

Package: FreqEstimationModel (via r-universe)

October 5, 2026

Type Package

Title Multi-SNP Haplotype Frequency Estimation

Version 0.1.0

Date 2015-08-02

Description A method for multi-SNP haplotype frequency estimation.

License MIT + file LICENSE

Imports foreach, doMC, rngtools

Suggests abind, coda, gplots, knitr, lattice, plyr, rmarkdown,
vioplot, R.rsp

URL <https://github.com/aimeertaylor/FreqEstimationModel>

Encoding UTF-8

LazyData FALSE

RoxygenNote 7.2.3

VignetteBuilder R.rsp, knitr

Repository <https://plasmogenepi.r-universe.dev>

Date/Publication 2026-10-02 14:18:14 UTC

RemoteUrl <https://github.com/aimeertaylor/FreqEstimationModel>

RemoteRef HEAD

RemoteSha 6e1c796f5225dfb92c53d124a818a06e356eda69

Contents

calculate_genotype_counts_truncated	2
calculate_log_genotype_counts_proposal_ratio	3
calculate_log_moi_proposal_ratio	3
compatible_genotypes	4
data_genotype_count_compatibility	4
ddirichlet	5

draw_frequency_proposed_child	5
draw_genotype_counts_initial	5
draw_moi_initial	6
draw_moi_proposed	6
hello	7
log_likelihood_NGS	7
log_likelihood_prevalence	7
log_moi_prior_calculation_geom	8
log_moi_prior_calculation_nbinom	8
log_moi_prior_calculation_pois	8
log_moi_prior_calculation_unif	9
log_prior_moi_ratio_geom	9
log_prior_moi_ratio_nbinom	9
log_prior_moi_ratio_pois	10
log_prior_moi_ratio_unif	10
mcmc_sampling	11
mcmc_sampling_parallel	11
my_log_dmultinom	12
preprocess_data	12
repp_hello	12
rdirichlet	13
update_genotype_counts	13
update_y_mixed_add	13
update_y_mixed_sub	14
Index	15

calculate_genotype_counts_truncated
<i>Calculate legal genotype count moves</i>

Description

Function to calculate the genotype counts that can be removed without rendering the ensuing likelihood equal to zero

Usage

```
calculate_genotype_counts_truncated(  
  genotype_counts,  
  comp_genotypes,  
  raw_data,  
  moi_prior_min2  
)
```

`calculate_log_genotype_counts_proposal_ratio`*Calculate the log genotype counts proposal ratio*

Description

Function to calculate the log genotype counts proposal ratio

Usage

```
calculate_log_genotype_counts_proposal_ratio(  
  moi_diff,  
  genotype_counts_diff,  
  neg_log_sum_frequency_truncated,  
  mh_genotype_counts_truncated,  
  y_character,  
  log_target_ratio,  
  datasampleID,  
  no_total  
)
```

`calculate_log_moi_proposal_ratio`*Calculate log MOI proposal ratio*

Description

Function to calculate the log MOI proposal ratio

Usage

```
calculate_log_moi_proposal_ratio(  
  moi_max,  
  moi_current,  
  moi_proposed,  
  moi_current_equal_moi_min,  
  moi_proposed_equal_moi_min,  
  datasampleID,  
  no_total  
)
```

compatible_genotypes *Compatible genotypes*

Description

Function to ascertain the genotypes that are compatible with a given observation

Usage

```
compatible_genotypes(observation, as_numeric = TRUE)
```

Arguments

observation A vector of allelic calls for a given individual, with one call per loci. Heteroallelic loci can either be encoded as 0.5. Alternatively, all alleles detected at heteroallelic loci can be listed. See examples below.

Examples

```
# Example where heteroallelic loci are encoded as 0.5
y <- c(0, 1, 0.5, 0.5)
compatible_genotypes(y)
compatible_genotypes(y, FALSE)

# Example where all alleles detected at heteroallelic loci are listed
y <- list(0, 1, c(0,1), c(0,1,2))
compatible_genotypes(y)
compatible_genotypes(y, FALSE)
```

data_genotype_count_compatibility
Data genotype count compatibility

Description

Data and genotype compatibility check function

Usage

```
data_genotype_count_compatibility(comp_genotypes, genotype_counts, raw_data)
```

ddirichlet	<i>Dirichlet density</i>
------------	--------------------------

Description

Calculate the density of the dirichlet distribution. This function is exactly the same as that from MCMCpack, copied to minimise dependencies.

Usage

```
ddirichlet(x, alpha)
```

draw_frequency_proposed_child
<i>Draw frequency proposed child</i>

Description

Function for drawing frequency proposed per child

Usage

```
draw_frequency_proposed_child(Alpha, genotype_counts_current)
```

draw_genotype_counts_initial
<i>Draw initial genotype counts</i>

Description

Function for drawing initial genotype counts.

Usage

```
draw_genotype_counts_initial(  
  moi_current,  
  processed_data_list,  
  frequency_current  
)
```

draw_moi_initial	<i>Draw initial MOIs</i>
------------------	--------------------------

Description

Function for drawing initial MOIs.

Usage

```
draw_moi_initial(moi_list, datasampleID, y_no_mxed, y_mxed)
```

Arguments

datasampleID	is a vector of sample IDs; that is, a character vector such as <code>c('id1:1', 'id2:1', 'id3:1', 'id4:1')</code> . It is required to ensure the order of the initial MOI estimates is consistent with the data.
y_no_mxed	is a vector comprising a subset of <code>datasampleID</code> ; for example <code>c('id1:1', 'id4:1')</code> . The subset includes sample IDs for all samples for which there is no evidence of multiclonality; that is, samples for which an MOI of one is possible.
y_mxed	is a vector comprising a subset of <code>datasampleID</code> ; for example <code>c('id1:2', 'id4:3')</code> . The subset includes sample IDs for all samples for which there is evidence of multiclonality; that is, samples for which an MOI of one is not possible. Note that, <code>y_no_mxed</code> and <code>y_mxed</code> are disjoint sets that together union of is equal to the unordered equivalent of <code>datasampleID</code> .
moi_list\$moi_prior	is a character, either 'Poisson', 'nBinomial', 'Geometric' or 'Uniform', that defines the prior distribution over the MOI.
moi_list\$moi_max	is an integer defining the maximum moi possible in the model
moi_list\$moi_hyperparameter	is a parameter of the prior distribution
moi_list\$moi_size_hyperparameter	is a parameter of the prior distribution

draw_moi_proposed	<i>Propose MOI</i>
-------------------	--------------------

Description

Function for drawing new MOI vector. Note that there is no option to stay still, and that Proposal density does not have equal probability over state space $[m_{\min}, \dots, m_{\max}]$ and is not symmetric: $\Pr(m^*|m) = \Pr(m|m^*)$

Usage

```
draw_moi_proposed(moi_current, moi_current_equal_moi_min, moi_max)
```

hello	<i>Hello, World!</i>
-------	----------------------

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

log_likelihood_NGS	<i>Log likelihood, NGS data</i>
--------------------	---------------------------------

Description

Calculate log likelihood of NGS data

Usage

```
log_likelihood_NGS(processed_data_list, genotype_counts)
```

log_likelihood_prevalence	<i>Log likelihood, prevalence data</i>
---------------------------	--

Description

Calculate log likelihood of prevalence data

Usage

```
log_likelihood_prevalence(processed_data_list, genotype_counts)
```

log_moi_prior_calculation_geom
<i>Log moi prior</i>

Description

Log moi prior calculation

Usage

log_moi_prior_calculation_geom(moi_store, moi_list, y_no_mxed, y_mxed)

log_moi_prior_calculation_nbinom
<i>Log moi prior</i>

Description

Log moi prior calculation

Usage

log_moi_prior_calculation_nbinom(moi_store, moi_list, y_no_mxed, y_mxed)

log_moi_prior_calculation_pois
<i>Log moi prior</i>

Description

Log moi prior calculation

Usage

log_moi_prior_calculation_pois(moi_store, moi_list, y_no_mxed, y_mxed)

```
log_moi_prior_calculation_unif
```

Log moi prior

Description

Log moi prior calculation

Usage

```
log_moi_prior_calculation_unif(moi_store, moi_list, y_no_mxed, y_mxed)
```

```
log_prior_moi_ratio_geom
```

llg prior MOI ratio

Description

Function to calculate the log of the prior MOI ratio when the MOI prior distribution is geometric. Note that the normalising constants due to truncation cancel.

Usage

```
log_prior_moi_ratio_geom(
  moi_proposed,
  moi_current,
  moi_hyperparameter,
  moi_size_hyperparameter
)
```

```
log_prior_moi_ratio_nbinom
```

Log prior MOI ratio

Description

Function to calculate the log of the prior MOI ratio when the MOI prior distribution is negative binomial. Note that the normalising constants due to truncation cancel.

Usage

```
log_prior_moi_ratio_nbinom(  
  moi_proposed,  
  moi_current,  
  moi_hyperparameter,  
  moi_size_hyperparameter  
)
```

log_prior_moi_ratio_pois

Log prior MOI ratio

Description

Function to calculate the log of the prior MOI ratio when the MOI prior distribution is poisson. Note that the normalising constants due to truncation cancel.

Usage

```
log_prior_moi_ratio_pois(  
  moi_proposed,  
  moi_current,  
  moi_hyperparameter,  
  moi_size_hyperparameter  
)
```

log_prior_moi_ratio_unif

Log prior MOI ratio

Description

Function to calculate the log of the prior MOI ratio when the MOI prior distribution is a discrete uniform distribution. Note that the normalising constants due to truncation cancel.

Usage

```
log_prior_moi_ratio_unif(  
  moi_proposed,  
  moi_current,  
  moi_hyperparameter,  
  moi_size_hyperparameter  
)
```

mcmc_sampling

MCMC

Description

Function to implement mcmc sampler, which is embedded within data preprocessing and retrospective posterior calculations

Usage

```
mcmc_sampling(
  processed_data_list,
  moi_list,
  frequency_list,
  mcmc_variable_list
)
```

mcmc_sampling_parallel

MCMC

Description

Function to implement mcmc sampler chains in parallel, followed retrospective posterior calculations

Usage

```
mcmc_sampling_parallel(
  processed_data_list,
  moi_list,
  frequency_list,
  mcmc_variable_list,
  cores_max,
  seed = 1234352
)
```

Arguments

```
processed_data_list      List of processed data
moi_list                 List of MOI settings
frequency_list           List of frequency settings
mcmc_variable_list       List of MCMC settings
```

cores_max	Maximum number of cores to register with doMC
seed	Integer seed for the per-chain RNG streams built by RNGseq().
my_log_dmultinom	<i>Log Multinomial Density</i>

Description

Function to calculate the density of multiple realisations of the multinomial distribution simultaneously given either a single probability or a matrix of probability

Usage

```
my_log_dmultinom(x, prob, pop_fequency)
```

preprocess_data	<i>Preprocess data</i>

Description

Function to preprocess data

Usage

```
preprocess_data(  
  data_summary,  
  log_like_zero,  
  NGS,  
  augment_missing_data,  
  moi_prior_min2  
)
```

rcpp_hello	<i>Hello, Rcpp!</i>

Description

Returns an R list containing the character vector c("foo", "bar") and the numeric vector c(0, 1).

Usage

```
rcpp_hello()
```

Examples

```
rcpp_hello()
```

rdirichlet	<i>Random draw from the dirichlet</i>
------------	---------------------------------------

Description

Same as the rdirichlet function in MCMCpack but rewritten to minimise dependencies.

Usage

```
rdirichlet(n, alpha)
```

Details

Need to add log option

update_genotype_counts	<i>Update genotype counts</i>
------------------------	-------------------------------

Description

Function to update genotype counts conditional on current genotype counts and new multiplicity of infection (MOI)

Usage

```
update_genotype_counts(
  moi_diff,
  genotype_counts_current,
  current_genotype_counts_truncated,
  processed_data_list
)
```

update_y_mixed_add	<i>Update mixed data samaple clone addition</i>
--------------------	---

Description

Function for addition of clone to a mixed data sample

Usage

```
update_y_mixed_add(to_update, genotype_counts_mixed_add, frequency_truncated)
```

update_y_mixed_sub	<i>Update mixed data samaple clone removal</i>
--------------------	--

Description

Function for removal of clone from mixed data sample

Usage

```
update_y_mixed_sub(  
  to_update,  
  genotype_counts_mixed_sub,  
  genotype_counts_truncated  
)
```

Index

`calculate_genotype_counts_truncated`, [2](#)
`calculate_log_genotype_counts_proposal_ratio`,
 [3](#)
`calculate_log_moi_proposal_ratio`, [3](#)
`compatible_genotypes`, [4](#)

`data_genotype_count_compatibility`, [4](#)
`ddirichlet`, [5](#)
`draw_frequency_proposed_child`, [5](#)
`draw_genotype_counts_initial`, [5](#)
`draw_moi_initial`, [6](#)
`draw_moi_proposed`, [6](#)

`hello`, [7](#)

`log_likelihood_NGS`, [7](#)
`log_likelihood_prevalence`, [7](#)
`log_moi_prior_calculation_geom`, [8](#)
`log_moi_prior_calculation_nbinom`, [8](#)
`log_moi_prior_calculation_pois`, [8](#)
`log_moi_prior_calculation_unif`, [9](#)
`log_prior_moi_ratio_geom`, [9](#)
`log_prior_moi_ratio_nbinom`, [9](#)
`log_prior_moi_ratio_pois`, [10](#)
`log_prior_moi_ratio_unif`, [10](#)

`mcmc_sampling`, [11](#)
`mcmc_sampling_parallel`, [11](#)
`my_log_dmultinom`, [12](#)

`preprocess_data`, [12](#)

`rcpp_hello`, [12](#)
`rdirichlet`, [13](#)

`update_genotype_counts`, [13](#)
`update_y_mixed_add`, [13](#)
`update_y_mixed_sub`, [14](#)