FastEPRR: a Fast Estimator for Population Recombination Rate

目录

Introduction

This manual describes how to use FastEPRR, an R package for estimating population recombination rate.

FastEPRR has the cross-platform compatibility since R can run on different operating systems. The first section describes how to install R environment and FastEPRR package on different operating systems. The follow sections describe the parameters of main functions and several ways to run the main function on different operating systems. This manual is divided into several subsections according to different operating systems and functions, and users can read relevant sections directly.

All packages of FastEPRR and relevant example files are available at http://www.picb.ac.cn/evolgen/. If you have any questions about usage or report bugs, please feel free to contact haoziqian@picb.ac.cn or lihaipeng@picb.ac.cn.

Changes compared to FastEPRR1.0

[1] Several bug corrections
[2] Speed optimization
[3] Optional confidence interval estimations
[4] Optional demographic model simulation times
[5] Optional training set inputs
[6] Addition of a method to detect variable recombination rates within windows
[7] User manual update

How to cite

Please cite FastEPRR2.0 as following:

Hao, Z., Du, P., Pan, YH. et al. Fine human genetic map based on UK10K data set. Hum Genet (2022). https://doi.org/10.1007/s00439-021-02415-8

GAO Feng, MING Chen, HU Wangjie, LI Haipeng* (2016). New software for the fast estimation of population recombination rates (FastEPRR) in the genomic era. G3 Genes Genomes Genetics , Early Online March 30, 2016, doi: 10.1534/g3.116.028233

LIN Kao, FUTSCHIK Andreas, LI Haipeng* (2013). A fast estimate for the population recombination rate based on regression. Genetics 1 94:473 484.

Downloading and Installation

FastEPRR is a standard R package, so you should install R environment first. The minimum required version of R for FastEPRR is 2.14.0.

The subsequent subsections describe how to install FastEPRR package, and you can only read relevant sections according to specific operation system.

Linux/Unix

There are two ways to install FastEPRR package in Linux/Unix.

[1] Command line

Once the R environment has been installed, you can type the following command in the command line to install FastEPRR package.

```
$R CMD INSTALL "package_path/FastEPRR_version.tar.gz"
```

[2] Within R console

Typing "R" in the command line can start the R console.

```
$R
```

Then, you can type the following command to install FastEPRR package in the R console.

```
>install.packages("package_path/FastEPRR_version.tar.gz", repos = NULL,
type = "source")
```

The *package_path* is the file path of FastEPRR_version.tar.gz

Windows

There are three ways to install FastEPRR package in Windows. R console is needed when using the first two ways.

[1] Menu bar

"Package" $\longrightarrow$ "Install packages from local zip file..."

[2] Install package command

Type the following command at R console.

```
> install.packages("package_path/FastEPRR_version.zip", repos = NULL)
```

[3] Command line

This way avoids opening R console. Click

```
"Start" ⟶ "All Programs" ⟶ "Accessories" ⟶ "Run"
```

to open the Run box, then change to the R directory or add R directory to the PATH of your system, and type the following command in your Run box.

```
>R CMD INSTALL "package_path/FastEPRR_version.zip"
```

The *package_path* is the file path of FastEPRR_version.tar.gz

Please ignore the warning message during installation. Note that successful running of FastEPRR needs other R packages. "mboost" package is required when analyzing VCF file and both of "mboost" package and "seqinr" package are required when analyzing alignment file. "mboost" package is used to obtain the regression model and "seqinr" package is used to read alignment file. These two are standard R packages. ***FastEPRR will install them automatically when there is a need. This involves downloading packages to your computer. Therefore, you must have an active Internet connection when you first run FastEPRR and relevant packages are not available on your computer.***

Input

FastEPRR supports three common alignment formats (fasta, clustal and phylip) and VCF format. Missing data (denoted by "?") or gaps (denoted by "-") are allowed in your alignment file, and missing data (denoted by ".") is allowed in your VCF file. Segregating sites with two or more alleles are taken into consideration and indels (insertions and deletions) are excluded for VCF format.

Main function and arguments specification

Functions and arguments for alignment file

FastEPRR_ALN

The main function for analyzing alignment file is FastEPRR_ALN().

```
FastEPRR_ALN (alnFilePath = NULL, fileFormat = NULL, erStart = NU
LL, erEnd = NULL, winLength = NULL, stepLength = NULL, getCI =
TRUE, replicateNum = 100, demoParameter = NULL, trainingSet1 =
NULL, trainingSet2 = NULL, outputFilePath = NULL)
```

Parameter	Type	Description

alnFilePath	Character	Full path of the alignment file. The file name should end with [.fas|.aln|.phy].
fileFormat	Numeric	Format number (1: fasta, 2: clustal, 3:phylip) of the input file. Numbers except 1, 2 and 3 are not allowed.
erStart	Numeric	Start position from which to estimate recombination rate. If missing, calculation will start from the beginning of the input. Default is NULL.
erEnd	Numeric	End positon relative to erStart. The maximum value of erEnd is the length of the alignment data. If missing, calculation will end at the last of the input. Default is NULL. If you want to estimate the whole input data, set erStart = NULL and erEnd = NULL.
winLengh	Numeric	Window size of the sliding window. The unit is 1. If missing, all input data will be regard as one window. Default is NULL.
stepLength	Numeric	Length of the sliding step. The unit is 1. Default is NULL. If stepLength is NULL and winLength is not NULL, data will be scanned by sliding windows with the step length equals to winLength successively. This is equivalent to set stepLength equals to winLength.
getCI	Boolean	Whether or not to get confidence interval (CI). Default is TRUE. Estimating CI is time-consuming. If you do not need CI, set getCI = FALSE.
replicateNum	Numeric	Number of estimation replicates. Default is 100. Larger value means better accuracy and longer time.
demoParameter	Character	Population demographic changes the sample experienced. Default is NULL. The format is consistent with the format of Hudson’s ms simulator.
trainingSet1	Character	Rho values of training set 1. Default is (0.0, 0.5, 1.0, 2.0, 5.0, 10.0, 20.0, 40.0, 70.0, 110.0, 170.0). If a specified training set1 is needed, please separate each by “;”. Example: “0.0;1.0;2.0;5.0;10.0;20.0;40.0;70.0”
trainingSet2	Character	Rho values of training set 2. If the observed number of different haplotypes (H) does not fall within the range of H in the training set 1, training set 2 will be used. Default is (180.0, 190.0, 200.0, 220.0, 250.0, 300.0, 350.0). If a specified training set 2 is needed, please separate each by “;”. The minimum value of training set 2 should be larger than the maximum value of training set 1 Example: “200.0;250.0;300.0;350.0;400.0”
outputFilePath	Character	Full path of the output file. If the file exists, FastEPRR will remove it and open a new one.

Notes

[1] alnFilePath, fileFormat and outputFilePath must be assigned.
[2] All character values should be within “”. Please refer to examples.
[3] A window will be excluded if the number of SNPs except folded singletons is less than log(the number of sample).
[4] The unit of output “rho” is 4Ner (Ne is the effective population size and r the recombination rate for the window).

Examples

The following examples show different parameter configurations. You should read them carefully before using this function.

Example 1:

```
FastEPRR_ALN(alnFilePath = "exampleData_Path/alignmentExampleFile.
fas", fileFormat = 1, outputFilePath = "exampleData_Path/alignment
ExampleFileRecRate.txt")
```

Comments:
[1] The input file is fasta format (fileFormat = 1).
[2] This estimates recombination rate for the whole input as one window (winLenght = NULL, stepLength = NULL, erStart = NULL, erEnd = NULL).
[3] The output file “alignmentExampleFileRecRate.txt” contains estimated recombination rate.

Example 2:

```
FastEPRR_ALN(alnFilePath = "exampleData_Path/alignmentExampleFile.
fas", fileFormat = 1, outputFilePath = "exampleData_Path/alignment
ExampleFileRecRate.txt", winLength = 500, stepLength = 250, demoP
arameter = "-G 6.93 -eN 0.1 0.5")
```

Comments:
[1] The sliding-window strategy is used to estimate recombination rate for the whole input (erStart = NULL, erEnd = NULL). The sliding-window size is 500 (winLength = 500), and the step length is 250 (steplength = 250).
[2] Population demographic parameters (demoParameter = "-G 6.93 -eN 0.1 0.5") represent the population experienced exponential growth. The time of expansion is 0.1, and N_0/N_1 is 5, where N_0 and N_1 are current and ancestral effective population size, respectively. You can refer to the manual of Hudson’s ms simulator for more details.
[3] The output file “alignmentExampleFileRecRate.txt” contains estimated recombination rate.
Example 3:

```
FastEPRR_ALN(alnFilePath = "exampleData_Path/alignmentExampleFile.
fas", fileFormat = 1, outputFilePath = "exampleData_Path/alignmentE
xampleFileRecRate.txt", winLength = 500, stepLength = 250, erStart
= 51, erEnd = 1000, getCI = FALSE, trainingSet1 = "0;0.5;1;2;3;4;5;
8;10;15;20;25;30", trainingSet2 = "35;40;45;50")
```

Comments:
[1] The sliding-window strategy is used to estimate recombination rate for interval between positon 51 and positon 1000 (erStart = 51, erEnd = 1000). The sliding-window size is 500 (winLength = 500), and the step length is 250 (steplength = 250).
[2] It saves a lot of time when no need for CI (getCI = FALSE).
[3] The training set 1 is reset to (0, 0.5, 1, 2, 3, 4, 5, 8, 10, 15, 20, 25, 30), and the training set 2 is reset to (35, 40, 45, 50) (trainingSet1 = "0;0.5;1;2;3;4;5;8;10;15;20;25;30", trainingSet2 = "35;40;45;50").
[4] The output file "alignmentExampleFileRecRate.txt" contains estimated recombination rate.

Functions and arguments for VCF files

There three steps for analyzing VCF files. Each step is represented by one function. The following sections describe the three functions in detail.

FastEPRR_VCF_step1

This function is used to scan the input file and save required information for step2.

```
FastEPRR_VCF_step1(vcfFilePath = NULL, erStart = NULL, erEnd = N
ULL, winLength = NULL, stepLength = NULL, winDXThreshold = 30,
samConsidered = "all", qualThreshold = 20, gapFilePath = NULL, src
OutputFilePath = NULL)
```

Parameter	Type	Description
vcfFilePath	Character	Full path of the VCF file. The file name should end with [.vcf|.gz]. The VCF file should include the polymorphic information of only one chromosome.
erStart	Numeric	Start position from which to estimate recombination rate. The position corresponds to "POS" column in VCF file. If missing, calculation will start from the beginning of the input. Default is NULL.
erEnd	Numeric	End positon relative to erStart. The maximum value of erEnd is the length of the VCF data. If missing, calculation will end to the last of the input.

		Default is NULL. If you want to estimate the whole input data, set erStart = NULL and erEnd = NULL.
winLengh	Numeric	Window size of the sliding window. The unit is 1. Since the input VCF file contains polymorphic information of the whole chromosome, it is not allowed to be NULL.
stepLength	Numeric	Length of the sliding step. The unit is 1. Default is NULL. If stepLength is NULL, data will be scanned by sliding windows with the step length equals to winLength successively. This is equivalent to set stepLength equals to winLength.
winDXThreshold	Numeric	Minimum threshold of sum of folded doubletons and folded xtons for each window. The window will be excluded if the sum of folded doubletons and folded xtons is smaller than winDXThreshold. Default is 10 and the minimum is 2.
samConsidered	Character	Samples considered in the input file. “all” means all individuals are considered. If you want to specify interested individuals, input names of these individuals and separate each by “;”. If you only want to study the first chromosome of a given individual, input “[1:0]” after the name of this individual. If you only want to study the second chromosome of a given individual, input “[0:1]” after the name of this individual. If you want to study both chromosomes, do not input anything after the name of this individual. Default is “all”. Example: “all[1:0]” “NA0007;NA0010[1:0];NA0016[0:1]”
qualThreshold	Numeric	Threshold of SNP quality. It corresponds to “QUAL” column in VCF file. SNPs will be excluded if their quality scores are less than qualThreshold. Default is 20.
gapFilePath	Character	Full path of the gap file. The gap file saves gap information of the input chromosome and has a specified format. Windows which overlap with gap intervals in gapFilePath are excluded. Please follow the format of our example file. Default is NULL. The first base in gap file is 0, and the End base of each interval is not included (please refer to position definition of BED format).
srcOutputFilePath	Character	Full path of the output file. The out file contains useful information of each window for step 2.

Notes

[1] vcfFilePath, winLengh and srcOutputFilePath must be assigned.
[2] All character values should be within “”. Please refer to examples.
[3] Be careful of the format when specifying interested samples. Please refer to examples.

Examples

The following examples show different parameter configurations. You should read them carefully before using this function.

Example 1:

```
FastEPRR_VCF_step1(vcfFilePath = "exampleData_Path/CHB.chr1.omni_
example.vcf.gz", winLength = 50000, winDXThreshold = 10, srcOutpu
tFilePath = "exampleData_Path/step1/chr1")
```

Comments:
[1] This example shows how to analyze all individuals of chromosome 7. The example file is a part of 1000 Genomes phased OMNI data set, which includes several different populations.
[2] The start position is the start of the input file (erStart = NULL), and the end position is the end of the input file (erEnd = NULL).
[3] All individuals (samConsidered = “all”) are taken into consideration.
[4] The sliding-window size is 50 kb (winLength = 50000), and the step length is also 50 kb (stepLength = NULL).
[5] Windows are excluded if the number of folded doubleton and xton is fewer than 10 (winDXThreshold = 10).
[6] Results are saved in file “chr1”.

Example 2

```
FastEPRR_VCF_step1(vcfFilePath = "exampleData_Path/CHB.chr1.omni_
example.vcf.gz", erStart = 1591421, winLength = 50000, winDXThres
hold = 10, samConsidered = “NA18525[1:0];NA18526[0:1];NA18527;N
A18528[1:0];NA18530[0:1];NA18532”, gapFilePath = "exampleData_Pa
th/chr1_gap.txt", srcOutputFilePath = "exampleData_Path/step1/chr1
")
```

Comments:
[1] The start position is 1591421 (erStart = 1591421), and the end position is the end of the input file (erEnd = NULL).
[2] The sliding-window size is 50 kb (winLength = 50000), and the step length is also 50 kb (stepLength = NULL).

[3] The first chromosome of NA18525 and NA18528, the second chromosome of NA18526 and NA18530, as well as both chromosomes of NA18527 and NA18532 are considered (samConsidered = "NA18525[1:0];NA18526[0:1];NA18527;NA18528[1:0];NA18530[0:1];NA18532").
[4] Windows are excluded if the number of folded doubleton and xton is fewer than 10 (winDXThreshold = 10).
[5] Gap information is used.
[6] Results are saved in file "chr1".

Example 3:

```
FastEPRR_VCF_step1(vcfFilePath = "exampleData_Path/CHB.chr1.omni_example.vcf.gz", erStart = 1591421, erEnd = 33841300, winLength = 50000, stepLength = 25000, winDXThreshold = 10, gapFilePath = "exampleData_Path/chr1_gap.txt", srcOutputFilePath = "exampleData_Path/step1/chr1")
```

Comments:
[1] The start position is 1591421 (erStart = 1591421), and the end position is 159091420 (erEnd = 159091420).
[2] The sliding-window size is 50 kb (winLength = 50000), and the step length is 25 kb (stepLength = 25000).
[3] All individuals (samConsidered = "all") are taken into consideration.
[4] Windows are excluded if the number of folded doubleton and xton is fewer than 10 (winDXThreshold = 10).
[5] Gap information is used.
[6] Results are saved in file "chr1".

FastEPRR_VCF_step2

This function is used to estimate recombination rate for each window scanned in step 1. Only one model is constructed for specific folded doubleton and folded xton configuration, and this model is used to infer recombination rate for all windows which have the same number of folded doubleton and folded xton. The output files are named by the number folded doubleton and foled xton.

Note that if your data has the same "demoParameter", the input of this step should contain all chromosomes' results of step1 to avoid wasting time on Repetitive modeling.

```
FastEPRR_VCF_step2(srcFolderPath = NULL, jobNumber = 1, currJob = 1, demoParameter = NULL, replicateNum = 100, getCI = TRUE, trainingSet1 = NULL, trainingSet2 = NULL, DXOutputFolderPath = NULL)
```

Parameter	Type	Description
srcFolderPath	Character	Full path of parent directory of "srcOutputFilePath" in step 1. This directory should contain information of all chromosomes.
jobNumber	Numeric	Number of jobs used to estimate recombination rate in parallel. Default is 1.
currJob	Numeric	Current job number. Default is 1. If jobNumber = 100, currJob should be 1,2…100.
demoParameter	Character	Population demographic changes the sample experienced. Default is NULL. The format is consistent with the format of Hudson's ms simulator.
replicateNum	Numeric	Number of estimation replicates. Default is 100. Larger value means better accuracy and longer time.
getCI	Boolean	Whether or not to get confidence interval (CI). Default is TRUE. Estimating CI is time-consuming. If you do not need CI, set getCI = FALSE.
trainingSet1	Character	Rho values of training set 1. Default is (0.0, 0.5, 1.0, 2.0, 5.0, 10.0, 20.0, 40.0, 70.0, 110.0, 170.0). If a specified training set 1 is needed, please separate each by ";". Example: "0.0;1.0;2.0;5.0;10.0;20.0;40.0;70.0"
trainingSet2	Character	Rho values of training set 2. If the observed number of different haplotypes (H) dose not fall within the range of H in the training set 1, training set 2 will be used. Default is (180.0, 190.0, 200.0, 220.0, 250.0, 300.0, 350.0). If a specified training set 2 is needed, please separate each by ";". The minimum value of training set 2 should be larger than the maximum value of training set 1 Example: "200.0;250.0;300.0;350.0;400.0"
DXOutputFolderPath	Character	Full path of output directory. Note that all jobs should have the same DXOutputFolderPath.

Notes

[1] srcFolderPath and DXOutputFolderPath must be assigned, and **DO NOT** add “/” at the end of srcFolderPath or DXOutputFolderPath.
[2] All character values should be within “”. Please refer to examples.
[3] Total job number is jobNumber, therefore the minimum value of currJob is 1 and the maximum value is jobNumber.

Examples

Example 1

```
FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", jobNumber = 100, currJob = 1, DXOutputFolderPath = "exampleData_Path/step2")
```

```
FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", jobNumber = 100, currJob = 7, DXOutputFolderPath = "exampleData_Path/step2")
```

```
FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", jobNumber = 100, currJob = 100, DXOutputFolderPath = "exampleData_Path/step2")
```

Comments:
[1] 100 jobs are used in total (jobNumber = 100). Only three jobs numbered by 1, 7 and 100 are listed here.
[2] The input for step 2 is the parent directory of “srcOutputFilePath” in step 1.
[2]The three jobs must have the same output path (DXOutputFolderPath = "*exampleData_Path*/step2").

Example 2:

```
FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", demoParameter = "-G 6.93 -eN 0.1 0.5", DXOutputFolderPath = "exampleData_Path/step2")
```

Comments:
[1] Only one job is used to perform estimation (default jobNumber is 1).
[2] There is no need to set currjob since jobNumber = 1.
[3] Population demographic parameters (demoParameter = "-G 6.93 -eN 0.1 0.5") represent the population experienced exponential growth. The time of expansion is 0.1, and N_0/N_1 is 5, where N0 and N1 are current and ancestral effective population size, respectively. You can refer to the manual of Hudson’s ms simulator for more details.

Example 3:

```
FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", getCI
 = FALSE, trainingSet1 = "0;0.5;1;5;10;25;50;80;120;170;250", trainin
gSet2 = "260;300;350;450", DXOutputFolderPath = "exampleData_P
ath/step2")
```

Comments:
[1] Only one job is used to perform estimation (default jobNumber is 1).
[2] There is no need to set currjob since jobNumber = 1.
[3] It saves a lot of time when no need for CI (getCI = FALSE).
[4] The training set 1 is reset to (0, 0.5, 1, 5, 10, 25, 50, 80, 120, 170, 250), and the training set 2 is reset to (260, 300, 350, 450) (trainingSet1 = "0;0.5;1;5;10;25;50;80;120;170;250;", trainingSet2 = "260;300;350;450").

FastEPRR_VCF_step3

This function is used to merge the output of step 1 and step 2. The output of this step is one file per chromosome which includes recombination rate of all sliding windows on this chromosome.

```
FastEPRR_VCF_step3(srcFolderPath = NULL, DXFolderPath = NULL, fin
alOutputFolderPath = NULL)
```

Parameter	Type	Description
srcFolderPath	Character	Full path of parent directory of “srcOutputFilePath” in step 1. This directory should contain information of all chromosomes.
DXFolderPath	Character	Value of “DXOutputFolderPath” in step 2.
finalOutputFolderPath	Character	Full path of output directory.

Notes

[1] All parameters must be assigned, and they should be within “”. **DO NOT** add “/” at the end of parameters.
[2] The unit of output “rho” is 4Ner (Ne is the effective population size and r the recombination rate for the window).

Examples

Example 1:

```
FastEPRR_VCF_step3(srcFolderPath = "exampleData_Path/step1", DXF
 olderPath = "exampleData_Path/step2", finalOutputFolderPath = "ex
 ampleData_Path/step3")
```

Comments:
[1] The parent directory of the output of step 1 is used as the value of "srcFolderPath" (srcFolderPath = "*exampleData_Path*/step1"). The output directory of step 2 is used as the value of "DXFolderPath" (DXFolderPath = "*exampleData_Path*/step2").

Variable recombination rates within windows

When estimation recombination rate for a given window, we assume that it is constant. However, this may not be correct. FastEPRR_VAR() is designed to investigate the effect of a variable recombination rate within a given window. In this function, we assume that recombination rate of the window which is excluded because of "winDXThreshold" is the mean of nearest estimated window of both sides. **Note that you can only use this function when "stepLength" is half of "winLength".**

```
FastEPRR_VAR(varFilePath = NULL, winLength = NULL, stepLength =
 NULL, gapFilePath = NULL, varOutputFilePath = NULL)
```

Parameter	Type	Description
varFilePath	Character	Full path of an output file of FastEPRR_ALN() or FastEPRR_VCF_step3().
winLength	Numeric	Window size of the sliding window. This value should be equal to the "winLength" in FastEPRR_ALN() or FastEPRR_VCF_step1().
stepLength	Numeric	Length of the sliding step. This value should be equal to the "stepLength" in FastEPRR_ALN() or FastEPRR_VCF_step1().
gapFilePath	Character	Full path of the gap file. If any, this value should be equal to the "gapFilePath" in FastEPRR_VCF_step1(). Default is NULL.
varOutputFilePath	Character	Full path of the output file.

Notes

[1] varFilePath, winLength, stepLength and varOutputFilePath must be assigned.
[2] All character values should be within “”.

Examples

Example 1:

```
FastEPRR_VAR(varFilePath = "exampleData_Path/step3/chr1", winLength = 50000, stepLength = 25000, gapFilePath = "exampleData_Path/chr1_gap.txt", varOutputFilePath = "exampleData_Path/varRate/chr")
```

Comments:
[1] The output file in “finalOutputFolderPath” of step 3 is used as the input in this function (varFilePath = "*exampleData_Path*/step3/chr1").
[2] winLength, stepLength and gapFilePath should be the same as these used in step 1.

Transform Rho to r(cM/Mb)

The unit of output “rho” is 4Ner (Ne is the effective population size and r the recombination rate for the window). If you know the effective population size, you can use FastEPRR_rho2r() to transform Rho to r(cM/Mb).

```
FastEPRR_rho2r(inputFilePath = NULL, outputFilePath = NULL, Ne = null)
```

Parameter	Type	Description
inputFilePath	Character	Full path of an output file of FastEPRR_ALN(), FastEPRR_VCF_step3() or FastEPRR_VAR().
outputFilePath	Character	Full path of output file path.
Ne	Numeric	Effective population size

[1] The unit of output r is cM/Mb, regardless of window size.

Usage

Linux/Unix

There are three modes to run FastEPRR on Linux/Unix: bash script, batch mode and within R console. Running R commands from a bash script allows submitting them to a computer cluster, which is suitable for parallel executing. For the batch mode, R provides a standard command line (R CMD BATCH) for running R. The batch modes allows executing a set of commands in sequence.

The following section describe these three modes for running alignment file and VCF file, respectively.

ALN

[1] Batch mode

First, the following commands should be written in a file, for example, batchExample_ALN.R

```
Library(FastEPRR)
FastEPRR_ALN(alnFilePath = "exampleData_Path/alignmentExampleFile.
 fas", fileFormat = 1, outputFilePath = "exampleData_Path/alignmen
 tExampleFileRecRate.txt", winLength = 100, stepLength = 50)
```

Then, type the following command in the command line to run the example.

```
$R CMD BATCH batchExample_ALN.R
```

The outputs generated by FastEPRR_ALN() are saved in the output file named alignmentExampleFileRecRate.txt automatically.

[2] Within R console

Type “R” in the command line to open the R console. Then in the R console, the user can type the following commands.

```
>Library(FastEPRR)
>FastEPRR_ALN(alnFilePath = "exampleData_Path/alignmentExampleFil
 e.fas", fileFormat = 1, outputFilePath = "exampleData_Path/alignme
 ntExampleFileRecRate.txt", winLength = 100, stepLength = 50)
```

[3] Bash script mode

First, the following commands should be written in a file, for example, bashExample_ALN.sh.

```
#!bin/bash
R_Path/R --no-save << EOF
Library(FastEPRR)
FastEPRR_ALN(alnFilePath = "exampleData_Path/alignmentExampleFile.
 fas", fileFormat = 1, outputFilePath = "exampleData_Path/alignmen
 tExampleFileRecRate.txt", winLength = 100, stepLength = 50)
EOF
```

Then, submit bashExample_ALN.sh.

```
qsub –cwd –q queue.q bashExample_ALN.sh
```

Please refer to the introduction of “qsub” or other command line used to submit jobs on your system for details.

VCF

[1] Batch mode

[1] Step 1

One R file is created for each chromosome, for example, step1_chr1.R.

```
Library(FastEPRR)
FastEPRR_VCF_step1(vcfFilePath = "/exampleData_Path/ALL.chr1.omni
 _example.vcf.gz", winLength = 50000, winDXThreshold = 10, gapFile
 Path = "/exampleData_Path/chr1_gap_example.txt", srcOutputFilePat
 h = "/exampleData_Path/step1/chr1")
```

Then, type the following command in the command line to run the example.

```
$R CMD BATCH step1_chr1.R
```

Of course, command lines for multiple chromosomes can be written in one R file, for example, step1_total.R.

```
Library(FastEPRR)
FastEPRR_VCF_step1(vcfFilePath = "/exampleData_Path/ALL.chr1.omni
 _example.vcf.gz", winLength = 50000, winDXThreshold = 10, gapFile
 Path = "/exampleData_Path/chr1_gap_example.txt", srcOutputFilePat
 h = "/exampleData_Path/step1/chr1")

…

FastEPRR_VCF_step1(vcfFilePath = "/exampleData_Path/ALL.chr22.omn
 i_example.vcf.gz", winLength = 50000, winDXThreshold = 10, gapFil
 ePath = "/exampleData_Path/chr22_gap_example.txt", srcOutputFile
 Path = "/exampleData_Path/step1/chr22")
```

Then, type the following command in the command line to run the example.

```
$R CMD BATCH step1_total.R
```

[2] Step 2

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1". The step2.R is described as follows.

```
Library(FastEPRR)
FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", jobN
 umber = 1, currJob = 1, DXOutputFolderPath = "exampleData_Path
 /step2")
```

Then, type the following command in the command line to run the example.

```
$R CMD BATCH step2.R
```

[3] Step 3

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1" and output files of step 2 have been saved in "exampleData_Path/step2". The step3.R is described as follows.

```
Library(FastEPRR)
FastEPRR_VCF_step3(srcFolderPath = "exampleData_Path/step1", DXF
 olderPath = "exampleData_Path/step2", finalOutputFolderPath = "ex
 ampleData_Path/step3")
```

Then, type the following command in the command line to run the example.

```
$R CMD BATCH step3.R
```

[2] Within R console

[1] Step 1

Type "R" in the command line to open the R console. Then in the R console, the user can type the following commands.

```
>Library(FastEPRR)
>FastEPRR_VCF_step1(vcfFilePath = "exampleData_Path/ALL.chr1.omni
 _example.vcf.gz", winLength = 50000, winDXThreshold = 10, gapFile
 Path = "exampleData_Path/chr1_gap_example.txt", srcOutputFilePath
  = "exampleData_Path/step1/chr1")
```

[2] Step 2

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1". Type "R" in the command line can open the R console. Then in the R console, the user can type the following commands.

```
>Library(FastEPRR)
>FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", job
 Number = 1, currJob = 1, DXOutputFolderPath = "/exampleData_P
 ath/step2")
```

[3] Step 3

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1" and output files of step 2 have been saved in "exampleData_Path/step2". Type "R" in the command line can open the R console. Then in the R console, the user can type the following commands.

```
Library(FastEPRR)
FastEPRR_VCF_step3(srcFolderPath="exampleData_Path/step1", DXFold
 erPath="exampleData_Path/step2", finalOutputFolderPath = "exampl
 eData_Path/step3")
```

[3] Bash script mode

When analyzing VCF files, we recommend you to use **bash script mode**, especially when you want to estimate recombination rate for the whole genome. The following examples show how to execute three steps for VCF file on a Linux cluster in this mode. Suppose that a grid computing computer cluster software system (e.g. Oracle Grid Engine) is available, and jobs can be submitted via Linux commands (e.g. qsub).

[1] Step 1

One bash script is created for each chromosome, for example, step1_chr1.sh.

```
#!bin/bash
R_Path/R – no-save << EOF
Library(FastEPRR)
FastEPRR_VCF_step1(vcfFilePath = "exampleData_Path/ALL.chr1.omni_
 example.vcf.gz", winLength = 50000, winDXThreshold = 10, gapFile
 Path = "exampleData_Path/chr1_gap_example.txt", srcOutputFilePath
  = "exampleData_Path/step1/chr1")
EOF
```

r.

```
#!bin/bash
R_Path/R – no-save << EOF
Library(FastEPRR)
FastEPRR_VCF_step1(vcfFilePath = "exampleData_Path/ALL.chr22.omni
 _example.vcf.gz", winLength = 50000, winDXThreshold = 10, gapFile
 Path = "exampleData_Path/chr22_gap_example.txt", srcOutputFilePat
 h = "exampleData_Path/step1/chr22")
EOF
```

Then, another script, such as step1_submit.sh need creating for submit above *.sh files.

```
#!bin/bash
for((i=1;i<23;i++))
do
qsub –cwd –q queue.q step1_chr${i}.sh
done
```

At last, execute step1_submit.sh.

```
$./step1_submit.sh
```

[2] Step 2

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1". Next, step2.sh which contains 100 jobs is submitted by step2_submit.sh. The step2.sh is described as follows.

```
#!bin/bash
R_Path/R – no-save << EOF
Library(FastEPRR)
FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", jobN
 umber = 100, currJob = ${1}, DXOutputFolderPath = "exampleData
 _Path/step2")
EOF
```

The step2_submit.sh is described as follows.

```
#!bin/bash
for((i=1;i<101;i++))
do
qsub –cwd –q queue.q step2_submit.sh ${i}
done
```

At last, execute step2_submit.sh.

```
$./step2_submit.sh
```

[3] Step 3

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1" and output files of step 2 have been saved in "exampleData_Path/step2". The step3.sh is described as follows.

```
#!bin/bash
R_Path/R – no-save << EOF
Library(FastEPRR)
FastEPRR_VCF_step3(srcFolderPath = "exampleData_Path/step1", DXF
 olderPath = "/exampleData_Path/step2", finalOutputFolderPath = "e
 xampleData_Path/step3")
EOF
```

Now, submit step3.sh.

```
$./step3.sh
```

Variable recombination rates within windows

[1] Batch mode

One R file is created for each chromosome, for example, var_chr1.R.

```
Library(FastEPRR)
FastEPRR_VAR(varFilePath = "exampleData_Path/step3/chr1", winLeng
 th = 50000, stepLength = 25000, gapFilePath = "exampleData_Path
 /chr1_gap_example.txt", varOutputFilePath = "exampleData_Path/var
 Rate/chr1")
```

Then, type the following command in the command line to run the example.

```
$R CMD BATCH var_chr1.R
```

[2] Within R console

Type "R" in the command line to open the R console. Then in the R console, the user can type the following commands.

```
>Library(FastEPRR)
> FastEPRR_VAR(varFilePath = "exampleData_Path/step3/chr1", winLe
 ngth = 50000, stepLength = 25000, gapFilePath = "exampleData_P
 ath/chr1_gap_example.txt", varOutputFilePath = "exampleData_Path
 /varRate/chr1")
```

[3] Bash script mode

First, the following commands should be written in a file, for example, var_example.sh.

```
#!bin/bash
R_Path/R – no-save << EOF
Library(FastEPRR)
FastEPRR_VAR(varFilePath = "exampleData_Path/step3/chr1", winLeng
 th = 50000, stepLength = 25000, gapFilePath = "exampleData_Path
 /chr1_gap_example.txt", varOutputFilePath = "exampleData_Path/var
 Rate/chr1")
EOF
```

Then, submit var_example.sh.

```
qsub –cwd –q queue.q var_example.sh
```

Please refer to the introduction of "qsub" or other command line used to submit jobs on your system for details.

Please do not input any space before or after "=" when write your *.sh file when using this mode.

Windows

There are two ways to run functions in FastEPRR package on Windows: batch mode and within R mode. For batch mode, R provides a standard command line (R CMD BATCH) for running R commands. For within R console, the R software must be started first.

The following subsections describe these two modes for alignment file and VCF file, respectively.

ALN

[1] Within R mode

Type "R" in the command line to open the R console. Then in the R console, the user can type the following commands.

```
>Library(FastEPRR)
>FastEPRR_ALN(alnFilePath = "exampleData_Path/alignmentExampleFil
 e.fas", fileFormat = 1, outputFilePath = "exampleData_Path/alignme
 ntExampleFileRecRate.txt", winLength = 100, stepLength = 50)
```

[2] Batch mode

First, the following commands should be written in a file, for example, batchExample_ALN.R

```
Library(FastEPRR)
FastEPRR_ALN(alnFilePath = "exampleData_Path/alignmentExampleFile.
 fas", fileFormat = 1, outputFilePath = "exampleData_Path/alignmen
 tExampleFileRecRate.txt", winLength = 100, stepLength = 50)
```

Click

```
"Start" ⟶ "All Programs" ⟶ "Accessories" ⟶ "Run"
```

to open the Run box, and then change to the R file directory. Type the following command in your Run box to run this example.

```
>R CMD BATCH batchExample_ALN.R
```

VCF

[1] Within R console

[1] Step 1

Type "R" in the command line to open the R console. Then in the R console, the user can type the following commands.

```
>Library(FastEPRR)
>FastEPRR_VCF_step1(vcfFilePath = "exampleData_Path/ALL.chr1.omni
 _example.vcf.gz", winLength = 50000, winDXThreshold = 10, gapFile
 Path = "exampleData_Path/chr1_gap_example.txt", srcOutputFilePath
  = "exampleData_Path/step1/chr1")
```

[2] Step 2

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1". Type "R" in the command line can open the R console. Then in the R console, the user can type the following commands.

```
>Library(FastEPRR)
>FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", job
 Number = 1, currJob = 1, DXOutputFolderPath = "/exampleData_P
 ath/step2")
```

[3] Step 3

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1" and output files of step 2 have been saved in "exampleData_Path/step2". Type "R" in the command line can open the R console. Then in the R console, the user can type the following commands.

```
Library(FastEPRR)
FastEPRR_VCF_step3(srcFolderPath = "exampleData_Path/step1", DXF
 olderPath = "exampleData_Path/step2", finalOutputFolderPath = "ex
 ampleData_Path/step3")
```

[2] Batch mode

[1] Step 1

One R file is created for each chromosome, for example, step1_chr1.R.

```
Library(FastEPRR)
FastEPRR_VCF_step1(vcfFilePath = "exampleData_Path/ALL.chr1.omni_
 example.vcf.gz", winLength = 50000, winDXThreshold = 10, gapFile
 Path = "exampleData_Path/chr1_gap_example.txt", srcOutputFilePath
  = "exampleData_Path/step1/chr1")
```

Then, type the following command in the command line to run the example.

```
$R CMD BATCH step1_chr1.R
```

Of course, command lines for multiple chromosomes can be written in one R file, for example, step1_total.R.

```
Library(FastEPRR)
FastEPRR_VCF_step1(vcfFilePath = "exampleData_Path/ALL.chr1.omni_
 example.vcf.gz", winLength = 50000, winDXThreshold = 10, gapFile
 Path = "exampleData_Path/chr1_gap_example.txt", srcOutputFilePath
  = "exampleData_Path/step1/chr1")

…

FastEPRR_VCF_step1(vcfFilePath="exampleData_Path/ALL.chr22.omni_e
 xample.vcf.gz", winLength=50000, winDXThreshold=10, gapFilePath="
 exampleData_Path/chr22_gap_example.txt", srcOutputFilePath="exam
 pleData_Path/step1/chr22")
```

Click

```
"Start" ⟶ "All Programs" ⟶ "Accessories" ⟶ "Run"
```

to open the Run box, and then change to the R directory. Type the following command in your Run box to run this example.

```
>R CMD BATCH step1_total.R
```

[2] Step 2

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1". The step2.R is described as follows.

```
Library(FastEPRR)
FastEPRR_VCF_step2(srcFolderPath = "exampleData_Path/step1", jobN
 umber = 1, currJob = 1, DXOutputFolderPath = "exampleData_Path
 /step2")
```

Then, open the Run box and type the following command in the command line to run the example.

```
>R CMD BATCH step2.R
```

[3] Step 3

We assume that output files of step 1 have been saved in "*exampleData_Path*/step1" and output files of step 2 have been saved in "/exampleData_Path/step2". The step3.R is described as follows.

```
Library(FastEPRR)
FastEPRR_VCF_step3(srcFolderPath = "exampleData_Path/step1", DXF
 olderPath = "exampleData_Path/step2", finalOutputFolderPath = "ex
 ampleData_Path/step3")
```

Then, open the Run box and type the following command in the command line to run the example.

```
>R CMD BATCH step3.R
```

Variable recombination rates within windows

[1] Batch mode

One R file is created for each chromosome, for example, var_chr1.R.

```
Library(FastEPRR)
FastEPRR_VAR(varFilePath = "exampleData_Path/step3/chr1", winLeng
 th = 50000, stepLength = 25000, gapFilePath = "exampleData_Path
 /chr1_gap_example.txt", varOutputFilePath = "exampleData_Path/var
 Rate/chr1")
```

Click

"Start" ⟶ "All Programs" ⟶ "Accessories" ⟶ "Run"

to open the Run box, and then change to the R directory. Type the following command in your Run box to run this example.

```
>R CMD BATCH step1_total.R
```

[2] Within R console

Type “R” in the command line to open the R console. Then in the R console, the user can type the following commands.

```
>Library(FastEPRR)
> FastEPRR_VAR(varFilePath = "exampleData_Path/step3/chr1", winLe
 ngth = 50000, stepLength = 25000, gapFilePath = "exampleData_P
 ath/chr1_gap_example.txt", varOutputFilePath = "exampleData_Path
 /varRate/chr1")
```

Reference

Altshuler DM, Durbin RM, Abecasis GR, Bentley DR, Chakravarti A, Clark AG, Donnelly P, Eichler EE, Flicek P, Gabriel SB, et al. 2012. An integrated map of genetic variation from 1,092 human genomes. *Nature* 491:56-65.

Buhlmann P, Hothorn T. 2007. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science* 22:477-505.

Charif D and Lobry JR. 2007. SeqinR 1.0 2: A Contributed Package to the R Project for Statistical Computing Devoted to Biological Sequences Retrieval and Analysis. Structural Approaches to Sequence Evolution. *Biological and Medical Physics, Biomedical Engineering* pp207-232

Gao F, Ming C, Hu WJ, Li HP. 2016. New Software for the Fast Estimation of Population Recombination Rates (FastEPRR) in the Genomic Era. *G3-Genes Genomes Genetics* 6:1563-1571.

Hudson RR. 2002. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics* 18:337-338.

Lin K, Futschik A, Li HP. 2013. A Fast Estimate for the Population Recombination Rate Based on Regression. *Genetics* 194:473-484.