# EggLib's Statistics

**Version: 3.0.0**

Stéphane De Mita Mathieu Siol

March 19, 2015

This documents lists all statistics computed by EggLib in the context of population genetics analysis and provides the formulæ used and bibliographic references.

The standard form of accepted data is site, allowing to load data taken from sequence alignments, large-scale sequencing or genotyping data, or other kinds of markers. Typically, it is expected that the user verifies that the site is polymorphic. In addition, the user may have previously checked that the site contains enough non-missing data (using `FreqBase::nsam()`, also available from `FreqBase`'s subclasses). `FreqBase`

In equations, statistics are named using mathematical notations that are not necessarily matching notations from the literature (in particular when notations have been inconsistent or when several statistics have identical names). For example, in this document, Tajima's $D$ and Fu and Li's $D$ are named $D_t$ and $D_{fl}$, respectively, in this document. The corresponding accessing functions in the C++ library are given in this format: `Diversity1::D()` and `Diversity1::Dfl()`.

**Reference:** De Mita S. & M. Siol. 2012. EggLib: processing, analysis and simulation tools for population genetics and genomics. *BMC Genet.* **13**: 27

**URL:** `http://egglib.sourceforge.net/`

# Contents

# 1 Single-site statistics

These statistics are computed from a single site. Many biological marker can fit in this category, including SNPs, SSRs and haplotypes after phase reconstruction. Analyzes are provided by the `SiteDiversity` class and data may be loaded from several classes (`Site`, `DataMatrix` or `VcfParser`).

## 1.1 Basic statistics

The first set of statistics are not really statistics in themselves but are important. They can be computed by several methods. In addition to this list, the sample sizes are available in the object used to analyze polymorphism (usually a subclass of `FreqBase`).

| Stat | Accessing method | Explanation |
|---|---|---|
| $k$ | `SiteDiversity::k()` | Number of populations |
| $k_e$ | `SiteDiversity::keff()` | — with enough samples † |
| $n_s$ | `SiteDiversity::ns()` | Number of analyzed samples |
| $I_o$ | `SiteDiversity::orientable()` | 1 if the site is orientable, 0 otherwise ‡ |
| $d$ | `SiteDiversity::derived()` | Sum of frequencies of derived alleles § |

† The criterion depends on the method called: at least two samples for standard and differentiation statistics, at least one sample (or diploid sample) for Weir and Cockerham's $F$-statistics (except with two levels of structure where this statistic is not computed) and for allele size variance.

‡ A site is orientable if and only if exactly one ingroup-specific allele is present in the outgroup.

§ Also available per population.

As set of standard statistics are computed by `SiteDiversity::stats()`, `SiteDiversity::vstats()` (for the allele size variance) and for $\theta$ estimators, directly by accessors.

| Stat | Accessing method | Explanation |
|---|---|---|
| $A$ | `SiteDiversity::Atot()` | Number of alleles † |
| $A_e$ | `SiteDiversity::Aeff()` | — excluding outgroup-specific alleles †‡ |
| $\epsilon$ | `SiteDiversity::S()` | — present in one copy (singletons) † |
| $\epsilon_d$ | `SiteDiversity::Sd()` | — present in one copy (only derived) † |
| $R$ | `SiteDiversity::R()` | Allelic richness ‡ |
| $H'$ | `SiteDiversity::pairdiff()` | Average number of pairwise differences |
| $H'_{ij}$ | `SiteDiversity::pairdiff_inter(i,j)` | — between populations $i$ and $j$ |
| $H_e$ | `SiteDiversity::He()` | Unbiased heterozygosity ‡ |
| $V$ | `SiteDiversity::V()` | Allele size variance ‡ |
| $\hat{\theta}_I$ | `SiteDiversity::thetaIAM()` | Estimator of $\theta$ assuming IAM ‡ |
| $\hat{\theta}_S$ | `SiteDiversity::thetaSMM()` | Estimator of $\theta$ assuming SMM ‡ |

† An allele is an instance of a marker (nucleotide, haplotype, band or any other) which is present at a non-null frequency (even if fixed) in the considered sample.

‡ Also available per population.

$$H' = 1 - \sum_{i}^{A_e} {p_i}^2$$

$$H'_{ij} = \sum_{k}^{A_e} p_{ki} \cdot p_{kj}$$

$$H_e = H' \frac{n_s}{n_s - 1}$$

$$R = \frac{A_e - 1}{n_s - 1}$$

$$V = \frac{1}{n_s} \sum_{i}^{A_e} (p_i X_i)^2 - \left( \frac{1}{n_s} \sum_{i}^{A_e} p_i X_i \right)^2$$

$$\hat{\theta}_I = \frac{H_e}{1 - H_e}$$

$$\hat{\theta}_S = \frac{1}{2} \left[ \frac{1}{(1 - H_e)^2} - 1 \right]$$

with $p_i$, the relative frequency of allele $i$, $p_{ij}$ the relative frequency of allele $i$ in population $j$, and $X_i$ the integer value of allele $i$, which is interpreted as the allele size when it is used.

The observed (absolute of frequency can be accessed from the class `GenoFreq` which is designed to hold diploid data (actually, it can support any ploidy, provided that it is consistent over samples).

| Accessing method | Explanation |
|---|---|
| `GenoFreq::Ho()` | Absolute frequency of heterozygotes |
| `GenoFreq::Ho(k)` | — in population $k$ |
| `GenoFreq::Ho_out()` | — in the outgroup |
| `GenoFreq::het(a)` | — for allele $a$ † |
| `GenoFreq::het(a,p)` | — for allele $a$ in population $k$ † |

† Number of heterozygotes containing the allele $a$ (at least one copy, whenever higher level of ploidy are used).

**Note** $H_e$ and $R$ are not defined if $n_s < 2$. $V$ is not defined if $n_s = 0$.

## 1.2 Weir and Cockerham's F-statistics

The fixation indices defined by Weir & Cockerham (1984) are implemented in three variants of the methods `SiteDiversity::fstats()`, depending on whether one or two levels of structure are processed and whether genotypes are available. The statistics are not computed directly in the C++ library (in order to let the user compute sums if needed). They can be computed using the components of variances.

**Diploid data, one level of structure** In the standard case, the components of variance are computed as follows (Weir & Cockerham 1984):

| Stat | Accessing method | Source of variance |
|---|---|---|
| $a$ | `SiteDiversity::a()` | Between populations |
| $b$ | `SiteDiversity::b()` | Between individuals within populations |
| $c$ | `SiteDiversity::c()` | Within individuals |

$$a = \sum_{i}^{A_e} \frac{\bar{n}}{n_c} \left\{ s_i^2 - \frac{1}{\bar{n}-1} \left[ \bar{p}_i(1-\bar{p}_i) - \frac{k_e-1}{k_e} s_i - \frac{1}{4}\bar{h}_i \right] \right\}$$

$$b = \sum_{i}^{A_e} \frac{\bar{n}}{\bar{n}-1} \left[ \bar{p}_i(1-\bar{p}_i) - \frac{k_e-1}{k_e} s_i^2 - \frac{2\bar{n}-1}{4\bar{n}}\bar{h}_i \right]$$

$$c = \sum_{i}^{A_e} \frac{1}{2}\bar{h}_i$$

with:

$$\bar{n} = \frac{1}{k_e} \sum_{i}^{k_e} n_i$$

$$n_c = \frac{1}{k_e - 1} \left( k_e \cdot \bar{n} - \frac{1}{k_e \cdot \bar{n}} \sum_{i}^{k_e} {n_i}^2 \right)$$

$$\bar{h}_i = \frac{1}{\bar{n} \cdot k_e} \sum_{j}^{k_e} h_j$$

$$\bar{p}_i = \frac{1}{k_e} \sum_{j}^{k_e} \frac{p_{ij}}{n_j}$$

$$s_i^2 = \frac{1}{\bar{n}(k_e-1)} \sum_{j}^{k_e} n_j (p_{ij} - \bar{p}_i)$$

where $n_j$ is the number of diploid individuals in population $j$, $h_j$ is the number of

heterozygotes in population $j$ and $p_{ij}$ is the relative frequency of allele $i$ in population $j$.

The user may compute the final statistics using the relations, where $\hat{F}$ is the estimator corresponding to $F_{it}$, $\hat{\theta}$ to $F_{st}$, and $\hat{f}$ to $F_{is}$, summing the numerator and denominator as needed over loci:

$$1 - \hat{F} = \frac{c}{a + b + c}$$

$$\hat{\theta} = \frac{a}{a + b + c}$$

$$1 - \hat{f} = \frac{c}{b + c}$$

**Diploid data, two levels of structure** Here is how are computed the components of variance when populations are arranged in clusters (Weir & Cockerham 1984):

| Stat | Accessing method | Source of variance |
|---|---|---|
| $a$ | `SiteDiversity::a()` | Between clusters |
| $b_2$ | `SiteDiversity::b2()` | Between populations within clusters |
| $b_1$ | `SiteDiversity::b1()` | Between individuals within populations |
| $c$ | `SiteDiversity::c()` | Within individuals |

$$a = \sum_{i}^{A_e} \frac{1}{2n_2n_3} \left[n_3\alpha_i - n_1\beta_i - (n_3 - n_1)\gamma_i\right]$$

$$b_2 = \sum_{i}^{A_e} \frac{1}{2n_3} \left(\beta_i - \gamma_i\right)$$

$$b_1 = \sum_{i}^{A_e} \frac{1}{2} \left(\gamma_i - \delta_i\right)$$

$$c = \sum_{i}^{A_e} \delta_i$$

$\alpha_i$, $\beta_i$, $\gamma_i$ and $\delta_i$ correspond, respectively, to *MSP*, *MSD*, *MSI* and *MSG* in Weir and Cockerham (1984). To compute them, we define: $K$, the number of clusters; $r_i$, the number of populations in cluster $i$; $n_{ij}$ the number of diploid samples of population $j$ of cluster $i$; $n_{i.}$ the number of diploid samples in cluster $i$; and $n_{..}$ the total number of diploid samples. $r'_i$ is the number of populations with at least one sample in cluster $i$, $K'$ the number of clusters $i$ such that $r'_i > 0$ and $r'$ is the number of populations with at least one diploid sample. Then we have:

$$n_1 = \frac{1}{K' - 1} \sum_{i}^{K'} \sum_{j}^{r'_i} \frac{(n_{..} - n_{i.}) {n_{ij}}^2}{n_{i.} n_{..}}$$

$$n_2 = \frac{1}{K' - 1} \left( n_{..} - \frac{1}{n_{..}} \sum_{i}^{K'} {n_{i.}}^2 \right)$$

$$n_3 = \frac{1}{r' - K'} \left( n_{..} - \sum_{i}^{K'} \frac{1}{n_{i.}} \sum_{j}^{r'_i} {n_{ij}}^2 \right)$$

Let $p_{ijk}$ be the relative frequency of allele $i$ in population $k$ of cluster $j$, $p_{ij.}$ the relative frequency of allele $i$ in cluster $j$ and $p_{i.}$ the relative frequency of allele $i$ in the whole sample. $k_{ijk}$ is the relative frequency of heterozygotes for allele $i$ in population $k$ of cluster $j$. Then we have:

$$\alpha_i = \frac{2}{K' - 1} \sum_{j}^{K'} n_{j.} (p_{ij.} - p_{i..})^2$$

$$\beta_i = \frac{2}{r' - K'} \sum_{j}^{K'} \sum_{k}^{r'_j} n_{jk} (p_{ijk} - p_{ij.})^2$$

$$\gamma_i = \frac{1}{n_{..} - r'} \left[ 2 \sum_{j}^{K'} \sum_{k}^{r'_j} n_{jk} \cdot p_{ijk}(1 - p_{ijk}) - \frac{1}{2} \sum_{j}^{K'} \sum_{k}^{r'_j} n_{jk} \cdot h_{ijk} \right]$$

$$\delta_i = \frac{1}{n_{..}} \sum_{j}^{K'} \sum_{k}^{r'_j} n_{jk} \cdot h_{ijk}$$

There is now two levels fixation indices corresponding to between-population differentiation that may be computed, $\hat{\theta}_1$ and $\hat{\theta}_2$, but once again only the components of variance are exposed by the library, to allow summing over loci:

$$1 - \hat{F} = \frac{c}{a + b_1 + b_2 + c}$$

$$\hat{\theta}_1 = \frac{a + b_2}{a + b_1 + b_2 + c}$$

$$\hat{\theta}_2 = \frac{a}{a + b_1 + b_2 + c}$$

**Haploid data, one level of structure** It is also possible to compute $\hat{\theta}$ (corresponding to $F_{st}$) with haploid data. Again, the C++ library provides only the terms to compute

the value, as $n$, the numerator, and $d$, the denominator (Weir and Hill, 2002).

| Stat | Accessing method | Source of variance |
|---|---|---|
| $n$ | `SiteDiversity::n()` | Between population |
| $d$ | `SiteDiversity::d()` | Total |

$$n = \sum_{i}^{A_e} \alpha_i - \delta_i$$

$$d = \sum_{i}^{A_e} \alpha_i + (n_c - 1)\delta_i$$

where $\alpha_i$ is *MSP* and $\delta_i$ is *MSG* (for locus $i$) in Weir and Cockerham (2002) and are computed as follows (all terms as previously defined):

$$\alpha_i = \frac{1}{k_e} \sum_{j}^{k_e} n_j (p_{ij} - \bar{p}_i)^2$$

and

$$\delta_i = \frac{1}{\bar{n}(k_e - 1)} \sum_{i}^{k_e} n_j p_{ij}(1 - p_{ij})$$

Finally, $\hat{\theta}$ can be obtained easily:

$$\hat{\theta} = \frac{n}{d}$$

**Note** The components of variance are not computed if $k_e < 2$ or if $\bar{n} = k_e$ (meaning that there is not more than one sample per population). In the case with two levels of structure, the computation is also skipped if the number of clusters is $< 2$ or if there is not more than one population per cluster (that is, if $r' = K'$).

## 1.3 Differentiation statistics

The method `SiteDiversity::hstats()` computes the differentiation index $D$ of Jost (2008), Nei's $G_{st}$ and Hudson's $H_{st}$ (Hudson *et al.* 1992a), as well as Nei and Chesser's $\hat{G}_{st}$ (Nei and Chesser 1983) and Hedrick's $\hat{G}'_{st}$ (Hedrick 2005). All but $D$ are not available directly but must be computed by the user using terms provided by the C++ library.

| Stat | Accessing method | Explanation |
|---|---|---|
| $D_j$ | `SiteDiversity::D()` | Jost's differentiation index $D$ |
| $H_s$ | `SiteDiversity::Hs()` | Within-population diversity |
| $\tilde{H}_t$ | `SiteDiversity::Httilde()` | Total diversity, Hudson's formula |
| $\hat{H}_s$ | `SiteDiversity::Hse()` | Estimator of $H_s$ |
| $\hat{H}_t$ | `SiteDiversity::Hte()` | Estimator of the total diversity |

$$H_s = \frac{1}{n_s} \sum_{i}^{k_e} n_i H_{e,i}$$

$$\tilde{H}_t = H' + \frac{H_s}{k_e \tilde{n}}$$

where $H_{e,i}$ is $H_e$ for population $i$ and $\tilde{n}$ is such that:

$$\frac{1}{\tilde{n}} = \frac{1}{k_e} \sum_{i}^{k_e} \frac{1}{n_i}$$

Hudson's, Nei's, Nei and Chesser's and Hedrick's $F$-statistics must be computed by the user as:

$$H_{st} = 1 - \frac{H_s}{H_e}$$

$$G_{st} = 1 - \frac{H_s}{\tilde{H}_t}$$

$$\hat{G}_{st} = 1 - \frac{\hat{H}_s}{\hat{H}_t}$$

$$\hat{G}'_{st} = \frac{\hat{G}_{st}(k_e - 1 + \hat{H}_s)}{(k_e - 1)(1 - \hat{H}_s)}$$

In contrast, Jost's $D$ is computed directly using the relation:

$$D_j = \frac{(\hat{H}_t - \hat{H}_s)}{1 - \hat{H}_s} \cdot \frac{h_e}{k_e - 1}$$

where:

$$\hat{H}_s = \frac{1}{k_e} \sum_{i}^{k_e} H'_i$$

$$\hat{H}_t = 1 - \left( \sum_{i}^{A_e} \frac{1}{k_e} \sum_{j}^{k_e} p_{ij} \right)^2 + \frac{\hat{H}_s}{2\tilde{n}k_e}$$

$H_i'$ (as defined above) is the biaised heterogyzosity, or the average number of pairwise differences, in population $i$. The statistics are not defined if $k_e < 2$.

### 1.4 Allele status

The class `AlleleStatus` is dedicated to analyze the qualitative pattern of allele frequencies over several population. The following categories are defined:

| Method | Explanation |
|---|---|
| `AlleleStatus::num_pop()` | Number of populations |
| `AlleleStatus::fixed()` | Number of fixed alleles † |
| `AlleleStatus::shared_sl()` | Number of shared alleles † |
| `AlleleStatus::shared_ss()` | Number of shared polymorphisms † |
| `AlleleStatus::specific()` | Number of population-specific alleles ‡ |
| `AlleleStatus::specific()` | — only derived alleles ‡ |

† Also available for all population pairs.

‡ Also available for all populations.

If several sites are loaded, the sums are available as separate methods. A fixed allele is at frequency 0 in a population and at relative frequency 1 in another population. A shared allele is present in two populations, but might be fixed in either or both. A shared polymorphism is like a shared allele, but must be segregating in both populations. A population-specific allele is present in only one population.

## 2 Multi-site statistics with unphased data

These statistics are computed by the `Diversity1` class using several sites but only considering allele frequencies.

### 2.1 Basic statistics

They are computed on the fly by `Diversity1::load()` and updated at each loaded site.

| Stat | Accessing method | Explanation |
|---|---|---|
| $L$ | `Diversity1::num_sites()` | Number of loaded sites †‡ |
| $n_m$ | `Diversity1::nsmax()` | Maximal number of exploitable samples † |
| $S$ | `Diversity1::S()` | Number of loaded sites † |
| $\pi$ | `Diversity1::Pi()` | Diversity |
| $\eta$ | `Diversity1::eta()` | Minimal number of mutations † |
| $\Upsilon$ | `Diversity1::singletons()` | Total number of singletons |
| $\hat{\theta}_t$ | `Diversity1::thetaT()` | Tajima's $\theta$ estimator |
| $D_{a,ij}$ | `Diversity1::Da()` | Net pairwise distance $D_a$ ♯ |
| $D_{xy,ij}$ | `Diversity1::Dxy()` | Net pairwise distance $D_{xy}$ ♯ |
| $P_m$ | `Diversity1::pM(i,j)` | $P$-value of Li's MFDM test |

† A version is also available for orientable sites only.

‡ Also available for the number of sites used in Li's MDFM test.

♯ Provided for the first pair of populations ($i = 1$ and $j = 2$).

$$\pi = \sum_{i}^{S} H_{e,i}$$

$$\eta = \sum_{i}^{S} A_{e,i} - 1$$

$$\Upsilon = \sum_{i}^{S} \epsilon_i - 1$$

$$\hat{\theta}_t = \sum_{i}^{S} 1 - \sum_{j}^{A_{e,i}} \frac{c_{ij}(c_{ij}-1)}{n_{s,i}(n_{s,i}-1)} \quad \text{(Tajima 1983)}$$

$$D_{xy,ij} = \sum_{k}^{S} H'_{kij}$$

$$D_{a,ij} = \sum_{k}^{S} H'_{kij} - \frac{H_{e,ki} + H_{e,kj}}{2}$$

$$P_m = \min_{i}^{S} \left( \begin{cases} \text{undefined} & \text{if } d_i < n_{s,i}/2 \\ 1 & \text{if } d_i = n_{s,i}/2 \\ \frac{2(n_{s,i}-d_i)}{n_{s,i}-1} & \text{otherwise} \end{cases} \right) \quad \text{(Li 2011)}$$

with:

$n_{s,i}$ – the number of exploitable samples for site $i$

$H_{e,i}$ – the unbiased heterozygosity of site $i$

$H_{e,ij}$ – the unbiased heterozygosity of site $i$ for population $j$ only

$A_{e,i}$ – the number of alleles of sites $i$

$\epsilon_i$ – the number of singletons at site $i$

$c_{ij}$ – the absolute frequency of allele $j$ in site $i$

$H'_{kij}$ – the average number of pairwise differences at site $k$ between populations $i$ and $j$

$d_i$ – the sum of frequencies of derived alleles at site $i$

## 2.2 Neutrality tests without outgroup

The method `Diversity1::basic()` computes the following neutrality tests and associated statistics:

| Stat | Accessing method | Explanation |
|---|---|---|
| $n_e$ | `Diversity1::nseff()` | Average number of exploitable samples |
| $\hat{\theta}_w$ | `Diversity1::thetaW()` | $\theta$ estimator based on $S$ |
| $D_t$ | `Diversity1::D()` | Tajima's $D$ |
| $D_\eta$ | `Diversity1::Deta()` | Tajima's $D$ using $\eta$ for $S$ |
| $D^*_{fl}$ | `Diversity1::Dstar()` | Fu and Li's $D$ without an outgroup |
| $F^*_{fl}$ | `Diversity1::Fstar()` | Fu and Li's $F$ without an outgroup |

$$n_e = \frac{1}{L}\sum_i^L n_{s,i}$$

$$\hat{\theta}_w = \frac{S}{a_1} \quad \text{(Watterson 1975)}$$

$$D_t = \frac{\pi - \hat{\theta}_w}{\sqrt{\text{var}(\pi - \hat{\theta}_w)}} \quad \text{(Tajima 1989)}$$

$$D_\eta = \frac{\pi - \eta}{\sqrt{\text{var}(\pi - \hat{\theta}_w)}}$$

with:

$$\text{var}(\pi - \hat{\theta}_w) = e_1 S + e_2 S(S-1)$$

using:

$n'_e = \text{round}(n_e)$

$a_1 = \sum_i^{n'_e - 1} \frac{1}{i}$

$a'_1 = \sum_i^{n'_e} \frac{1}{i}$

$a_2 = \sum_i^{n'_e - 1} \frac{1}{i^2}$

$b_1 = \frac{n_e + 1}{3(n_e - 1)}$

$b_2 = \frac{2(n_e^2 + n_e + 3)}{9 n_e (n_e - 1)}$

$c_1 = b_1 - \frac{1}{a_1}$

$c_2 = b_2 - \frac{n_e + 2}{a_1 n_e} + \frac{a_2}{a_1^2}$

$$D^*_{fl} = \frac{\frac{n_e}{n_e-1}\eta - a_1\Upsilon}{\sqrt{u_{d*}\eta + v_{d*}\eta^2}} \quad \text{(Fu \& Li 1993)}$$

$$F^*_{fl} = \frac{\Pi_n - \frac{n_e-1}{n_e}S_s}{\sqrt{u_{d*}\eta + v_{d*}\eta^2}} \quad (id.)$$

using:

$c_n = \frac{2[n_e a_1 - 2(n_e-1)]}{(ne-1)(n_e-2)}$ if $n_e > 2$ (otherwise: 1)

$d_n = c_n + \frac{n_e-2}{(n_e-1)^2} + \frac{2}{n_e-1} \cdot \left(\frac{3}{2} - \frac{2a'_1-3}{n_e-2} - \frac{1}{n_e}\right)$

$v_{d*} = \frac{1}{{a_1}^2+a_2}\left[\left(\frac{n_e}{n_e-1}\right)^2 a_2 + {a_1}^2 d_n - \frac{2n_e}{(n_e-1)^2}a_1(a_1+1)\right]$

$u_{d*} = \frac{n_e}{n_e-1}(a_1 - \frac{n_e}{n_e-1}) - v_{d*}$

$\Pi_n = \frac{2}{n_e(n_e-1)}\sum_i^S H'_i$ where $H'_i$ is the average number of pairwise differences for site $i$

$v_{f*} = \frac{1}{{a_1}^2+a_2}\left[d_n + \frac{2({n_e}^2+n_e+3)}{9n_e(n_e-1)} - \frac{2}{n_e-1}(4a_2 - 6 + \frac{8}{n_e})\right]$

$u_{f*} = \frac{1}{a_1}\left[\frac{n_e}{n_e-1} + \frac{n_e+1}{3(n_e-1)} - \frac{4}{n_e(n_e-1)} + \frac{2(n_e+1)}{(n_e-1)^2} \cdot (a'_1 - \frac{2n_e}{n_e+1})\right] - v_{f*}$

**Note** It is up to the user to skip computations in cases where data are not computable, that is when $n_e < 2$.

## 2.3 Neutrality tests with outgroup

The method `Diversity1::oriented()` assumes that loaded data contain at least one outgroup with exploitable data and computes the following neutrality tests and associated statistics:

| Stat | Accessing method | Explanation |
|---|---|---|
| $n_o$ | `Diversity1::nseffo()` | Average number of samples for orientable sites |
| $\hat{\theta}_l$ | `Diversity1::thetaH()` | θ estimator based on the number of mutations |
| $\hat{\theta}_h$ | `Diversity1::thetaH()` | θ estimator based on derived alleles |
| $H_{fw}$ | `Diversity1::Hns()` | Fay and Wu's $H$, unstandardized |
| $H'_{fw}$ | `Diversity1::Hsd()` | Fay and Wu's $H$, standardized |
| $E$ | `Diversity1::E()` | Zeng *et al.*'s $E$ |
| $D_{fl}$ | `Diversity1::Dfl()` | Fu and Li's $D$ |
| $F_{fl}$ | `Diversity1::F()` | Fu and Li's $F$ |

$$n_o = \frac{1}{S_o}\sum_i^{S_o} n_{s,i}$$

$$\hat{\theta}_h = \frac{2}{n_{mo}(n_{mo}-1)} \sum_{i}^{n_{mo}-1} i^2 \xi_i$$

$$\hat{\theta}_l = \frac{1}{n_{mo}} \sum_{i}^{n_{mo}-1} i \xi_i$$

$$H_{fw} = \pi \frac{S_o}{S} - \hat{\theta}_h \quad \text{(Fay \& Wu 2000)}$$

$$H'_{fw} = \frac{\pi \frac{S_o}{S} - \hat{\theta}_l}{\sqrt{V_z}} \quad \text{(Zeng } et\ al. \text{ 2006)}$$

$$E = \frac{\hat{\theta}_l - \hat{\theta}_w}{\sqrt{V_e}} \quad (id.)$$

using:

$S_o$ – the number of loaded orientable sites, assuming they are all polymorphic

$n_{mo}$ – the maximal number of samples over orientable sites

$n'_o = \text{round}(n_o)$

$\xi_i$ – the number of sites for which $d = i$

$a_{1o} = \sum_i^{n'_o-1} \frac{1}{i}$

$a'_{1o} = \sum_i^{n'_o} \frac{1}{i}$

$a_{2o} = \sum_i^{n'_o-1} \frac{1}{i^2}$

$a'_{2o} = \sum_i^{n'_o} \frac{1}{i^2}$

$\hat{\theta}_s = \frac{S_o}{a_{1o}}$

$\hat{\theta}'_s = \frac{S_o(S_o-1)}{{a_{1o}}^2+a_{2o}}$

$V_z = \frac{n_o-2}{6(n_o-1)}\hat{\theta}_s + \frac{18{n_o}^2(3n_o+2)a'_{2o}-(88{n_o}^3+9{n_o}^2-13n_o+6)}{9n_o(n_o-1)^2}\hat{\theta}'_s$

$V_e = \left[\frac{n_o}{2(n_o-1)} - \frac{1}{a_{1o}}\right]\hat{\theta}_s + \left[\frac{a_{2o}}{{a_{1o}}^2} + 2a_{2o}\left(\frac{n_o}{n_o-1}\right)^2 - \frac{2(n_o a_{2o}-n_o+1)}{a_{1o}(n_o-1)} - \frac{3n_o+1}{n_o-1}\right]\hat{\theta}'_s$

$$D_{fl} = \frac{\eta_o - a_{1o}\Upsilon_d}{\sqrt{u_d\eta_o + v_d{\eta_o}^2}} \quad \text{(Fu \& Li 1993)}$$

$$F_{fl} = \frac{\Pi_{no} - \Upsilon_d}{\sqrt{u_f\eta_o + v_f{\eta_o}^2}} \quad (id.)$$

with:

$c_{no} = \frac{2[n_o a_{1o} - 2(n_o - 1)]}{(no-1)(n_o-2)}$ if $n_o > 2$ (otherwise: 1)

$A_{d,i}$ – the number of derived alleles at site $i$

$\epsilon_{d,i}$ – the number of derived singletons at site $i$

$\eta_o = \sum_i^{S_o} A_{d,i} - 1$

$\Upsilon_d = \sum_i^{S_o} \epsilon_{d,i}$

$\Pi_{no} = \frac{2}{n_o(n_o-1)} \sum_i^{S_o} H'_i$ where $H'_i$ is the average number of pairwise differences for site $i$

$v_d = 1 + \frac{{a_{1o}}^2}{a_{2o}+{a_{1o}}^2} \cdot \left(c_{no} - \frac{n_o+1}{n_o-1}\right)$

$u_d = a_{1o} - 1 - v_d$

$v_f = \frac{1}{{a_{1o}}^2+a_{2o}} \left[c_{no} + \frac{2({n_o}^2+n_o+3)}{9n_o(n_o-1)} - \frac{2}{n_o-1}\right]$

$u_f = \frac{1}{a_{1o}} \left[1 + \frac{n_o+1}{3(n_o-1)} - \frac{4(n_o+1)}{(n_o-1)^2} \cdot \left(a'_{1o} - \frac{2n_o}{n_o+1}\right)\right] - v_f$

**Note** The same restrictions as for the statistics without outgroup apply. The fact that statistics without outgroup can be computed does not necessarily imply that statistics with outgroup can be computed, as non-orientable sites may be rejected. In contrast, the opposite holds (if statistics with outgroup can be computed, then statistics without outgroup can be computed as well).

## 2.4 Paralog divergence

The method of Innan (2003) is implemented in the class `ParalogPi` which takes standard `Site` objects for which the populations represent paralog classes, and for which each sample represent a paralog copy of an individual, where individuals must be represented by one copy in each class. The class computes the following statistics:

| Stat | Accessing method | Explanation |
|---|---|---|
| $K$ | `ParalogPi.num_paralogs()` | Number of paralogs |
| $n$ | `ParalogPi.num_samples()` | Number of samples |
| $S$ | `ParalogPi.num_sites()` | Number of analyzed sites |
| $S_i$ | `ParalogPi.num_sites(i)` | Number of exploitable sites for paralog $i$ † |
| $S_{ij}$ | `ParalogPi.num_sites(i,j)` | Number of exploitable sites for paralogs $i$ and $j$ † |
| $\pi_{w,i}$ | `ParalogPi.Piw()` | Within-paralog $\pi$ for paralog $i$ |
| $\pi_{b,ij}$ | `ParalogPi.Pib()` | Between-paralog $\pi$ for paralogs $i$ and $j$ |

† A site is exploitable is there if at least two samples with exploitable data for each of the concerned paralogs.

$$\pi_{w,i} = \sum_{s}^{S_i} \frac{2}{n_{si}(n_{si}-1)} \sum_{m}^{n_{si}-1} \sum_{n=m+1}^{n_{si}} \begin{cases} 0 & \text{if } d_{mis} = d_{nis} \\ 1 & \text{otherwise} \end{cases}$$

$$\pi_{b,ij} = \sum_{s}^{S_{ij}} \sum_{k}^{n_p-1} \sum_{l=k+1}^{n_p} \frac{1}{n_{si}n_{sj}} \sum_{m}^{n_{si}} \sum_{n}^{n_{sj}} \begin{cases} 0 & \text{if } d_{mis} = d_{njs} \\ 1 & \text{otherwise} \end{cases}$$

where $n_{si}$ is the number of exploitable samples for paralog $i$ at site $s$, the number of exploitable samples for both paralogs $i$ and $j$ at site $s$ and $d_{mis}$ is the allele of sample $m$ for paralog $i$ at site $s$.

**Note** $\pi_{w,i}$ is not defined if $S_i < 1$ and $\pi_{b,ij}$ is not defined if $S_{ij} < 1$.

# 3 Multi-site statistics with phased data

These statistics are computed by the `Diversity2` class using several sites for which the phase is known. The class `Site` (which implies that the phase is known) must be used.

## 3.1 Basic statistics

They are computed on the fly by `Diversity2::load()` and updated a each loaded site.

| Stat | Accessing method | Explanation |
|---|---|---|
| $n_s$ | `Diversity2::num_samples()` | Number of samples |
| $S$ | `Diversity2::num_sites()` | Number of loaded sites † |
| $S^*$ | `Diversity2::num_orientable()` | — orientable only † |
| $S_c$ | `Diversity2::num_clear()` | — with no missing data † |
| $k$ | `Diversity2::k()` | Average number of pairwise differences |
| $k^*$ | `Diversity2::ko()` | — for orientable sites only |

† They are all assumed to be polymorphic.

$$k = \sum_{i}^{S} H'_i$$

$$k^* = \sum_{i}^{S^*} H'_i$$

with:

$H'_i$ – the average number of pairwise differences for site $i$

## 3.2 Singleton-based statistics

They are computed by `Diversity2::singletonStats()` following Ramos-Onsins & Rozas (2002).

| Stat | Accessing method | Explanation |
|---|---|---|
| $R_2$ | `Diversity2::R2()` | Test based on the number of singletons |
| $R_2^*$ | `Diversity2::R2E()` | — using external mutations |
| $R_3$ | `Diversity2::R3()` | — variant |
| $R_3^*$ | `Diversity2::R3E()` | — variant, using external mutations |
| $R_4$ | `Diversity2::R4()` | — variant |
| $R_4^*$ | `Diversity2::R4E()` | — variant, using external mutations |
| $C_h$ | `Diversity2::Ch()` | Test based on the difference of the number of singletons to its expectation |
| $C_h^*$ | `Diversity2::ChE()` | — using external mutations |

$$R_n = \frac{1}{S}\left(\frac{1}{n_s}\sum_i^{n_s}(\tau_i - k/2)^n\right)^{1/n}$$

$$R_n^* = \frac{1}{S}\left(\frac{1}{n_s}\sum_i^{n_s}(\tau_i^* - k/2)^n\right)^{1/n}$$

$$C_h = \frac{S}{m(S-m)}\left(\sum_i^{n_s}\tau_i - m\right)^2$$

$$C_h^* = \frac{S^*}{m^*(S^*-m^*)}\left(\sum_i^{n_s}\tau_i^* - m^*\right)^2$$

with:

$\tau_i$ – the number of singletons carried by sequence $i$

$\tau_i^*$ – the number of derived singletons carried by sequence $i$

$m = k\frac{n_s}{n_s-1}$

$m^* = k^*\frac{n_s}{n_s-1}$

**Note** Statistics are undefined if there is less than one polymorphic site loaded (less than one polymorphic orientable site for statistics based on the number of derived singleton).

## 3.3 Partition-based statistics

They are computed by `Diversity2::partitionStats()` following Wall (1999).

| Stat | Accessing method | Explanation |
|---|---|---|
| $B$ | `Diversity2::B()` | Wall's $B$ |
| $Q$ | `Diversity2::Q()` | Wall's $Q$ |

$$B = \frac{B'}{S_c - 1}$$

$$Q = \frac{B + n_p}{S_c}$$

where $B'$ is the number of pairs of adjacent sites that induce the sample partition and $n_p$ is the number of distinct partitions induced by all sites.

**Note** Only sites without any missing data are considered, and $B$ and $Q$ are not defined if $S_c < 2$.

## 4 Haplotype analysis

The class `Haplotypes` allows to identify haplotypes from a set of sites. By default, the method `Haplotypes::find_haplotypes()` ignores samples with missing data. Samples without missing data are affected to haplotypes that all differ at at least one site. The method `Haplotypes::impute_haplotypes()` attempts to affect each samples with some (small) number of missing data to one of the haplotypes identified by `Haplotypes::find_haplotypes()`. This class can generate a `Site` instance for computing site statistics (which are described above).

In this section we discuss statistics available in the `Haplotypes` class, plus the `Fs()` function.

| Stat | Accessing method | Explanation |
|---|---|---|
| $K$ | `Haplotypes::num_haplotypes()` | Number of haplotypes |
| $D_{ij}$ | `Haplotypes::get_dist()` | Differences between haplotypes $i$ and $j$ |
| $F_{st}$ | `Haplotypes::Fst()` | Population fixation index |
| $K_{st}$ | `Haplotypes::Hst()` | Population fixation index |
| $S_{nn}$ | `Haplotypes::Snn()` | Nearest neighbor statistics |
| $F_s$ | `Fs()` | Fu's $F_s$ |

The method `Haplotypes::compute_dist()` generates the matrix of $D_{ij}$ values, and `Haplotypes::stats()` computes the fixation indexes, and `Haplotypes::Snn()` computes $S_{nn}$ directly (on the fly).

$$F_{st} = 1 - \frac{H_w}{H_b} \quad \text{(Hudson } et\ al. \text{ 1992b)}$$

where $H_w$ is the average number of pairwise differences of samples within populations and $H_b$ is the average number of pairwise differences of samples between populations.

$$K_{st} = 1 - \frac{K_s}{K_t} \quad \text{(Hudson } et\ al. \text{ 1992a)}$$

with:

$$K_s = \frac{1}{n_s} \sum_{i}^{k} \frac{K_i}{n_i - 1}$$

$$K_t = \frac{1}{n_s(n_s - 1)} \sum_{i}^{n_s - 1} \sum_{j=i+1}^{n_s} D_{ij}$$

where $n_s$ is the total number of samples, $k$ is the number of populations, $n_i$ is the number of samples in population $i$, $K_i$ is the sum of number of pairwise differences between samples of population $i$. Note that $K_s$ is half the average number of pairwise differences within populations, and $K_t$ is half the average number of pairwise differences in the total.

$$S_{nn} = \frac{1}{n_s} \sum_{i}^{n_s} \frac{N_i^*}{N_i} \quad \text{(Hudson 2000)}$$

where $N_i$ is the number of "nearest neigbhor" of sample $i$ (that is, sequences that have the minimum of pairwise differences), and $N_i^*$ is the number of nearest neighbors that belong to the population of sample $i$.

$$F_s = \ln \frac{S'}{1 - S'} \quad \text{(Fu 1997)}$$

with:

$$S' = \sum_{i=K}^{n_s} \frac{|S_i| \pi^k}{\sum_{i=0}^{n_s - 1} \pi + i}$$

where $S_{nk}$ are the Stirling numbers of the first kind.

**Note** $F_{st}$ is not defined if there is no polymorphism at all, $K_{st}$ is not if there is no between-population polymorphism, or no within- or between-population pairs at all. $S_{nn}$ is not defined if there is less than two samples. $F_s$ is not defined if there is no polymorphism and cannot be computed if the number of samples is over a given threshold.

## 5 Linkage disequilibrium analyses

The analyses based on linkage disequilibrium are available through three different classes: `PairwiseLD` for processing a single pair of sites, `MatrixLD` for processing all pairs within a set of sites, and `Rd`, which is devoted to a single statistic ($\bar{r}_d$).

## 5.1 Pairwise linkage disequilibrium

The class `PairwiseLD` processes a single pair of sites, which will be represented by indices $i$ and $j$. The method `PairwiseLD.process()` determines the following variables:

| Stat | Accessing method | Explanation |
|---|---|---|
| $A_i$ | `PairwiseLD::num_alleles1()` | Number of alleles for site $i$ |
| $A_j$ | `PairwiseLD::num_alleles2()` | Number of alleles for site $j$ |
| $n_s$ | `PairwiseLD::nsam()` | Number of analyzed samples |
| $P_{im}$ | `PairwiseLD::freq1()` | Absolute frequency of allele $m$ at site $i$ |
| $P_{jn}$ | `PairwiseLD::freq2()` | Absolute frequency of allele $n$ at site $j$ |
| $P_{ij,mn}$ | `PairwiseLD::freq()` | Absolute frequency of genotype $\{m, n\}$ |

The relative frequencies are:

$$p_{im} = {}^{P_{im}}/_{n_s}$$

$$p_{jn} = {}^{P_{jn}}/_{n_s}$$

$$p_{ij,mn} = {}^{P_{ij,mn}}/_{n_s}$$

The method `PairwiseLD.compute()` computes linkage disequilibrium statistics for a given pair of alleles at the two sites, say $a$ at site $i$ and $b$ at site $j$:

| Stat | Accessing method | Explanation |
|---|---|---|
| $D_{ij,ab}$ | `PairwiseLD::D()` | Standard linkage disequilibrium $D$ |
| $D'_{ij,ab}$ | `PairwiseLD::Dp()` | Standardized linkage disequilibrium |
| $r_{ij,ab}$ | `PairwiseLD::r()` | Correlation coefficient |
| ${r_{ij,ab}}^2$ | `PairwiseLD::rsq()` | Squared correlation coefficient |

$$D_{ij,ab} = p_{ij,ab} - p_{im}p_{jn}$$

$$D'_{ij,ab} = \frac{D_{ij,ab}}{D^*_{ij,ab}} \quad \text{(Lewontin 1964)}$$

$$\text{where } D^*_{ij,ab} = \begin{cases} \min[p_{im}p_{jn}, (1-p_{im})(1-p_{jn})] & \text{if } D_{ij,ab} < 0 \\ \min[p_{im}(1-p_{jn}), (1-p_{im})p_{jn}] & \text{otherwise} \end{cases}$$

$$r_{ij,ab} = \frac{D_{ij,ab}}{\sqrt{p_{im}(1-p_{im})p_{jn}(1-p_{jn})}} \quad \text{(Hill \& Robertson 1968)}$$

**Note** Statistics are not computed if there is no valid samples (samples must be exploitable at both sites), no polymorphism, or if the pairwise comparison does not pass the criteria fixed by options to the `Pairwise::process()` method.

## 5.2 Linkage disequilibrium matrix

The class `MatrixLD` processes all pairs from a set of `Site` instances. After loading all sites and computing linkage disequilibrium using `MatrixLD::computeLD()`, the following variables are available:

| Stat | Accessing method | Explanation |
|---|---|---|
| $n_s$ | `MatrixLD::nsam()` | Number of analyzed samples |
| $n_t$ | `MatirxLD::num_tot()` | Total number of processed pairs of sites |
| $n_p$ | `MatrixLD::num_pairs()` | Number of accepted pairs of sites † |
| $n_{ap}$ | `MatrixLD::num_alleles()` | Total number of pairs of alleles |
| $d_{ij}$ | `MatrixLD::distance()` | Distance between sites $i$ and $j$ |

† Sites are filtered according to user-provided thresholds (number of samples used and allele frequency).

The linkage disequilibrium statistics can be accessed through the class `PairwiseLD` which is provided for all accepted pairs of sites.
The method `MatrixLD::computeStats()` computes neutrality tests based on pairwise linkage disequilibrium.

| Stat | Accessing method | Explanation |
|---|---|---|
| $n'_{ap}$ | `MatrixLD::num_allele_pairs()` | Number of used pairs of alleles † |
| $n''_{ap}$ | `MatrixLD::num_allele_pairs_adj()` | Number of used pairs of alleles ‡ |
| $Z_{nS}$ | `MatrixLD::ZnS()` | Kelly's $Z_{nS}$ |
| $Z^*_{nS}$ | `MatrixLD::ZnS_star1()` | Kelly's $Z^*_{nS}$ |
| ${Z^*_{nS}}^*$ | `MatrixLD::ZnS_star2()` | Kelly's ${Z^*_{nS}}^*$ |
| $Z_a$ | `MatrixLD::Za` | Rozas *et al.*'s $Z_a$ |
| $Z_Z$ | `MatrixLD::ZZ` | Rozas *et al.*'s $Z_Z$ |

† $n'_{ap}$ may be smaller than $n_{ap}$ if there are sites with more than two alleles (see remark below for the treatment of sites with more than two alleles).

‡ $n''_{ap}$ is the same as $n'_{ap}$ but considering pairs of adjacent sites.

The user may select three strategies for processing sites that exhibit more than two alleles:

1. Ignore all pairs for which one site has more than two alleles.
2. Use the most frequent allele at each site.
3. Use all possible pairs of alleles.

$$Z_{nS} = \frac{1}{n'_{ap}} \sum_{i}^{n'_{ap}} {r_i}^2 \quad \text{(Kelly 1997)}$$

$$Z^*_{nS} = Z_{nS} + 1 - \frac{1}{n'_{ap}} \sum_i^{n'_{ap}} {D'_i}^2 \quad (id.)$$

$$Z^*_{nS}{}^* = \frac{Z_{nS}}{\frac{1}{n'_{ap}} \sum_i^{n'_{ap}} {D'_i}^2} \quad (id.)$$

$$Z_a = \frac{1}{n''_{ap}} \sum_i^{n'_{ap}} \begin{cases} {r_i}^2 & \text{for adjacent sites} \\ 0 & \text{otherwise} \end{cases} \quad \text{(Rozas } et\ al. \text{ 2001)}$$

$$Z_Z = Z_a - Z_{nS} \quad (id.)$$

where ${r_i}^2$ is the squared correlation coefficient for a given pair of alleles of a pair of sites, noted ${r_{ij,ab}}^2$ above, and $D'_i$ is Lewontin's $D'$ value for the same pair of sites ($D'_{ij,ab}$).

**Note** Kelly's and Rozas *et al.*'s statistics are not defined if the number of processed pairs $n'_{ap}$ and $n''_{ap}$, respectively, is 0.

## 5.3 $R_{min}$ and $\bar{r}_d$

$R_{min}$ (Hudson & Kaplan 1985) is computed by the method `MatrixLD::computeRmin()` which exposes the following values:

| Accessing method | Explanation |
|---|---|
| `MatrixLD::Rmin()` | $R_{min}$, the minimal number of recombination events |
| `MatrixLD::Rmin_num_sites()` | Number of sites used to compute $R_{min}$ |
| `MatrixLD::Rmin_left()` | Get an recombination interval (left bound) |
| `MatrixLD::Rmin_right()` | Get an recombination interval (rigth bound) |

For a rigorous description of the algorithm for computing $R_{min}$, see Appendix 2 of Hudson & Kaplan (1985). In short, $R_{min}$ is computed as the smallest possible number of non-overlapping intervals between sites violating the four-gamete rule (if the four gametes are present in the sample for a given pair of sites, then a recombination must have occurred between these sites). In case of oriented sites, the four-gamete rule becomes the three-gamete rule and the approach is unchanged. The positions of sites defining the final set of non-overlapping intervals are available as `MatrixLD::Rmin_left()` and `MatrixLD::Rmin_right()`.

**Note** If there are less than two sites, `MatrixLD::Rmin_num_sites()` is set to 0. If `MatrixLD::Rmin_num_sites()` is less than 2, all the other $R_{min}$ variables are defined.

The class `Rd` is specifically dedicated to the $\bar{r}_d$ statistic.

| Stat | Accessing method | Explanation |
|---|---|---|
| $\bar{r}_d$ | `Rd::rD()` | Multilocus linkage disequilibrium † |

† $\bar{r}_d$ is available for per population and for the total sample.

$$\bar{r}_d = \frac{\sum_i^{S-1} \sum_{j=i+1}^{S} \mathrm{cov}(d_i, d_j)}{\sum_i^{S-1} \sum_{j=i+1}^{S} \sqrt{\mathrm{var}(d_i)\,\mathrm{var}(d_j)}}$$

where $S$ is the number of loaded sites (which must all be polymorphic), $\mathrm{var}(d_i)$ is the variance of the genetic distance between samples at site $i$ (skipping samples that have missing data at this site), and $\mathrm{cov}(d_i, d_j)$ is the covariance of the genetic distance between samples at sites $i$ and $j$ (skipping samples with missing data at either site). The genetic distance is 0 or 1 when alleles are loaded, and 0, 1 or 2 when genotypes are loaded.

**Note** $\bar{r}_d$ is always computed, but its value is not defined if there is not polymorphism (this may happen easily within populations).

# 6 Extended haplotype heterozygosity

EHH statistics are computed by two classes, `EHH` for phased data and `EHHG` for unphased diploid data (using observed heterozygosity). Both `EHH` and `EHHG` classes process full sites. Individuals must match over sites in both cases. `EHH` ignores genotypes while `EHHG` requires them. Both classes require that a core site is loaded first and then distant sites. In this section, all EHH statistics are renamed using a variety of Greek letters (the correspondance can be found in tables listing EHH statistics).

## 6.1 EHH with phased data

**Variables** Here are the variables used for computing EHH statistics:

| Stat | Explanation |
|---|---|
| $L$ | Number of sites (excluding core) |
| $K_0$ | Number of core haplotypes |
| $K_s$ | Number of haplotypes at site $s$ |
| $R$ | Number of populations |
| $n_{ir}$ | Number of samples of core haplotype $i$ in population $r$ |
| $n_{i*}$ | Number of samples of core haplotype $i$ in total |
| $n_{*r}$ | Number of samples in population $r$ |
| $n_{**}$ | Total number of samples |
| $P_{sijk}$ | Probability that sample $j$ of core haplotype $i$ is haplotype $k$ at site $s$ |
| $D_{sij}$ | Allele index for sample $j$ of core haplotype $i$ at site $s$ |
| $H_{sk}$ | Allele index for haplotype $k$ at site $s$ |
| $A_s$ | Number of alleles at site $s$ |
| $f_{sak}$ | Frequency (at current site $s$) of allele $a$ in samples of haplotype $k$ at site $s-1$ |
| $d_s$ | Distance of site $s$ from core site (note: $d_s \geq 0$, $d_s \geq d_{s-1}$ and $d_0 = 0$) |

For readability, iterators names are used consistently in this section and the next one:

| Iterator |
|---|
| $1 \leq s \leq L$ |
| $1 \leq i \leq K_0$ |
| $1 \leq j \leq n_{i*}$ |
| $1 \leq k \leq K_s$ |
| $1 \leq r \leq R$ |
| $1 \leq a \leq A_s$ |

In addition, the following indicator functions are defined:

$$I(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases}$$

$$V(s, i, j, k) = \begin{cases} 0, & \text{if } D_{sijk} \text{ is missing} \\ 1, & \text{otherwise} \end{cases}$$

$$M(i, j, r) = \begin{cases} 1, & \text{if sample } i \text{ of population } j \text{ belongs to population } r \\ 0, & \text{otherwise} \end{cases}$$

With $*$ used as a wild card for representing the whole sample, we have $M(i, j, *) = 1$

**Accessory statistics** Below is the list of accessory statistics computed by the class `EHH`:

| Stat | Method | Explanation |
|---|---|---|
| $K_0$ | `EHH::Kcore()` | Number of core haplotypes † |
| $K_s$ | `EHH::K()` | Current number of haplotypes ‡ |
| $R$ | `EHH::R()` | Number of populations † |
| $n_{**}$ | `EHH::nsam()` | Number of samples †§♯ |
| | `EHH::ncur()` | Number of exploitable samples at last distant site ‡§ |
| | `EHH::num_decay()` | Number of EHH values that reached decay threshold ‡ |
| | `EHH::num_intrg()` | Number of iHH values still incrementing ‡ |
| | `EHH::num_intrg_c()` | Number of iHHc values still incrementing ‡ |
| | `EHH::intrg_S()` | True if iES is still incrementing ‡ |

† Available after the core site has been loaded, if there was exploitable polymorphism.

‡ Available after at least one distant site has been loaded, also requiring exploitable polymorphism.

§ Available for population $r$ ($n_{*r}$), for core haplotype $i$ ($n_{i*}$), for population $r$ and haplotype $i$ jointly ($n_{ir}$), and for the total ($n_{**}$).

♯ Samples with missing data at core are ignored throughout all loaded distant sites.

**EHH statistics** Below is the list of EHH statistics computed by the class `EHH`:

| Stat | Method | Explanation |
|---|---|---|
| $\Psi_{sir}$ | `EHH::EHHi()` | EHH for core haplotype $i$ † |
| $\Psi'_{sir}$ | `EHH::EHHc()` | EHH for the complement of core haplotype $i$ † |
| $\Upsilon_{sir}$ | `EHH::rEHH()` | EHHr for core haplotype $i$ † |
| $\psi_{sir}$ | `EHH::iHH()` | iHH for core haplotype $i$ ‡ |
| $\psi'_{sir}$ | `EHH::iHHc()` | iHH for the complement of core haplotype $i$ ‡ |
| $\lambda_{sir}$ | `EHH::iHS()` | iHS for core haplotype $i$ ‡ |
| $\Phi_{sr}$ | `EHH::EHHS()` | EHHS (for whole site) § |
| $\phi_{sr}$ | `EHH::iES()` | iES ratio § |
| $\delta^t_{ir}$ | `EHH::decay()` | EHH decay distance for core haplotype $i$ at threshold $t$ ♯ |
| $\bar{\delta}^t_r$ | `EHH::davg()` | average of EHH decay distance with threshold $t$ ♯ |
| $\dot{\delta}^t_r$ | `EHH::dmax()` | maximum of EHH decay distance with threshold $t$ ♯ |
| $\gamma^t_r$ | `EHH::decayS()` | EHHS decay distance for at threshold $t$ ♯ |

– Note: $s$ stands for the current site and and $r$ for one population. All statistics that are expressed for a population are also available for the total.

† Following Sabeti *et al.* (2002).

‡ Following Voight *et al.* (2006).

§ Following Tan *et al.* (2007).

♯ After Ramírez-Soriano *et al.* (2008).

**Computation** Here, we will allow for missing data, therefore we need to compute the probabilities of haplotypes taking into account allele frequencies for each loaded distant site. For a starter, if $k > K_{s-1}$, then $P_{(s-1)ijk} = P_{(s-1)ijp}$ where $p$ is the parent of $k$ (this is how we find probabilities of haplotypes at previous sites by following the haplotype hierarchy).

$$f_{sak} = \sum_{i=1}^{K_0} \sum_{j=1}^{n_{i*}} I(D_{sij}, a) \cdot P_{(s-1)ijk} / \sum_{i=1}^{K_0} \sum_{j=1}^{n_{i*}} V(D_{sij}) \cdot P_{(s-1)ijk}$$

$$P_{0ijk} = \begin{cases} 1, & \text{if } i \neq k \\ 0, & \text{otherwise} \end{cases}$$

$$P_{sijk} = \begin{cases} P_{(s-1)ijk} \cdot f_{sD_{sij}k}, & \text{if } V(D_{sij}) = 0 \\ P_{(s-1)ijk}, & \text{if } D_{sij} = H_{sk} \\ 0, & \text{otherwise} \end{cases}$$

The EHH statistics are computed as follows (note that statistics computed for a population $r$ can also be computed for the total):

$$\Psi_{sir} = \sum_{k=1}^{K_s} \left( \frac{1}{n_{ir}} \sum_{j=1}^{N_{i*}} M(i,j,r) \cdot p_{sijk} \right)^2$$

$$\Psi'_{sir} = \sum_{k=1}^{K_s} \left( \frac{1}{n_{*r} - n_{ir}} \sum_{i' \neq i}^{K_0} \sum_{j=1}^{N_{i*}} M(i,j,r) \cdot p_{si'jk} \right)^2$$

$$\Upsilon_{sir} = \Psi_{sir} / \Psi'_{sir}$$

IHH:

$$\psi_{sir} = \sum_{s'=1}^{s} \frac{(d_{s'} - d_{s'-1})(\Psi_{s'ir} + \Psi_{(s'-1)ir})}{2}$$

$$\psi'_{sir} = \sum_{s'=1}^{s} \frac{(d_{s'} - d_{s'-1})(\Psi'_{s'ir} + \Psi'_{(s'-1)ir})}{2}$$

$$\lambda_{sir} = \ln \left( \frac{\psi'_{sir}}{\psi_{sir}} \right)$$

EHHS and iES:

$$\Phi_{sr} = \frac{1 - \frac{n_{*r}}{n_{*r}-1} \left[ 1 - \frac{1}{{n_{*r}}^2} \sum_{k=1}^{K_s} \left( \sum_{i=1}^{K_0} \sum_{j=1}^{n_{i*}} M(i,j,r) \cdot p_{ijk} \right)^2 \right]}{1 - \frac{n_{*r}}{n_{*r}-1} \left( 1 - \frac{1}{{n_{*r}}^2} \sum_{i=1}^{K_0} {n_{ir}}^2 \right)}$$

$$\phi_{sr} = \sum_{s'=1}^{s} \frac{(d_{s'} - d_{s'-1})(\Phi_{s'r} + \Phi_{(s'-1)r})}{2}$$

Decay statistics:

$$\delta_{ir}^{t} = d_s \text{ such as } \Psi_{(s-1)ir} > t \text{ and } \Psi_{sir} \leq t$$

$$\bar{\delta}_{r}^{t} = \frac{1}{N_{*r}} \sum_{i=1}^{K_0} N_{ir} \delta_{ir}^{t}$$

$$\dot{\delta}_{r}^{t} = \max_{i} \delta_{ir}^{t}$$

$$\gamma_{r}^{t} = d_s \text{ such as } \Phi_{(s-1)r} > t \text{ and } \Phi_{sr} \leq t$$

## 6.2 EHH with unphased data

**Variables** Here are the variables used for computing EHHG statistics:

| Stat | Explanation |
|---|---|
| $L$ | Number of sites (excluding core) |
| $K_0$ | Number of core genotypes |
| $K_s$ | Number of genotypes at site $s$ |
| $R$ | Number of populations |
| $n_r$ | Number of samples in population $r$ |
| $n_*$ | Total number of samples |
| $T_{sgh}$ | Probability of $g$ at site $s$ given $h$ at $s-1$ |
| $P_{sig}$ | Probability that sample $i$ is genotype $g$ at site $s$ |
| $D_{si}$ | Genotype index for sample $i$ at site $s$ |
| $d_s$ | Distance of site $s$ from core site (note: $d_s \geq 0$, $d_s \geq d_{s-1}$ and $d_0 = 0$) |

Here are the iterators used while computing EHHG statistics:

| Iterator |
|---|
| $1 \leq s \leq L$ |
| $1 \leq i \leq n_*$ |
| $1 \leq r \leq R$ |
| $1 \leq g \leq K_s$ |
| $1 \leq h \leq K_{s-1}$ |

In addition, the following indicating functions are defined:

$$I(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases}$$

$$V(s,i,k) = \begin{cases} 1, & \text{if } D_{sik} \text{ is not missing} \\ 0, & \text{otherwise} \end{cases}$$

$$M(i,r) = \begin{cases} 1, & \text{if sample } i \text{ belongs to population } r \\ 0, & \text{otherwise} \end{cases}$$

With $*$ used as a wild card for representing the whole sample, we have $M(i,*) = 1$

$$H(s,g) = \begin{cases} 1, & \text{if } g \text{ is homozygote} \\ 0, & \text{otherwise} \end{cases}$$

**Accessory statistics** Below is the list of accessory statistics computed by the class `EHHG`:

| Stat | Method | Explanation |
|---|---|---|
| $K_s$ | `EHHG::K()` | Current number of haplotypes ‡ |
| $R$ | `EHHG::R()` | Number of populations † |
| $n_*$ | `EHHG::nsam()` | Number of samples (including missing data at core) †§ |
| | `EHHG::ncur()` | Number of exploitable samples at last distant site †§ |
| | `EHHG::intrg()` | True if iES is still incrementing ‡ |

† Available after the core site has been loaded, if there was exploitable polymorphism.

‡ Available after at least one distant site has been loaded, also requiring exploitable polymorphism.

§ Available for population $r$ ($n_r$) and for total ($n_*$).

**EHHG statistics** Below is the list of EHH statistics computed by the class `EHHG`:

| Stat | Method | Explanation |
|---|---|---|
| $\Phi_{sr}$ | `EHHG::EHHS()` | EHHS value § |
| $\phi_{sr}$ | `EHHG::iES()` | iES ratio § |
| $\gamma_r^t$ | `EHHG::decay()` | EHHS decay distance ♯ |

– Note: $s$ stands for the current site and and $r$ for one population. All statistics are also available for the total.

§ Following Tan *et al.* (2007).

♯ After Ramírez-Soriano *et al.* (2008).

**Computation** The computation of EHHG statistics is based on the matrix of probabilities of all possible genotypes which is updated at each site.

$$T_{sgh} = \frac{\sum_{i=1}^{n} I(D_{si}, g) \cdot P_{(s-1)ih}}{\sum_{i=1}^{n} P_{(s-1)ih}}$$

$$P_{sig} = \begin{cases} \sum_{h=1}^{K_{s-1}} P_{(s-1)ih} \cdot T_{sgh}, & \text{if } V(D_{si}) = 0 \\ 1, & \text{if } D_{si} = g \\ 0, & \text{otherwise} \end{cases}$$

The initial probability values are determined as follows:

$$P_{0ig} = \begin{cases} \frac{\sum_{j=1}^{n*} I(D_{sj}, g)}{n*}, & \text{if } V(D_{si}) = 0 \\ 1, & \text{if } D_{si} = g \\ 0, & \text{otherwise} \end{cases}$$

The EHHG statistics are computed as follows (note that statistics computed for a population $r$ can also be computed for the total):

$$\Phi_{sr} = \frac{\sum_{i=1}^{n_*} M(i,r) \sum_{g=1}^{K_s} H(s,g) \cdot P_{sig}}{\sum_{i=1}^{n_*} M(i,r) \sum_{g=1}^{K_0} H(0,g) \cdot P_{0ig}}$$

$$\phi_{sr} = \sum_{s'=1}^{s} \frac{(d_{s'} - d_{s'-1})(\Phi_{s'r} + \Phi_{(s'-1)r})}{2}$$

$$\gamma_r^t = d_s \text{ such as } \Phi_{(s-1)r} > t \text{ and } \Phi_{sr} \leq t$$

# 7 Coding site analysis

The analysis of coding data is performed site-by-site. Each coding site is passed through a triplet of `Site` to the class `CodingSite` which performs elementary operations. `CodingSite` computes the following values:

| Stat | Method | Explanation |
|---|---|---|
| $n$ | `Codingsite::ns()` | Number of samples |
| $n_e$ | `Codingsite::nseff()` | Number of exploitable samples † |
| $n_{stop}$ | `Codingsite::nstop()` | Number of stop codons ‡ |
| $L_{NS}$ | `Codingsite::NSsites()` | Estimated number of non-synonymous sites |

† Samples containing missing data at either position of the codon are excluded. Samples presenting a stop codons may be excluded as well, depending of the value of an option.

‡ Returns the number of stop codons even if stop codons are considered as missing data.

In addition, `CodingSite` provides a `Site` instance containing the amino acid value for every sample, and another for codons recoded as a single integer. These objects can be used to compute actual diversity statistics using any other class of the libraty, typically `SiteDiversity`.

The estimated number of non-synonymous sites $L_{NS}$ is computed as:

$$L_{NS} = \frac{1}{n_e} \sum_{i}^{n_e} \sum_{j}^{3} F_{ij} \quad \text{(Nei \& Gojobori 1986)}$$

where $F_{ij}$ is the proportion of nucleotide substitutions affecting the codon of sample $i$ at position $j$ that would lead to an amino acid changes. The possible values of $F_{ij}$ are in principle 0, $1/3$, $2/3$ and 1 (there are three possible nucleotide substitutions). If stop codons are ignored, nucleotide substitutions causing a change to a stop codons are discarded. Otherwise there are considered as non-synonymous changes.

# 8 References


Agapow P.-M & A. Burt. 2001. Indices of multilocus linkage disequilibrium. *Mol. Ecol. Res.* **1**: 101-102.

Fay J. C. & C. I Wu. 2000. Hitchhiking under positive Darwinian selection. *Genetics* **155**: 1405-1413.

Fu Y.-X. 1997. Statistical tests of neutrality of mutations against population growth, hitchhiking and background selection. *Genetics* **147**: 915-925.

Fu Y.-X. & W.-H. Li 1993. Statistical tests of neutrality of mutations. *Genetics* **133**: 693-709.

Hedrick P. W. 2005. A standardized genetic differentiation measure. *Evolution* **17**: 4015-4026.

Hill W. G. & A. Robsetson A. 1968. Linkage disequilibrium in finite populations. *Theor. Popul. Biol.* **38**: 226-231.

Hudson R. R. 2000. A new statistic for detecting genetic differentiation. *Genetics* **155**: 2011-2014.

Hudson R. R., D. D. Boos & N. L. Kaplan. 1992a. A statistical test for detecting geographic subdivision. *Mol. Biol. Evol.* **9**: 138-151.

Hudson R. R & N. L Kaplan. 1985. Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics* **111**: 147-164.

Hudson R. R., M. Slatkin & W. P. Maddison. 1992b. Estimation of levels of gene flow from DNA sequence data. *Genetics* **132**: 583-589.

Innan H. 2003. The coalescent and infinite-site model of a small multigene family. *Genetics* **163**: 803-810.

Jost L. 2008. $G_{st}$ and its relatives do not measure differentiation. *Mol. Ecol.* **17**: 4015-4026.

Kelly J. K. 1997. A test of neutrality based on interlocus associations. *Genetics* **146**: 1197-1206.

Lewontin R. C. 1964. The interaction of selection and linkage. I. General considerations; heterotic models. *Genetics* **49**: 49-67.

Li H.-P. 2011. A new test for detecting recent positive selection that is free from the confounding impacts of demography. *Mol. Biol. Evol.* **28**: 365-375.

Nei, M. & R. K. Chesser. 1983. Estimation of fixation indexes and gene diversities. *Annals Human Genet.* **47**: 253-259.

Nei M. & T. Gojobori. 1986. Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. *Mol. Biol. Evol.* **3**: 418-426.

Ramírez-Soriano A., S. E. Ramos-Onsins, J. Rozas, F. Calafell & A. Navarro. 2008. Statistical power analysis of neutrality tests under demographic expansions, contractions and bottlenecks with recombination. *Genetics* **179**: 555-567.

Ramos-Onsins S. E. & J. Rozas. 2002. Statistical properties of new neutrality tests against population growth. *Mol. Biol. Evol.* **19**: 2092-2100.

Rozas J., M. Gullaud, G. Blandin & M. Aguadé. 2001. DNA variation at the *rp49* gene region of *Drosophila simulans*: evolutionary inferences from an unusual haplotype structure. *Genetics* **158**: 1147-1155.

Sabeti P. C., D. E. Reich, J. M. Higgins, H. Z. P. Levine, D. J. Richter, S. F. Schaffner, S. B. Gabriel, J. V. Platko, N. J. Patterson, G. J. McDonald, H. C. Ackerman, S. J. Campbell, D. Altshuler, R. Cooper, D. Kwiatkowski, R. Ward & E. S. Lander. 2002. Detecting recent positive selection in the human genome from haplotype structure. *Nature* **419**: 832-837.

Tajima T. 1983. Evolutionary relationship of DNA sequences in finite populations. *Genetics* **105**: 437-460.

Tajima T. 1989. Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. *Genetics 123*: 585-595.

Tan K., K. R. Thornton & M. Stoneking. 2007. A new approach for using genome scans to detect recent positive selection in the human genome. *PLoS Biol.* **5**: e171.

Voight B. F., S. Kudaravalli, X. Wen & J. K. Pritchard. 2006. A map of recent positive selection in the human genome. *PLoS Biol* **4**: e772

Wall J. D. 1999. Recombination and the power of statistical tests of neutrality. *Genet. Res.* **74**: 65-79.

Watterson G. 1975. On the number of segretating sites in genetical models without recombination. *Theor. Popul. Biol.* **7**: 256-276.

Weir B. C. & C. C. Cockerham. 1984. Estimating *F*-statistics for the analysis of population structure. *Evolution* **38**: 1358-1370.

Weir B. C. & W. H. Hill. 2002. Estimating *F*-statistics. *Annu. Rev. Genet.* **36**: 721-750.

Zeng K., Y.-X. Fu, S. Shi & C. I Wu. 2006. Statistical tests for detecting positive selection by utilizing high-frequency variants. *Genetics* **174**: 1431-1439.