Definition of the Data Classes Used in the R Package pegas

Emmanuel Paradis

July 10, 2014

1 Introduction

This document describes two R data structures used in the package pegas. It is an update of a previous version (May 13, 2009).

2 Definition of the Class "loci"

2.1 General Structure

An object of class "loci" is a data frame where rows represent individuals and columns are loci and optional additional variables. One of these additional columns may be named "population" depending on whether the data were read in a Genetix file or imported from adegenet. To identify the locus columns, an attribute "locicol" is appended to the data frame: it is a vector of integers. Each locus is a factor; the other columns may be of any type. The class "loci" actually inherits the class "data.frame".

Formally, an object of class "loci" has the following components:

- The class c("loci", "data.frame").
- One or more columns made of factors defining the loci.
- An attribute "locicol" that identifies the indices of the loci; therefore it is a vector of integers of length one or more.
- Zero or more columns made of vectors and/or factors giving additional data.

The rownames and colnames (which are mandatory in a data frame) eventually give the names of the individuals and the variables (including the loci), respectively.

In pegas 0.6, the methods available for the class "loci" are:

```
> methods(class = "loci")
[1] cbind.loci*     edit.loci*      haplotype.loci*
[4] [.loci*         print.loci*     rbind.loci*
[7] summary.loci*

   Non-visible functions are asterisked
```

2.2 Individual Locus as Factor

A locus is coded with a factor which levels specify the different genotypes. From version 0.6, `pegas` can handle phased and unphased genotypes (only the latter were supported in previous versions).

Unphased genotypes are coded with the allele names separated with a forward slash. For instance, with alleles A and B the genotypes will be `"A/A"`, `"A/B"` and `"B/B"` for diploids, `"A/A/A"`, `"A/A/B"`, ..., for triploids, and so on. The convention is that within a genotype the alleles are first sorted with uppercase, then in alphabetical order, so that if a file contains `"A/a"` and `"a/A"`, the latter are changed into `"A/a"` and all are considered to be the same genotype. If a file contains only `"a/A"`, these are changed into `"A/a"` even if this was not in the file. Similarly, if the alleles are a and B, `"a/B"` will be changed into `"B/a"` (because ordering on case is done first).

For phased genotypes, no reordering of alleles is done and `"A|a"` and `"a|A"` are treated as two distinct genotypes.

There are utility functions `getPloidy`, `getAlleles` and `getGenotypes` that extract the levels of ploidy, the observed alleles and genotypes, respectively, for all loci in an object of class `"loci"`. The function `is.snp` tests whether a locus is a SNP (i.e., only two alleles are observed each made of a single character). The function `is.phased` tests whether a genotype is phased.

3 Definition of the Class `"haplotype"`

An object of class `"haplotype"` is a matrix of DNA sequences found to be unique in a possibly larger set. It has an additional attribute that identifies the original individuals (rows) and specifies which haplotype they belong to.

Formally, an object of class `"haplotype"` has the following components:

- The class `c("haplotype", "DNAbin")`.
- A matrix of aligned DNA sequences that follow the class `"DNAbin"` of the package `ape`.
- An attribute `"index"` that gives the indices of the rows of the original DNA matrix that belong to each haplotype. It is a list of length equal to the number of rows of the above matrix, and each of its elements is a vector of integers.

In `pegas` 0.6, the methods available for the class `"haplotype"` are:

```
> methods(class = "haplotype")
[1] [.haplotype*     plot.haplotype*  print.haplotype*
[4] sort.haplotype*

   Non-visible functions are asterisked
```

In this version, the function `haplotype.loci` returns an object of class `"haplotype.loci"` with the following method:

```
> methods(class = "haplotype.loci")
[1] plot.haplotype.loci*

   Non-visible functions are asterisked
```