RIDGE User Manual

1. Summary

RIDGE is a tool designed to detect genomic regions resistant to gene flow between two populations. It uses an ABC approach to infer the demographic history of the two populations and in a second time to detect barriers. The program known as RIDGE requires significant computational resources and depends heavily on parallelization to decrease its computation time.. So it is highly recommended to install it on a cluster (and in a Linux environment). To manage the various programs and allow reproducibility, RIDGE uses the *Singularity* container technology, and *Snakemake* to manage the workflow.
As output, RIDGE provides for each locus (aka fragment of the genome) a probability of being a gene flow barrier.

2. Table of content

3. Installation (WIP)

3.1. Get the code

Download the code with the following command

```
git clone https://github.com/EwenBurban/RIDGE.git
```

3.2. Install containers

4. Getting Started

4.1. Gather all necessary information

Before any launch you must fill all mandatory input files. See Input & Outputs section for precisions.

4.2. Scan launch

To correctly fill your config files you will need a measure of the diversity. So first fill the field N_min and N_max with approximative values. Next you have to launch RIDGE in scan mode with the following command

```
bash <path to RIDGE>/RIDGE.sh <path to work folder>/config.yaml scan
```

RIDGE, will generate a file named ***ABCstat_locus.txt***. Open it with R and compute the quantile at 5 and 95% for piA_avg, piB_avg, thetaA_avg, thetaB_avg. The mean of a quantile across all the summary stats divided by 4μ (with the μ value from the config file) gives you a population size to use for N_min (quantile at 5%) and N_max (quantile at 95%).

4.3. Complete launch

Once N_min & N_max are correctly set, relaunch RIDGE, but this time with the whole process :

```
bash <path to RIDGE>/RIDGE.sh <path to work folder>/config.yaml all
```

5. RIDGE pipeline

The RIDGE approach is an ABC method to detect gene flow barriers by including the effect of the demography. To do so, RIDGE first infers an average demographic model including heterogeneity along the genome, based on the data provided; second for each locus it infers the probability of being a barrier.

5.1. Inferring average demographic history

To infer demographic history, RIDGE follows an ABC procedure. First, it generates summary statistics from the observed datasets. Second, it simulates datasets under a set of scenarios to produce a reference table using the same set of summary statistics as computed for the observed

data. The reference table is used by a machine learning algorithm as learning material to infer demographic parameters of the observed dataset. Please note that in order to avoid excessive computation time, it is strongly advised to set the option Nlocus=1000, or any other value that is lower than the total number of loci present in the dataset. Hence, within this chapter, the term "sampled loci" denotes the set of loci that have been randomly chosen to represent the entire genome.

5.1.1. Summary statistics

To summarize the multilocus dataset composed of the sampled loci, RIDGE first calculates the following locus specific summary statistics using the script ***vcf2abc.py***. :
(a) the number of SNPs, (b) two diversity indexes (**π** (Nei and Li, 1979) and Watterson **θ** (Watterson, 1975)) for each population (c) **Tajima's D** for each population (Tajima, 1989), (d) **Fst** using the method of Hudson (Hudson et al., 1992) elaborated by (Bhatia et al., 2013), (e) the absolute (**Dxy**) and net divergence (**Da**), (f) four statistics summarizing the joint Site Frequency Spectrum (**jsfs**): **ss** (the proportion of shared polymorphism), **sf** (the proportion of fixed differences), **sxA** and **sxB** (the proportion of specific polymorphism to each population).
Once that summary statistics are generated for each locus, the script ***vcf2abc.py*** calculates global summary statistics across loci: the mean, the standard deviation and the median is calculated for each previously described summary statistics. In addition RIDGE computes: (1) the Pearson correlation between the **π** of each population, between Watterson **θ** of each population, between Dxy and Da, between Dxy and Fst; between Da and Fst (2) the number of loci with specific jsfs status: ss_sf, the number of loci which contain shared polymorphism (so ss > 0) and fixed differences (sf >0); ss_NoSf, the number of loci which contain shared polymorphism and no fixed differences; noSs_sf, the number of loci which contain no shared polymorphism and fixed differences; noSs_noSf, the number of loci which contain neither shared polymorphism nor fixed differences.
All the global summary statistics are written in the file ***ABCstat_global.txt***. It is worth noting that locus specific summary statistics on sampled loci are not recorded.

5.1.2. Coalescent simulations of the reference table

RIDGE simulates multiple random datasets under 14 different models. Some parameters of the 14 each model are shared, others are specific to some models. Using 14 models, allows RIDGE to cover a large diversity of historic demography, and allows testing for heterogeneous migration and heterogeneous diversity due to gene flow barrier or local variations in Ne due to linked selection. RIDGE takes rho heterogeneity across loci into account. To simulate all datasets, RIDGE uses the program ***scrm***. ***scrm*** is a ms-like program used to simulate independent neutral loci. A dataset is composed of `Nlocus` independent loci with two levels of parameters : (1) Global demographic parameters, which define the average demographic history of the dataset ; (2) Parameters that define the heterogeneity among loci.

Demographic history parameters

The demographic models used in RIDGE are as follows : one ancestral population with effective size Na, splits into two daughter populations of size N1 and N2 at time Tsplit. From this, there are 4 different scenarios:

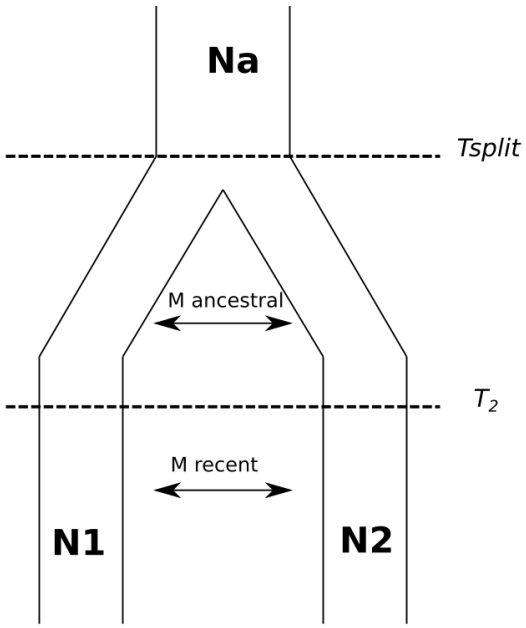
- Strict Isolation (SI): there is no migration between populations
- Isolation-Migration (IM): there is migration between populations
- Ancestral Migration (AM): there is migration until Tam. Tam cannot be inferior to 0.7 * Tsplit
- Secondary Contact (SC): there is no migration until Tsc. Tsc must be contained between 0.05 * Tsplit < Tsc < 0.3 Tsplit

Migration is symmetrical, so there is only one parameter per model. In the AM model, the migration is considered ancestral and defined by the M_ancestral parameter. In IM and SC models, migration is considered as actual/current and defined by M_current parameter.
In the end, there are 8 parameters, with 4 parameters shared by all models :

- Na, the size of the ancestral population in Ne unit
- N1, the size of one of the two daughters populations in Ne unit
- N2, same as N1
- Tsplit, the time of split into two daughter populations in generations units?

and 4 parameters specific to some models :

-
- Tam, the time of arrest of ancestral migration in generation units
- Tsc, the time of start of current migration in generation units
- M_current, the rate of migration between the two daughter populations in N*m units
- M_ancestral, same as M_current but before Tam



SI	M ancestral = M recent = 0
IM	M ancestral = M recent > 0
SC	M ancestral = 0 ; M recent > 0 ; T2 : time of secondary contact
AM	M ancestral > 0 ; M recent = 0 ; T2 : end of ancestral contact

Genomic heterogeneity parameters

Heterogeneity in Ne and in migration among loci can be simulated. It mimics the effect of gene flow barriers (due to reproductive isolation) and linked selection. The combination of the two kinds of heterogeneity leads to four "genomic" scenario:

- Homogeneity of Ne and migration
- Heterogeneity of Ne and migration
- Homogeneity of migration and heterogeneity of Ne
- Heterogeneity of migration and homogeneity of Ne

Because there is no migration in SI models, there are only two genomic scenarios under strict isolation : SI_1N and SI_2N. Overall, there are 14 different scenarios.

In genome-wide homogeneous-Ne (homo-Ne) models, all loci share the same effective population size. In genome-wide heterogeneous-Ne (hetero-Ne) models, Ne distribution among loci follows a rescaled beta distribution with a mean equal to the defined value of Ne, and two shape parameters that are the same for all three populations.

$$N_i = N * \frac{\beta(\alpha,\beta)}{(\alpha/(\alpha+\beta))} \qquad (1)$$

In heterogeneous N.m models, N.m distribution follows a bimodal distribution with a proportion loci, *P_barrier*, have N.m=0 and a proportion 1-P_barrier have N.m=M (M_current or M_ancestral depending of the demographic model).

$$M_i = M * B(1, 1 - P_{barrier}) \qquad (2)$$

In the end, there is 3 parameter to regulate the genomic heterogeneity :

- α and β from the beta distribution of Ne
- P_barrier for the heterogeneity of migration

Parameter sampling

each value of each parameter is independently sampled in a distribution bounded by hyperpriors defined by the user. So :

- Na, N1 and N2 are independently sampled in uniform distributions bounded by `N_min` and `N_max`
- Tsplit is sampled in a uniform distribution bounded by `Tsplit_min` and `Tsplit_max`
- M_current and M_ancestral are sampled independently in an **log**-uniform distribution bounded between `M_min` and `M_max`
- α and β are sample in a uniform distribution between [0.1,10]
- P_barrier is sampled in an uniform distribution between 0 and `Pbarrier_max`

5.1.3. Estimate demographic parameters

All simulated dataset constitute a **reference table** where the true parameter used to generate simulation and the summary statistics of the simulation are gathered. RIDGE uses the ***regAbcrf*** function of the package ***abc-rf*** which performs a regression based on a random forest algorithm. It uses the reference table as a learning dataset and estimates each parameter independently. *In fine* RIDGE produces point estimates and posterior distributions (?) of all parameters and , which are used as priors for the next locus-scale simulations.

Reference table construction

The **reference table** is made of **priors** used to generate simulations and the **summary statistics** that describe the simulation. Because all datasets do not have the same number of parameters (*e.g* an SI model has no parameter of migration, whereas an IM model has one), RIDGE uses a hypermodel that integrates all demographic scenarios. So, to build the **reference table,** every dataset's parameters set are automatically completed to fit in the hypermodel using the following completion tables.

Demographic scenario / Missing parameter	SI	IM	SC	AM
Tsc	0	R*Tsplit		0
Tam	Tsplit	R*Tsplit	Tsplit	
M_current	0			0
M_ancestral	0	M_current	0	

Table 1: Completion of global demographic parameters when they are missing for each demographic scenario. 0 means that the parameter is not included in the model but its value is set to 0 for comparison with other models. When the name of a parameter (e.g Tsplit) is given, it mean the value taken by the missing parameter is equal to the value of the mentioned parameter.

R is a random value drawn from an uniform distribution bounded to [0,1]. Note that for one dataset, R value is drawn only once, so it means that Tsc=Tam=R*Tsplit.

Missing parameters / Demographic scenario	shape_N_a & b	PbarrierM_current	PbarrierM_ancestral
SI	1000	0	0
IM homo M	1000	0	0
IM hetero M	1000		PbarrierM_current
SC	1000	0	0
AM	1000	0	0

Table 2: Completion of genomic heterogeneity parameters when they are missing for each demographic-genomic scenario. 0 means that the parameter is not included in the model but its value is set to 0 for comparison with other models. When the name of a parameter (e.g Tsplit) is given, it mean the value taken by the missing parameter is equal to the value of the mentioned parameter. Note, here the IM scenario is mentioned twice, IM homo M means: all IM scenarios with homogeneous migration (IM homo M & hetero Ne and IM homo M & homo Ne)

Parameter estimation and posteriors generation

Each parameter is estimated independently by a random forest built on the reference table. **regAbcrf** function produces an output containing the predicted value and a weight for each dataset in the reference table. Predicted values, put together form a point estimation which is written in ***point_posterior.txt*** . So for each parameter, each dataset received a different weight. RIDGE calculates for each dataset the average of all parameter weights, $\overline{W}$. Next, the $\overline{W}$ are used as sampling probability to sample parameter sets (ensemble of parameters) used to generate the reference table which constitute the posterior written in ***posterior.txt*** file.

5.1.4. Quality control of demographic estimation

To estimate the goodness of fit of posterior, we used the same method as in (Lemaire et al., 2016). The goodness of fit is defined as p_D the proportion of posterior dataset with a higher mean euclidean distance to the other datasets than the observed dataset. For this we calculate euclidean distance on the normalized summary statistics of the datasets obtained by simulation of the posteriors.

5.2. Detection of gene flow barriers

Posterior generated during demographic inference (contained in ***posterior.txt***) are used as prior of simulations, but for the step, RIDGE produces summary statistics for each loci (contained in ***ABCstat_locus.txt***). It allows to build a reference table per locus (whereas in the previous step it was a multilocus dataset). Half of the simulated loci in the reference table are gene flow barriers (i.e the migration is equal to 0), and the rest experiences the migration corresponding to the average demographic model (i.e non-barrier). A random forest algorithm (**abcrf** function from **abcrf R** package) is feeded by the reference table and trained to detect gene flow barriers. Next the random forest predicts for each locus from the observed dataset, the most probable model ("barrier" or "non-barrier" model) and an estimation of the posterior probability of the selected model. Because there are only two models, the posterior probabilities verify: $P[m1] = 1 - P[$ m2]. So, it allows us to produce the posterior probability of the gene flow barrier model B for each locus.

Based on previously generated posterior, we calculate a Bayes factor BF for each i loci as :

$$BF_i = \frac{P(m_i=m1|P_n)}{P(m_i=m2|(1-P_n)} = \left(\frac{1}{N}\sum_{n=1}^{N}\frac{P_n}{1-P_n}\right) * \frac{B_i}{1-B_i}$$

Where P_n is the proportion of barrier of the posterior number n. The Bayes factor can be used to determine which loci can be considered as a barrier. By default, a loci with BF>1 is considered as a barrier. The posterior probabilities and the BF for each locus are written in ***Pbarrier.txt***. The proportion of gene flow barriers detected by BF>1 and by random forest are written in ***report_barrier_detection.txt***.

6. Inputs & Outputs

6.1. Input files

To launch RIDGE, you need to provide at least 4 files **in your work folder** (i.e the folder where RIDGE will work and generate output).

So your work folder must follow the following configuration before any launch :

timeStamp/
├─ vcf_file
├─ contig_data.txt
├─ popfile.csv
├─ config.yaml
├─ optional : rec_rate_map

6.1.1. Vcf file

In the actual version, RIDGE only takes as genomic polymorphism data a vcf file (vcf file format >= 4.0). RIDGE can manage haploid and diploid data . The vcf file must contain only biallelic sites.

6.1.2. Popfile

This file lists the individuals from each population using the csv format (with ‘,’ as separator). The name of each population must be present in the header. The Popfile must contain at least two populations (so two columns) and each list must be the same length as the others. If the populations are not of the same length, you can fill missing individuals with “**NA”**.

Exemple of file:

```
wild,dom
W1,Dx1
W3,Dx10
W5,NA
W7,Dx11
```

6.1.3. Contig data file

A file that contains the length of each chromosome/contig and their related names and order.

- **contig_name**: is the name of the chromosome/contig in the vcf file
- **contig_length**: the length in bp of the contig
- **index**: the index in the order of contigs
- **contig_name_gff**: the name of the contig in the gff file, if there is the use for a gff file (only with gwscan mode)

Exemple of file:

contig_name	contig_length	index	contig_name_gff
Chr1	43270923	1	chr01
Chr2	35937250	2	chr02
Chr3	36413819	3	chr03
Chr4	35502694	4	chr04
Chr5	29958434	5	chr05
Chr6	31248787	6	chr06
Chr7	29697621	7	chr07
Chr8	28443022	8	chr08
Chr9	23012720	9	chr09

6.1.4. Recombination rate data

RIDGE uses either a recombination map or a constant recombination rate to work.
If you choose to use a constant recombination rate, you must set **homo_rec : True** in the config file and fill the field **homo_rec_rate** with the mean recombination rate estimated for your dataset. Otherwise you need to fill the field **rec_rate_map** with the name of your recombination map and place it in the work folder. Note that the recombination rate **r** must be the number of recombination per and the recombination map uses the tabulation as a separator.

- chr : the index of the contig (cf contig file)
- start and stop : the beginning and ending in bp of the window
- r : the recombination rate inside the window

Example of file:

chr	start	end	r
9	21800000.0	21900000.0	7.170443918444081e-07
9	21900000.0	22000000.0	6.771961140602021e-07
9	22000000.0	22100000.0	6.44356192372138e-07
9	22100000.0	22200000.0	6.08745943319314e-07
9	22200000.0	22300000.0	5.709059200375581e-07

6.1.5. Config.yaml file

The config file contains all the data to start RIDGE. Note, that in this file, the hyperpriors used in ABC process are defined, and so, an error in the hyperpriors can drastically affect the performances and results of RIDGE
The field of the file are the followings :

- config_yaml : the name of the config.yaml file that you are actually filling. (Note that you do not need to give the absolute path, but only the filename, otherwise it will stop)
- Vcf_file : the name of vcf file (only filename expected)
- Contig_data : the name of the contig data file (only filename expected)
- Rec_rate_map : the name of the recombination map (only filename expected or NA if no map)
- Popfile : the name of the popfile (only filename expected)
- nameA and nameB : name of one of the two populations. The names must be the same as used in popfile
- Container_path : the absolute path to the container folder, which contain all the singularity container, and so all programs
- Ploidy : the level of ploidy of the dataset. 1 is for haploid, 2 is for diploid
- ligthMode: Activate the ligthMode, which is a faster but less precise version of RIDGE (WIP)
- timeStamp: absolute path to the work folder
- Nlocus_per_chr : number of locus sampled per chromosome, used to avoid unnecessary computational time. If nlocus_per_chr is set to **-1**, all the genome will be used in the process (it may slow down the process by 10 to 100 times, depending on the size of the

dataset). Note that a total number of loci (nb of chr * nlocus_per_chr) around 1000 loci is a good trade off

- Window_size : the size in bp of each locus. Choose the value, depending on the snp density in your data.
- Homo_rec : If True, set an homogeneous recombination rate along the genome, False it use the provided recombination map
- Homo_rec_rate : provide the recombination rate value to use along the genome
- Mu : provide the mutation rate to use for all the genome
- Nref : the population size of reference, used to rescale all values in coalescent unit.
- N_min & N_max : minimum and population size in Ne value. Note that it**is highly recommended to set the value on the basis of the real diversity value in the vcf rather than expected value from the literature** (to have a good estimation you can launch ridge in scan mode and compute the mean of the diversity see Getting started section).
- M_min & M_max : minimum and maximum migration rate. By default M_min=0.1 and M_max=50
- Tsplit_min & Tsplit_max : minimum and maximum time of split of the ancestral population **in generation !!!**
- Pbarrier_max : maximum proportion of the genome under barrier to gene flow. By default Pbarrier_max=0.2

Example of file:

```
M_max: 50
M_min: 0.1
N_max: 200000
N_min: 10000
Nref: 50000
Tsplit_max: 20000
Tsplit_min: 1000
Pbarrier_max: 0.2
config_yaml: config.yaml
container_path: /home/dygap/eburban/RIDGE/container
contig_data: contig_data.txt
gff_file: tt.gff
lightMode: False
mu: 1e-8
nameA: wild
nameB: dom
popfile: popfile.csv
rec_rate_map: rho_map_african_rice.txt
timeStamp: /home/dygap/eburban/african_rice_v2
window_size: 10000
ploidy: 1
vcf_file: haplotyped.vcf
```

homo_rec: False
homo_rec_rate: NA
nlocus_per_chr: 100

6.2. Output files

6.2.1. Genome scan

Summary statistics are computed for each locus and stored in ***ABCstat_locus.txt*** .
Additional measures can be added using **NA (WIP)**
Below is described the head field of the ***ABCstat_locus.txt***
The basic set first :

- *bialsite* : number of bi allele sites in the locus
- *Fst* : Fst estimation using the method of Hudson (1992) elaborated by Bhatia et al. (2013).
- *divAB* : Estimate nucleotide divergence between two populations (dxy) within a given region, which is the average proportion of sites (including monomorphic sites not present in the data) that differ between randomly chosen pairs of chromosomes, one from each population.
- *netDivAB* : nucleotide divergence without ancestral polymorphism (Da). The ancestral polymorphism is assumed to be the mean of population polymorphism.
  $Da = Dxy - (\pi_A + \pi_B)/2$
- *piA , piB* : Estimated nucleotide diversity within a given region, which is the average proportion of sites (including monomorphic sites not present in the data) that differ between randomly chosen pairs of chromosomes.
- *thetaA , thetaB* : Estimate nucleotide diversity with Watterson theta
- *DtajA, DtajB* : Tajima's D
- *ss* : proportion of polymorphism shared between populations
- *sf* : proportion of polymorphism with fixed difference between populations
- *sxA, sxB* : proportion of polymorphism specific to a population

In ***ABCstat_global.txt*** additional summary stats are added to describe the ensemble of loci. The mean (_avg) , the standard error (_std) and the median (_median) is calculated for each previously described summary statistics. In addition new summary stats are computed

- pearson_r_pi : pearson correlation between piA and piB
- Pearson_r_theta : pearson correlation between thetaA and thetaB
- pearson_r_divAB_netdivAB : pearson correlation between divAB and netDivAB
- pearson_r_divAB_FST : pearson correlation between divAB and Fst
- pearson_r_netdivAB_FST : pearson correlation between netDivAB and Fst
- ss_sf : the number of loci which contains shared polymorphism (so ss > 0) and fixed differences (sf >0)
- ss_noSf : number of loci which contain shared polymorphism and no fixed differences

- noSs_sf : noSs_sf number of loci which contain no shared polymorphism and fixed differences
- noSs_noSf : noSs_noSf number of loci which contain neither shared polymorphism nor fixed differences
- fst_outlier : proportion of locus with an fst level higher than the maximal tukey fence based on fst distribution across locus
- divAB_outlier : proportion of locus with an Dxy (divAB) level higher than the maximal tukey fence based on Dxy distribution across locus
- netDivAB_outlier : proportion of locus with an Da (netDivAB) level higher than the maximal tukey fence based on Da distribution across locus
- sf_outlier : proportion of locus with an Df (sf) level higher than the maximal tukey fence based on Df distribution across locus
- piA_outlier : proportion of locus with an piA level lower than the minimal tukey's fence based on fst distribution across locus
- piB_outlier : proportion of locus with an piB level higher than the minimal tukey's fence based on piB distribution across locus

6.2.2. Demographic inference

The average demographic model is stored in ***point_posterior.txt*** file. You can also find all the posterior demographic models in the ***posterior.txt*** file.

6.2.3. Quality control

To check the quality of RIDGE estimation about the demographic inference and locus sampling (if you set **nlocus_per_chr** with an other value than -1), there is :

- Goodness of fit estimations
- Graphical representations in QC_plot folder

Goodness of fit outputs

The quality of fit of the prior and posterior distributions can be evaluated using gof files and principal component analysis (PCA). The gof files, namely ***gof_prior.txt*** and ***gof_posterior.txt***, determine the mean normalized Euclidean distance and p-value between the observed data set and the summary statistics of the prior or posterior simulations. The prior simulations are stored in the **modelComp** folder, while the posterior simulations are stored in the **sim_posterior** folder. The p-value of gof represents how well the observed data set aligns with the prior or posterior distributions, while the distance value indicates the similarity between the observed data set and the prior or posterior distributions. A p-value greater than 0.05 (e.g., 0.5 or higher) suggests an excellent fit of the prior/posterior distribution. To gain a better understanding, the values of the prior and posterior distributions should be compared to determine whether the posterior distribution performs better than the prior. Additionally, a graphical representation of the p-value and distance can be found in the ***QC_plot/QC_posterior_acp.pdf*** file, which is generated using

PCA. However, it is important to note that PCA can be influenced by extreme variances and may therefore be biased.

6.2.4. Barrier detection

For each locus, a bayes_factor and posterior probability is computed and stored in the ***Pbarrier.txt*** file. Take in consideration ONLY the **bayes factor !!!** Also the proportion of barrier detected using the bayes factor is stored in the file ***report_barrier_detection.txt***. The only measure that can be trusted without any re-analysis is the bayes_factor.

7. Exemple cases (WIP)

8. References

Bhatia, G., Patterson, N., Sankararaman, S., Price, A.L., 2013. Estimating and interpreting FST: The impact of rare variants. Genome Res. 23, 1514–1521. https://doi.org/10.1101/gr.154831.113

Hudson, R.R., Slatkin, M., Maddison, W.P., 1992. Estimation of levels of gene flow from DNA sequence data. Genetics 132, 583–589. https://doi.org/10.1093/genetics/132.2.583

Nei, M., Li, W.H., 1979. Mathematical model for studying genetic variation in terms of restriction endonucleases. Proc. Natl. Acad. Sci. U. S. A. 76, 5269–5273.

Tajima, F., 1989. Statistical Method for Testing the Neutral Mutation Hypothesis by DNA Polymorphism. Genetics 123, 585–595.

Watterson, G.A., 1975. On the number of segregating sites in genetical models without recombination. Theor. Popul. Biol. 7, 256–276. https://doi.org/10.1016/0040-5809(75)90020-9