: chromosome number (positive integer), SNP ID (numeric or alphanumeric), centiMorgan (numeric), base pair position (numeric). The annotation information will not be directly used in the algorithm, but will be printed out with the results, except for centiMorgan, which will neither be used in the algorithm nor printed out with the results.
- Starting with Column 5, the genotypes of the individuals are given in the order listed in the header row. For a given SNP, the genotype of each individual is represented by two numbers, one for each allele. For example, `1 1 1 2` are for the genotypes of 2 individuals. This means that effectively there are more columns in the rows with genotype data than in the header row. The rows with the genotype data have the exact format that can be generated from `plink` with `--transpose --recode12` commands. Thus the CARAT format can be easily generated from `plink` by adding a header line.
- Alleles are labeled with '1' and '2'. Missing alleles should be labeled with '-9'. Currently the program does not support other allele labels. If other labels are present, the program will produce a warning message, and will proceed with the corresponding SNP ignored.
- Currently the program does NOT handle SNPs with more than 2 alleles.
- For a dataset with 4 individuals and 3 markers, the genotype file might look like

```
Chr  SNP       cm  bp       IND1    3    2    IND1
1    SNP1      1   1193109  1     2  1  1  2  1  2     2
2    rs424964  1   2099319  2     1  1  2  2  1  1     2
1    1a        1   902900   1     1  1  1  2  2  2     2
```

The flag `-g geno.txt` allows the user to specify the name of the genotype data file. The filename defaults to `geno.txt` if this flag is not used. To specify another filename, replace `geno.txt` with the appropriate filename.

3. **Optional eigen-decomposition file (specified by flag -e)**

The eigen-decomposition file is optional. It allows the program to improve computational efficiency by making use of existing eigen-decomposition 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 Φ be the $n \times n$ genetic relationship matrix. Let $\Phi = VDV^{-1}$ be an eigen-decomposition of Φ, where D is a diagonal matrix with the eigenvalues as the diagonal elements and V is an orthogonal matrix containing the eigenvectors. The eigen-decomposition 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 Φ underlying this file should correspond exactly to the set and ordering of individuals in the phenotype data file. In particular, the dimension

of Φ should equal the number of individuals in the phenotype data file. Mismatched dimension will likely result in a segmentation fault. Mismatched ordering will result in incorrect results.

- A scenario in which eigen-decomposition results may be available prior to the analysis occurs when, prior to the current analysis, CARAT has been run on the same set of individuals (with the same genetic relationship matrix) as in the current analysis, and thus the eigen-decomposition results can be reused.

The optional flag `-e eig_file` allows the user to indicate the availability of the eigen-decomposition file and its filename. There is no default filename. `eig_file`can be replaced by the appropriate filename.

4. **Optional GRM genotype file (specified by flag `-G` when `-e` is not used)**

When the flag `-e` is not used, the program offers the option to perform eigen-decompotion on a genetic relationship matrix computed based on the genotypes at a large number of SNPs. This file is a text file that includes the genotypes of all the individuals at SNPs from which to compute the genetic relationship matrix. The file may include all genotyped SNPs across the genome, or only a subset of them, for example by excluding the SNPs on the same chromosome as the tested SNP(s). The file may or may not be the same file as the genotype file specified by the flag `-g`, which includes the genotypes of the tested SNPs.

The GRM genotype file is required to have the exact same format as the genotype file (option `-g`). In particular, all individuals in the phenotype file must be included in the GRM genotype file in the same order.

The optional flag `-G grm_geno_file` specifies the name of the GRM genotype file. There is no default filename. `grm_geno_file` can be replaced by the appropriate filename.

5. **Optional GRM file (specified by flag `-R` when `-e` is not used)**

When the flag `-e` is not used, as an alternative to using the GRM genotype file to compute the genetic relationship matrix, the program offers the option to use a known genetic relationship matrix. The GRM file is a text file listing one entry of the matrix per row. The columns are

```
familyID indiv1 indiv2 entry_in_matrix
```

where "familyID" is a place holder and is not to be used by the program.

- The order of the rows does not matter.
- When `indiv1` and `indiv2` are the same, the row corresponds to a diagonal element in the matrix.
- For the tested individuals (i.e. those in the phenotype file), the individual IDs should agree with the phenotype file. However, the GRM file is allowed to contain individuals that are not to be included in the analysis. The program will simply ignore the rows that contain an individual ID that is not in the phenotype file, if any.

- It is important to ensure that the genetic relationship matrix for the tested individuals is symmetric and positive semi-definite. To ensure symmetry, 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 message and will ignore the information given the first time it appears. If a pair is not found in the file, zero will be used for the corresponding entry in the matrix. If the matrix is not positive semi-definite, the program will return an error.
- In practice, the known genetic relationship matrix may come from another program. Another scenario in which the genetic relationship matrix may be available prior to the analysis occurs when, prior to the current analysis, CARAT has been run on the same or a larger set of individuals as in the current analysis, and thus the genetic relationship matrix may have been output by the `-r` option and may be reused.
- For a sample with individuals, `IND1, 3, 2` and `IND3`, the GRM file may look like

```
IND1 3 0.01
3 3 1.02
2 IND1 0.02
IND3 2 0
IND3 IND3 0.99
2 2 1.01
IND1 IND1 0.99
```

  This file will result in the following genetic relationship matrix:

```
0.99 0.01 0.02 0.00
0.01 1.01 0.00 0.00
0.02 0.00 1.02 0.00
0.00 0.00 0.00 0.99
```

Remark: In the current version of the program, a user-specified file name should not exceed 2000 characters in length. Otherwise, an error will be reported. To increase this maximum length, one may open the file `carat.cpp` 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.

5 Output

The program will output up to 4 files: a text file recording the primary results of the association analysis, another text file that contains the parameter estimation results, an optional binary file that records lists the eigenvalues and the eigenvectors of the genetic relationship matrix, and another optional file that records the genetic relationship matrix computed from the GRM genotype file. The default filenames of are `CARAT_out.txt`, `CARAT_param.txt`, `CARAT_eig.txt`, and `CARAT_GRM.txt`, where the user may choose to replace "`CARAT`" by another prefix by the option `-o`. The file for the association analysis results will not be part of the output if flag `-n` is used. Below we explain each of the output files (with the default filenames.)

1. **CARAT_out.txt** is the primary output file for the results of the association tests. This file lists the testing result for each of SNPs in the order they are included in the genotype data

file. The prefix of the filename can be changed using the -o option. It contains the following information:

- Chromosome
- SNP ID
- Base pair position
- The p-value

2. **CARAT_param.txt** is a text file containing variance component and covariate parameter estimates under the null hypothesis of no association. The prefix of the filename can be changed using the -o option. With 2 covariates, the file might look like:

```
Parameter              Null_Estimate
Variance_Parameter_xi  2.1338e-01
Intercept              -1.2003e+01
Covariate_1            4.5116e-02
Covariate_2            8.4172e+00
```

3. **CARAT_eig** is a binary file that stores the eigenvalues and eigenvectors of the genetic relationship matrix, as computed by the CARAT program. It will be part of the output only when -G or -R is used. The formatting of this output file is the same as that of the binary eigen-decomposition input file described in Section 4.3.

4. **CARAT_GRM.txt** is a text file that stores the genetic relationship matrix, as computed by the CARAT program based on the input GRM genotype data file. It will be part of the output only when both -G and -r are used. The formatting of this output file is the same as that of the GRM input file described in Section 4.5.

6 Examples

The directory CARAT/examples provides example input files: pheno_ex.txt, geno_ex.txt, GRM_geno_ex.txt, GRM_ex.txt and eigfile_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 CARAT (you may need to run chmod a+x CARAT before using the binary executable).

1. `./CARAT -p pheno_ex.txt -g geno_ex.txt -e eigfile_ex -o chr21`

   This command instructs the program to perform the CARAT method using the known eigen-decomposition results given in the file eigfile_ex. The -o option specifies the prefix of the output files to be chr21. The program will generate two output files: chr21_out.txt and chr21_param.txt.

2. `./CARAT -p pheno_ex.txt -g geno_ex.txt -R GRM_ex.txt`

   With this command instructs, the program will first read the genetic relationship matrix from GRM_geno_ex.txt and then perform eigen-decomposition, whose result will be stored in output file with the default filename (as -o is not used) CARAT_eig. There will be two additional output files with default filenames: CARAT_out.txt and CARAT_param.txt.

3. `./CARAT -p pheno_ex.txt -n -G GRM_geno_ex.txt -r`

   This command instructs the program to only fit null model without running any association test. The genetic relationship matrix is constructed from the genotype data read from `GRM_geno_ex.txt`. Phenotype data are read from `pheno_ex.txt`. No genotype data file will be read. There will be three output files: `CARAT_param.txt`, which stores the parameter estimation results, `CARAT_eig`, which stores the eigen-decomposition results, `CARAT_GRM.txt`, which stores the genetic relationship matrix as requested by the option `-r`.

7 Acknowledgments

We gratefully acknowledge:

- Eigen. We used the package for some of the matrix computations.
- R. We used the Brent_fmin subprogram from the function `optimize()`.
- GNU Scientific Library. We used the `gsl_cdf_ugaussian_Q` routine to compute the p-value of the test statistic.
- LAPACK. We used the `dsyevr` routine and its dependencies as one of the options for performing eigen-decomposition.

8 References

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