FitCoal: a Fast Estimator for Population Demographic History Inference

Version 1.1

Contents

Introduction

This manual describes how to use FitCoal, a fast estimator for population demographic history inference. The fast infinitesimal time coalescent (FitCoal) process has been developed to allow the accurate calculation of the expected branch lengths under arbitrary demographic histories. Computational accuracy reaches 10^{-8} or 10^{-11}. Thus the new method allows the accurate calculation of the composite likelihood of a site frequency spectrum and provides the precise inference of recent and ancient demographic history. If you have any questions, please feel free to contact us: haoziqian@sdfmu.edu.cn, huwangjie@picb.ac.cn or lihaipeng@sinh.ac.cn.

Copyright

How to cite

Please cite FitCoal as following:

Wangjie Hu#, Ziqian Hao#, Pengyuan Du, Fabio Di Vincenzo, Giorgio Manzi, Jialong

Cui, Yun-Xin Fu, Yi-Hsuan Pan*, and Haipeng Li* (2023) Genomic inference of a severe human bottleneck at the Early to Middle Pleistocene transition. (submitted)

#These authors contributed equally.

*Corresponding Authors: yxpan@sat.ecnu.edu.cn; lihaipeng@sinh.ac.cn

Download and Installation

FitCoal has been developed as a free plug-in of eGPS software[1] while FitCoal can also be downloaded and run as an independent Java package. If you are familiar with DOS/Linux or need batch processing, you can run FitCoal on command lines. You need to install Java environment first. The recommended JAVA version is 1.8 or the latest. Users can visit https://www.java.com/ to download the JAVA for free.

Please note that the openJDK is not supported.

If you need Graphical User Interface (GUI), please download and install the eGPS software. Then copy the FitCoal to the eGPS plug-in directory (…/egps/config/plugin) and restart eGPS. You can find FitCoal under the plug-in menu. Please consult next sections for the details.

FitCoal and documentation are available via Zenodo at https://zenodo.org/record/4805461#.YK4HKLcza6I, our institute website at http://www.picb.ac.cn/evolgen/, and eGPS website http://www.egps-software.net/.

The basic command line

```
java -cp FitCoal.jar FitCoal.calculate.SinglePopDecoder -table tables/ -input example/constant.model.sfs.txt -output example/constant.model.sfs.output -mutationRate 0.000012 -generationTime 24 -genomeLength 10000
```

We use this command line to estimate demographic history of a SFS simulated under a constant population size model. The path of table file is example/tables, the path of input file is example/constant.model.sfs.txt, the path and prefix of output file is example/constant.model.sfs.output, mutation rate is 0.000012 per **kb per generation**, generation time is 24 years, and the considered genome length is 10,000 **kb**. You can get the input file and output results from our example data.

Simple output result

We then describe the results obtained from the command line above. FitCoal outputs results to standard output and output file. You can get detailed information of estimate from standard output, and plot results using output file.

1. Standard output (shown on the screen)

Data:1

numOfIntervals = 1

standardTime: 0.0 generations: 0.00 N: 10,144 changeTypes: 0

maxLogL = -153.97823167015702

numOfIntervals = 2

standardTime: 0.0 generations: 0.00 N: 68,452 changeTypes: 2

standardTime: 0.0018700230593434275 generations: 84.61 N: 10,066 changeTypes: 0

maxLogL = -152.33298768907252

******** Best: ********

numOfIntervals = 1

standardTime: 0.0 generations: 0.00 N: 10,144 changeTypes: 0

maxLogL = -153.97823167015702

computing time = 12.94 seconds

2. Output file

Results of output file correspond to the "Best result" in standard output. There are population sizes corresponding to time in years. Demographic events are traced **BACKWARD** in time.

year popSize
2000 10144
4000000 10144

SFS in an input file

There are two modes in FitCoal: normal mode and missing data mode. Normal mode is the most common used while missing data mode is designed for analyzing data set with missing genotypes. FitCoal distinguishes them by content of input file automatically.

1. Normal mode

The input file of normal mode should contain one or several SFSs. One SFS corresponds to one population. Each SFS should be written as a single line. SFS types should be split by space or tab. A mutation is said to be of type i if it is exactly appeared in i samples. The SFS is arranged from type 1 to type $n-1$, where n is sample size. There may be more than one SFS in the input file, and FitCoal estimates demographic histories independently because the SFSs are summarized from different populations.

Examples:

An input file containing one SFS of a population:

4979 2392 1644 1221 973 750 709 653 535 483 448 380 341 336 331 271 248 234 249 311 251 233 238 168 168 233 187 182 141

An input file containing three SFSs of three populations or simulations:

4979 2392 1644 1221 973 750 709 653 535 483 448 380 341 336 331 271 248 234 249 311 251 233 238 168 168 233 187 182 141

4818 2400 1743 1086 923 976 651 692 610 535 505 485 424 303 331 308 324 330 287 292 226 245 225 248 179 176 178 173 169

4791 2258 1597 1270 1014 794 733 525 495 480 466 387 384 334 284 308 299 282 257 194 208 234 197 187 201 172 186 160 179

2. Missing data mode

The input file of missing data mode should contain SFSs of ***a single population***. Each SFS should be written as a single line. The first element of a line should be the tag of "SampleSize".The second element should be the sample size of the SFS. The SFS should be written as follow. FitCoal estimates only **ONCE** because all SFSs are summarized from the same population.

Examples:

An input file containing a SFS calculated from sites with no missing data and a SFS summarized from sites with one missing individual of the same population having 16 individuals:

SampleSize 32 518376 209377 144169 115180 98507 84557 75097 69570 63383 56348 53520 50799 47346 44956 41820 40142 37601 36006 34355 33902 30965 30116 29534 28828 28025 27458 27669 28157 30334 32903 40672

SampleSize 30 21876 10758 8343 7376 7032 6601 6261 5992 5587 5333 5148 4887 4738 4429 4165 4136 3695 3643 3352 3196 2880 2765 2566 2346 2263 2010 1912 1890 2086

Estimate demographic history

Users can estimate demographic history using this one-step process. There are six parameters for users to provide. Other parameters are optional.

Parameter	Type	Necessary	Description
table	String	Yes	Path of the directory containing table files. Table will be established automatically during your first estimate, and can be reused for later estimates. .
input	String	Yes	Path of the input file containing SFS data.
output	String	Yes	Path and prefix of the output file. Suffix is set to be ".txt".
mutationRate	double	Yes	Mutation rate PER KB per generation.
generationTime	double	Yes	Generation time in years.
genomeLength	double	Yes	Length of genome IN KB.
noIG	boolean	No	To set whether instantaneous population growth is allowed. If true, there will be no instantaneous population growth. Default is false.
noID	boolean	No	To set whether instantaneous population decline is allowed. If true, there will be no instantaneous population decline. Default is false.
noEG	boolean	No	To set whether exponential population growth is allowed. If true, there will be no exponential population growth.

			Default is false.
noED	boolean	No	To set whether exponential population decline is allowed. If true, there will be no exponential population decline. Default is false.
numOfIntervals	int	No	Fixed number of time intervals. If it is specified, FitCoal will only estimate results with the given number of time intervals.
logLPRate	double	No	Log-likelihood promotion rate (%). This parameter determines when to stop the iteration. Default is 20, which is recommended for the most users.
repeats	int	No	Number of independent repeats to estimate the maximum likelihood. Default is 100. Larger value means better accuracy but consumes more time.
omitStartSFS	int	No	Number of omitted start types of SFS. Default is 0. Rare mutations can be removed using this option, such as singletons (omitStartSFS = 1). If you use missing data mode, this value should correspond to the number of omitted start types of the FIRST SFS. FitCoal calculates the ratio and use this ratio to truncate other SFSs.
omitEndSFS	int	No	Number of omitted end types of SFS. Default is 0. High-frequency mutations

			can be removed using this option. If omitEndSFS = 2, the two mutation types (n-2, and n-1) will be discarded. If you use missing data mode, this value should correspond to the number of omitted start types of the FIRST SFS. FitCoal calculates the ratio and use this ratio to truncate other SFSs.
randSeed	int	No	Seed of random number.
timeUpperBound	double	No	Upper bound of searched standard coalescent time. Default is 2.0. Users may slightly increase this value if the upper bound of estimated history is very close 2.0. The increased timeUpperBound should remain smaller than 4.0.
gammaTimeLUS	double	No	Gamma parameter of LUS[2] when sampling time parameters. Default is 600. Larger value means better accuracy and consumes more time. It is recommended to increase this value when timeUpperBound is increased.
gammaPopSizeLUS	double	No	Gamma parameter of LUS[2] when sampling population size parameters. Default is 600. Larger value means better accuracy and consumes more time. It is recommended to increase this value when timeUpperBound is increased.

minChangePerPhase	double	No	Minimum population size change rate per time interval. Default is 0.01.
maxChangePerPhase	double	No	Maximum population size change rate per time interval. Default is 100.
minRatioOverN0	double	No	Minimum ratio of population size over N0, where N0 is the current effective population size. Default value is 0.001.
maxRatioOverN0	double	No	Maximum ratio of population size over N0, where N0 is the current effective population size. Default value is 1000.
foldedSFS	boolean	No	The parameter describes whether the SFS is folded or unfolded. Usually, an unfolded SFS is obtained when the outgroup is available, and the length of SFS is $(n-1)$. In this case, the ancestral and derived alleles for each mutation are inferred. Default is false.
collapsedEndSFS	int	No	Collapsing the last (collapsedEndSFS) SFS categories into one class. Default is 0.
Plot parameters			
yearMax	int	No	The maximum year you want to plot. It is not relevant to demographic history inference, but the output. Default value is 4,000,000 years.
yearMin	int	No	The minimum year you want to plot ($\geq$ 1). It is not relevant to demographic history inference, but the output. When plotting the demography, the time is

	often log-scaled, and the time value must be larger than 0. Default value is 2,000 years.

Note:

It is relatively time-consuming to infer exponential change of population size. If users want to run the program for teaching, or to test the system, please use “noEG” and “noED” option, and a small “repeats”. Users should be able to get the results in seconds.

More examples

All input files and output results of command lines below can be found in our example data.

1. To estimate demographic histories of three SFSs simulated under constant size models:

```
java -cp FitCoal.jar FitCoal.calculate.SinglePopDecoder -table tables/ -input example/constant.model.3sfs.txt -output example/constant.model.3sfs.output -mutationRate 0.000012 -generationTime 24 -genomeLength 10000
```

2. To estimate demographic history of a SFS, which is simulated under PSMC standard model, conditional on instantaneous population size change and a small “repeats”:

```
java -cp FitCoal.jar FitCoal.calculate.SinglePopDecoder -table tables/ -input example/PSMC.model.sfs.txt -output example/PSMC.model.sfs.output -mutationRate 0.000012 -generationTime 24 -genomeLength 30000 -repeats 10 -noEG -noED
```

3. To estimate demographic history of CHB population in 1000GP using truncated SFS:

```
java -cp FitCoal.jar FitCoal.calculate.SinglePopDecoder -table tables/ -input example/CHB.sfs.txt -output example/CHB.test1.output -mutationRate 0.000012 -generationTime 24 -genomeLength 826650 -omitEndSFS 26
```

4. To estimate demographic history of CHB population in 1000GP using truncated SFS conditional on instantaneous population size change:

```
java -cp FitCoal.jar FitCoal.calculate.SinglePopDecoder -table tables/ -input example/CHB.sfs.txt -output example/CHB.test2.output -mutationRate 0.000012 -generationTime 24 -genomeLength 826650 -omitEndSFS 26 -noEG -noED
```

5. To estimate demographic history of CHB population in 1000GP using truncated SFS conditional on instantaneous population size change and three time intervals:

```
java -cp FitCoal.jar FitCoal.calculate.SinglePopDecoder -table tables/ -input example/CHB.sfs.txt -output example/CHB.test3.output -mutationRate 0.000012 -generationTime 24 -genomeLength 826650 -omitEndSFS 26 -noEG -noED -numOfIntervals 3
```

6. To estimate demographic history of YRI population in 1000GP using truncated SFS:

```
java -cp FitCoal.jar FitCoal.calculate.SinglePopDecoder -table tables/ -input example/YRI.sfs.txt -output example/YRI.test.output -mutationRate 0.000012 -generationTime 24 -genomeLength 826650 -omitEndSFS 27
```

7. To estimate demographic history of Adygei population in HGDP-CEPH using truncated SFS conditional on instantaneous population size change:

```
java -cp FitCoal.jar FitCoal.calculate.SinglePopDecoder -table tables/ -input example/Adygei.sfs.txt -output example/Adygei.sfs.output -mutationRate 0.000012 -generationTime 24 -genomeLength 791999 -omitEndSFS 4 -noEG -noED
```

How to truncate SFS

In this study, we developed a sliding-window strategy to determine where to truncate SFS (see paper for details). It is suitable for human data. FitCoal outputs the index of truncated types, and the percentage of truncated types.

Parameter	Type	Necessary	Description
input	String	Yes	Path of the input file containing SFS data. File should only contain one or several SFSs without any other tags.

Examples:

The following command provides the suggested index of SFS type where SFS could be truncated.

```
java -cp FitCoal.jar FitCoal.calculate.TruncateSFS -input example/CHB.sfs.txt
```

Convert demographic history to *ms* command style

Users can transform demographic parameters of FitCoal to *ms* format using class “ConvertCode” in FitCoal.

Parameter	Type	Necessary	Description
input	String	Yes	Path of the input file containing demographic parameters. The input file should contain information printed in standard output when estimating demographic history.

Examples:

The following command transforms the parameters of CHB.test1 to *ms* command style, which can be used to run *ms* simulations.

```
java -cp FitCoal.jar FitCoal.calculate.ConvertCode -input example/CHB.test1.stdput.txt
```

Number of lineages at time *t*

Users can calculate the expected number of lineages at time t. There are three parameters for users to provide. An example is provided as:

```
java -cp FitCoal.jar FitCoal.calculate.NumOfLineages -table YourPath -n 10 -t 0.5
```

Parameter	Type	Necessary	Description
table	String	Yes	Path of the directory containing table files. Table will be established automatically during your first estimate, and can be reused for later estimates. .
n	integer	Yes	The sample size, or the number of sampled chromosomes.
t	double	Yes	The time t was scaled by $2N(t)$ generations. To distinguish it from the one-point scaled time, time t was designated as the standard coalescent time.

Detailed output explaination

1. Standard output

For each estimate, FitCoal outputs the results of different time intervals and the best result to the standard output. Demographic events are traced **BACKWARD** in time.

Parameter	Description
Data	Estimate order.
numOfIntervals	Number of time intervals.
standardTime	Standard coalescent time.
generations	Time in generations.
N	Effective population size.
changeType	Three types of population size change.
	0: the last time interval
	1: instantaneous population size change
	2: exponential population size change
fitness	Minus log likelihood.
computing time	Consumed time measured in millisecond

Examples:

The standard output of an estimate with three time intervals:

Data:1

numOfIntervals = 1

standardTime: 0.0 generations: 0.00 N: 13,830 changeTypes: 0

maxLogL = -305205.24268867826

numOfIntervals = 2

standardTime: 0.0 generations: 0.00 N: 1,134,141 changeTypes: 2

standardTime: 0.0037026853020623402 generations: 390.68 N: 11,341 changeTypes: 0

maxLogL = -110971.15241060083

numOfIntervals = 3

standardTime: 0.0 generations: 0.00 N: 308,336 changeTypes: 2

standardTime: 0.012510110408205207 generations: 639.83 N: 6,482 changeTypes: 1

standardTime: 0.5671470843405646 generations: 7830.28 N: 19,803 changeTypes: 0

maxLogL = -2038.4889222141792

numOfIntervals = 4

standardTime: 0.0 generations: 0.00 N: 335,162 changeTypes: 2

standardTime: 0.009552954048616977 generations: 585.05 N: 8,003 changeTypes: 2

standardTime: 0.07003572416190967 generations: 1393.55 N: 5,639 changeTypes: 2

standardTime: 0.7982313105949165 generations: 15953.69 N: 20,281 changeTypes: 0

maxLogL = -1862.8073984939329

******** Best: ********

numOfIntervals = 3

standardTime: 0.0 generations: 0.00 N: 308,336 changeTypes: 2

standardTime: 0.012510110408205207 generations: 639.83 N: 6,482 changeTypes: 1

standardTime: 0.5671470843405646 generations: 7830.28 N: 19,803 changeTypes: 0

maxLogL = -2038.4889222141792

computing time = 5957.44 seconds

2. Output file

You can plot results using R language or other plot tools according to the results in output file. Output file contains time and population size of the "Best result" in standard output. There are two columns. We use two rows to record instantaneous change and 100 rows to record exponential change (see example file "CHB.test1.output.txt"). Therefore, the number of rows may be different in different estimates.

Parameter	Description
year	Time in years.
popSize	Effective population size.

Example:

An output file recording a demographic history with two instantaneous change:

year	popSize
2000	105774
11813	105774
11813	6970
212448	6970
212448	20738
4000000	20738

The simplest way to plot results using R language:

You can adjust the command line below according to your needs.

```
data = read.table("CHB.test1.output.txt",header = T)
plot(log10(data$year), log10(data$popSize),type="l")
```

In the future, the eGPS software may provide a visualization of estimated demography.

Update logs

FitCoal 1.1 (Jan 03, 2023)

1. A folded SFS was supported (although it is not recommended).
2. The last SFS categories could be collapsed into one class.
3. R-extended FitCoal was implemented to detect weak signals of the severe bottleneck in non-African populations.
4. The expected number of lineages at time *t* can be obtained.

References

1 Yu, D. L. *et al.* eGPS 1.0: comprehensive software for multi-omic and evolutionary analyses. *Natl. Sci. Rev.* **6**, 867-869 (2019).

2 Pedersen, M. E. H. Tuning and simplifying heuristical optimization. *PhD thesis, Univ. Southampton* (2010).