

A MANUAL ON HOW TO USE GWASREADER

HENGRUI LUO

GWASReader extends the R flexible computational framework and supports buffering computation when reading a large GWAS dataset using JRI mechanism with Java.

To use GWASReader, you need R 3.4.1 or above installed and *JRE*1.8.0/OpenJDK 8 or above.

- (1) Make sure you have download all the files in **prepared** folder.
- (2) Unzip the files on directory `/GWASReader/`
- (3) Unzip the JRI files on Directory `/GWASReader/`. You are expected to see `/GWASReader/org` hierarchy.

Following works were done under `/GWASReader/`.

- (4) Prepare your BSSeq data files.
 - (a) Put the first population BSSeq data and rename it as `/GWASReader/mc_h1_Y` without extension name.

These datafile names were chosen in accordance with the data format Lister, Ryan, et al. "*Human DNA methylomes at base resolution show widespread epigenomic differences.*" *nature* 462.7271 (2009): 315.

Further modification of the file names and data formats were possible.
 - (b) Put the second population BSSeq data and rename it as `/GWASReader/mc_i90_Y` without extension name.
- (5) Execute the command line `javac GWASReaderY.java`, the program will automatically select environmental variable `java.library.path` when you set things up correctly. If compilation succeed, then you will see
- (6) Check that environmental variable `R_HOME` is set up correctly

- (a) In Unix/Linux system, you can use a command line like

```
export R_HOME = /usr/lib64/R
```

- (b) In Windows, you can follow Computer→Properties→Advanced system settings to change or add the environmental variable R_HOME.

- (7) Use following command to execute the compiled class GWASReaderY.class under the /GWASReader/ directory,

```
java -Djava.library.path = /.../rJava/jri GWASReaderY
```

A minimal example of all command lines after you unzipped the files with correct hierarchy shall look like (↵ means execute OR enter key depending on your setting.)

```
javac GWASReaderY.java ↵
```

```
export R_HOME = /usr/lib64/Rjava ↵
```

```
-Djava.library.path =
```

```
/home/user.name/R/x86_64 - redhat - linux - gnu - library/3.x/rJava/jri GWASReaderY ↵
```

- (8) If the program is successfully working, then

- (a) In Unix/Linux you will see lines of testing results popping up and
(b) In Windows, you can see the same thing with slight difference if you redirect the printing source.

- (9) Three files will be generated after a successful execution of the program. They are /GWASReader/Index_ChrY.txt, /GWASReader/Combined_ChrY.txt, /GWASReader/Tested_ChrY.txt.

- (a) /GWASReader/Index_ChrY.txt is produced by package.indexmaker.R which records the common base pair serial number that two population shared.

- (b) `/GWASReader/Combined_ChrY.txt` is the combined data collected from two populations according to matched indices in `/GWASReader/Index_ChrY.txt`. The data will be processed into the 2×2 binomial table format and ready to be analyzed by the statistical tests implemented in `package.test.R`.
- (c) `/GWASReader/Tested_ChrY.txt` is the file that contains the testing results for each base pair and should be taken to the next stage of analysis. According to our `javac GWASReaderY.java` file the resulting file will adopt the format for each line as following (Serial number/population data/p-values from corresponding statistical test).

	from population 1		from population 2				
Serial Number #	methyalted read	unmethyalted read	methyalted read	unmethyalted read	Fisher's Exact test	Chi-squared test	Storer-Kim test
57439464	24.0	10.0	20.0	17.0	0.22119235889554598	0.2344527262035078	0.161940434943863

FAQ

- Q: Why do I failed to run...

A: We suggested that you follow our instructions above and check where you deviate from it.

- Q: Is there an example?

A: Yes, when you unzipped the `prepared` file correctly and put `mc_h1_Y` and `mc_i90_Y` to the same folder, a minimal example is

```
javac GWASReaderY.java ✓
```

```
export R_HOME = /usr/lib64/Rjava ✓
```

```
-Djava.library.path =
```

```
/home/user.name/R/x86_64-redhat-linux-gnu-library/3.x/rJava/jri GWASReaderY ✓
```

- Q: What if I do not have corrected versions of R or Java?

A: Although we have managed to make the code compiled in lower versions of Java and R, its performance is not tested under those circumstances.

- Q: What if I want to execute it in IDE like Eclipse?

A: You can setup Eclipse runtime environment in the project properties panel, the way we did it makes it unnecessary to setup the `java.library.path` manually because we used `System.setProperty("java.library.path", DIRE)` to set the current working directory as the place where JRI library is located.

- Q: What if I get an “cannot start R” error?

A: If you are working with Unix/Linux, then you need to check if you specify the environment variable `export R_HOME = /usr/lib64/R` correctly. If you are working with Windows, a problem could be you have installed more than one versions of R or the environmental variables are not specified correctly.

- Q: What if I get a “java.lang.UnsatisfiedLinkError: no jri in java.library.path” error?

A: This should not happen when you use `prepared` files and the hierarchy are correctly unzipped. But one possibility is that when you compile, the `java.library.path` has not been set up correctly, so you can try following compile command

```
javac -cp GWASReader/JRI.jar:GWASReader/JEngine.jar GWASReaderY.java
```

- Q: How do I set up my `R_HOME` variable?

A: Start your R and type `System.getProperty("user.dir")` to see the directory that R installed.