# Package ‘ASAFE’

May 14, 2025

**Type** Package

**Title** Ancestry Specific Allele Frequency Estimation

**Version** 1.34.0

**Date** 2016-04-24

**Author** Qian Zhang <qszhang@uw.edu>

**Maintainer** Qian Zhang <qszhang@uw.edu>

**Description** Given admixed individuals' bi-allelic SNP genotypes and ancestry pairs (where each ancestry can take one of three values) for multiple SNPs, perform an EM algorithm to deal with the fact that SNP genotypes are unphased with respect to ancestry pairs, in order to estimate ancestry-specific allele frequencies for all SNPs.

**License** Artistic-2.0

**biocViews** SNP, GenomeWideAssociation, LinkageDisequilibrium, BiomedicalInformatics, Genetics, ExperimentalDesign

**LazyData** TRUE

**Suggests** knitr, testthat

**VignetteBuilder** knitr

**Collate** estep.R nudge.R mstep.R increment_n.R em.R algorithm_1snp.R algorithm_1snp_wrapper.R get_mean_sd_error_anc_bin.R get_mean_sd_error_anc.R change_ancestry.R draw_allele_given_anc.R get_errors_1_scenario.R get_errors_summary_stats_1_scenario.R get_results_error.R get_scenario_errors.R get_true_freqs_1snp.R sample_ancestry.R

**NeedsCompilation** no

**Depends** R (>= 3.2)

**git_url** https://git.bioconductor.org/packages/ASAFE

**git_branch** RELEASE_3_21

**git_last_commit** 82f37de

**git_last_commit_date** 2025-04-15

**Repository** Bioconductor 3.21

**Date/Publication** 2025-05-14

# Contents



---

`adm_ancestries_test` *Ancestries of 250 admixed individuals at 6 SNPs*

---

### Description

This matrix adm_ancestries_test stores a subset of the full data set of simulated phased ancestries (meaning ancestries at different markers are phased with respect to each other) that was used in the ASAFE paper. adm_ancestries_test contains ancestries at 6 markers for 250 admixed individuals.

For each individual at a marker, the ancestry pair is also phased with respect to the genotype given in adm_genotypes_test, so that true ancestry-specific allele frequencies can be calculated from adm_ancestries_test and adm_genotypes_test by overlaying ancestries on genotypes. The ASAFE EM algorithm does not assume that ancestries and genotypes at the same marker are phased with respect to each other, or that ancestries at different markers are phased with respect to each other, or that genotypes at different markers are phased with respect to each other, and provides estimates of true ancestry-specific alelle frequencies.

### Usage

```
adm_ancestries_test
```

### Format

A 6 x 501 matrix with the following rows, columns, and entries:

1. Rows: 1 row per bi-allelic marker
2. Columns: First column is Marker ID. Following columns consist of 1 column per chromosome, with two consecutive columns per individual, corresponding to the individual's pair of homologous chromosomes. For example, the first 5 column names are Marker, ADM1, ADM1.1, ADM2, and ADM2.1. Columns ADM1 and ADM1.1 correspond to one individual's 2 homologous chromosomes, and columns ADM2 and ADM2.1 correspond to another individual's 2 homologous chromosomes.
3. Entries: For an entry that is not in the Marker ID column, an entry can take value 0, 1, or 2, which are arbitrary labels for three ancestries.

### Author(s)

Qian Zhang

## Source

Simulated ancestry data

## References

Zhang QS, Browning BL, and Browning SR (2016) "Ancestry Specific Allele Frequency Estimation." Bioinformatics.

---

adm_genotypes_test *Genotypes of 250 admixed individuals at 6 markers*

---

## Description

This matrix adm_genotypes_test stores a subset of the full data set of simulated phased genotypes (meaning genotypes at different markers are phased with respect to each other) that was used in the ASAFE paper. adm_genotypes_test contains genotypes at 6 markers for 250 admixed individuals.

For each individual at a marker, the genotype is also phased with respect to the ancestry pair given in adm_ancestries_test, so that true ancestry-specific allele frequencies can be calculated from adm_genotypes_test and adm_ancestries_test by overlaying ancestries on genotypes. The ASAFE EM algorithm does not assume that ancestries and genotypes at the same marker are phased with respect to each other, or that ancestries at different markers are phased with respect to each other, or that genotypes at different markers are phased with respect to each other, and provides estimates of true ancestry-specific alelle frequencies.

## Usage

```
adm_genotypes_test
```

## Format

A 6 x 251 matrix with the following rows, columns, and entries:

1. Rows: 1 row per bi-allelic marker, with alleles arbitrarily labeled 0 and 1
2. Columns: First column is Marker ID. Following columns consist of 1 column per individual. Individuals should be listed in the same order in the genotype matrix adm_genotypes_test as in the ancestry matrix adm_ancestries_test.
3. Entries: For an entry that is not in the Marker ID column, an entry can take value 0/0, 0/1, 1/0, or 1/1, where 0 and 1 are arbitrary labels for a bi-allelic SNP's two alleles. A slash "/" indicates an unphased genotype, so 0/1 and 1/0 are the same unphased genotype. It just so happens that this particular adm_genotypes_test matrix contains phased genotypes, despite the presence of slashes.

## Author(s)

Qian Zhang

## Source

Simulated genetic data

## References

Zhang QS, Browning BL, and Browning SR (2016) "Ancestry Specific Allele Frequency Estimation." Bioinformatics.

---

`algorithm_1snp` *Estimate ancestry-specific allele frequencies for 1 marker (e.g. a SNP) from individuals' alleles and ancestries at this marker.*

---

## Description

Take in genotypes (possibly unphased with respect to each other) and ancestries (possibly unphased with respect to each other) for all individuals at 1 marker to create the marker's vector of observed data category counts, and then call the function em() on that vector of counts, to obtain ancestry-specific allele frequency estimates for that marker.

## Usage

```
algorithm_1snp(alleles_1, ancestries_1)
```

## Arguments

`alleles_1` Vector of alleles for each individual's 2 chromosomes, with chromosomes for the same individual consecutive. Each allele is either 0 or 1. This is a numeric vector.

Example: If there are 250 admixed individuals, the alleles might be ordered like so: ADM1, ADM1, ADM2, ADM2, ..., ADM250, ADM250, where ADMi is the ID for the i-th individual.

`ancestries_1` Vector of ancestries for each individual's 2 chromosomes, with chromosomes for the same individual consecutive. Each ancestry is either 0, 1, or 2. This is a numeric vector.

Example: If there are 250 admixed individuals, the ancestries might be ordered like so: ADM1, ADM1, ADM2, ADM2, ..., ADM250, ADM250, where ADMi is the ID for the i-th individual.

## Value

Ancestry-specific allele frequency estimates of [P(Allele 1| Ancestry 0), P(Allele 1 | Ancestry 1), P(Allele 1 | Ancestry 2)] from the EM Algorithm. This a numeric vector with 3 entries.

## Author(s)

Qian Zhang

## Examples

```
# adm_ancestries_test is a matrix with
# Rows: Markers
# Columns: Marker ID, individuals' chromosomes' ancestries
# (e.g. ADM1, ADM1, ADM2, ADM2, and etc.)

# adm_genotypes_test is a matrix with
# Rows: Markers
# Columns: Marker ID, individuals' genotypes (a1/a2)
# (e.g. ADM1, ADM2, ADM3, and etc.)

# Make the rsID column row names
row.names(adm_ancestries_test) <- adm_ancestries_test[,1]
row.names(adm_genotypes_test) <- adm_genotypes_test[,1]

adm_ancestries_test <- adm_ancestries_test[,-1]
adm_genotypes_test <- adm_genotypes_test[,-1]

# alleles_list is a list of lists.
# Outer list elements correspond to SNPs.
# Inner list elements correspond to 250 individuals's alleles with no delimiter separating alleles.

alleles_list <- apply(X = adm_genotypes_test, MARGIN = 1,
                        FUN = strsplit, split = "/")

# Creates a matrix: Number of alleles
# (ADM1, ADM1, ..., ADM250, ADM250) x (SNPs)

alleles_unlisted <- sapply(alleles_list, unlist)

# Change elements of the matrix to numeric, producing a matrix:
# Number of alleles (ADM1, ADM1, ..., ADM250, ADM250) x (SNPs).

alleles <- apply(X = alleles_unlisted, MARGIN = 2, as.numeric)

# Perform the EM algorithm on the first SNP in the data, obtaining estimates for
# P(Allele 1 | Ancestry 0), P(Allele 1 | Ancestry 1), P(Allele 1 | Ancestry 2)

estimates <- algorithm_1snp(alleles[,1], adm_ancestries_test[1,])

estimates
```

---

`algorithm_1snp_wrapper` *Wrapper for function algorithm_1snp*

---

## Description

Applies the function algorithm_1snp to a particular bi-allelic marker's data stored in matrices alleles and ancestries. Can be used to apply algorithm_1snp to multiple bi-allelic markers.

## Usage

```
algorithm_1snp_wrapper(i, alleles, ancestries)
```

## Arguments

| | |
|---|---|
| i | Index of marker in matrices alleles and ancestries. This is the index for the marker that we want to apply function algorithm_1snp to. |
| alleles | Rows: Alleles for individuals' chromosomes ordered e.g. ADM1, ADM1, ..., ADM250, ADM250, where ADMi is the ID for the i-th individual. Cols: Bi-allelic markers. Each allele is either 0 or 1. This is a numeric matrix. |
| ancestries | Rows: Ancestries for individuals' chromosomes ordered e.g. ADM1, ADM1, ..., ADM250, ADM250, where ADMi is the ID for the i-th individual. Cols: Bi-allelic markers. Each ancestry is either 0, 1, or 2. This is a numeric matrix. |

## Details

Markers in matrix alleles should be in 1-to-1 correspondence with markers in matrix ancestries. Markers in both matrices should be in the same order.

## Value

A character vector with 4 elements. The first element is the Marker ID of the i-th marker in matrices alleles and ancestries. The next 3 elements are ancestry-specific allele frequency estimates of P(Allele 1 | Ancestry 0), P(Allele 1 | Ancestry 1), and P(Allele 1 | Ancestry 2), for the i-th marker in matrices alleles and ancestries.

## Author(s)

Qian Zhang

## Examples

```
# adm_ancestries_test is a matrix with
# Rows: Markers
# Columns: Marker ID, individuals' chromosomes' ancestries
# (e.g. ADM1, ADM1, ADM2, ADM2, and etc.)

# adm_genotypes_test is a matrix with
# Rows: Markers
# Columns: Marker ID, individuals' genotypes (a1/a2)
# (e.g. ADM1, ADM2, ADM3, and etc.)

# Making the rsID column row names
row.names(adm_ancestries_test) <- adm_ancestries_test[,1]
row.names(adm_genotypes_test) <- adm_genotypes_test[,1]

adm_ancestries_test <- adm_ancestries_test[,-1]
adm_genotypes_test <- adm_genotypes_test[,-1]

# alleles_list is a list of lists.
```

```
# Outer list elements correspond to bi-allelic markers.
# Inner list elements correspond to 250 people's alleles with no delimiter separating alleles.
alleles_list <- apply(X = adm_genotypes_test, MARGIN = 1, FUN = strsplit, split = "/")

# Creates a matrix: Number of alleles (ADM1, ADM1, ..., ADM250, ADM250) x (bi-allelic markers)
alleles_unlisted <- sapply(alleles_list, unlist)

# Change elements of the matrix to numeric, producing a matrix:
# Number of alleles (ADM1, ADM1, ..., ADM250, ADM250) x (bi-allelic markers).
alleles <- apply(X = alleles_unlisted, MARGIN = 2, as.numeric)

# Apply EM algorithm to first bi-allelic marker
algorithm_1snp_wrapper(i = 1, alleles, adm_ancestries_test)

# Apply the EM algorithm to each bi-allelic marker to obtain
# ancestry-specific allele frequency estimates for all markers in
# matrices alleles and ancestries.

adm_estimates_test <- sapply(X = 1:ncol(alleles), FUN = algorithm_1snp_wrapper,
                        alleles = alleles, ancestries = adm_ancestries_test)

adm_estimates_test
```

# Index