R documentation

of ‘man/obkData.Rd’

January 11, 2013

`obkData-class` *Formal class "obkData"*

Description

The class `obkData` is a formal (S4) class for storing data collected during outbreaks. This includes:

- individual data (age, sex, onset of symptoms, ...)
- sample data (swabs, serology, accession numbers, ...)
- genetic sequences
- contact information
- clinical data

Objects from the class obkData

`obkData` objects can be created using `new("obkData", ...)`, where '...' can be the following optional arguments (defaults are all `NULL`):

`individuals` a data.frame with a mandatory column named 'individualID', providing unique identifiers for the individuals.

`samples` a data.frame with 3 mandatory columns named 'individualID', 'sampleID', and 'date', providing identifiers for the individuals, for the samples, and dates. Dates must be provided in a way convertible to `Date` (see `?as.Date`). Default format for dates if in character strings is " argument date.format

`clinical` ... - to be filled

`dna` a list of DNA sequences in `DNAbin` or `character` format.

`contacts` ... - to be filled

`trees` a list of phylogenetic trees in the class `multiPhylo` (from the ape package)

Slots

obkData contain the following slots; note that in most cases, it is better to retrieve information via accessors (see below), rather than by accessing the slots manually. Empty slots are all NULL.

individuals: a data.frame containing individual information, with individual labels stored as row names.

samples: a data.frame containing sample information; the first three columns are 'individualID', 'sampleID', and 'date'.

clinical: ...

contacts: an object of the class obkContacts storing contact information.

dna: an object of the class obkSequences storing DNA sequences.

trees: an object of the class multiPhylo storing list of trees.

Methods

Here is a list of methods available for obkData objects. Most of these methods are accessors, that is, functions which are used to retrieve the content of the object. Specific manpages can exist for more complex functions. These are indicated by a '*' symbol next to the method's name. This list also contains methods for conversion from obkData to other classes.

show signature(x = "obkData"): printing of the object.

get.locus signature(x = "obkData"): returns the loci existing in the data.

get.nlocus signature(x = "obkData"): returns the number of loci.

get.dna* signature(x = "obkData"): returns the sequences for a given locus (locus argument, required if more than one locus was sequenced).

get.nsequences signature(x = "obkData"): returns the number of sequences in the data.

get.locus signature(x = "obkData"): returns the names of the loci in the sample.

get.individuals* signature(x = "obkData"): returns the IDs of individuals in a given source of data (argument data, can be 'samples' (default) or 'individuals').

get.nindividuals* signature(x = "obkData"): returns the number of individuals in a given source of data (argument data, can be 'samples' (default) or 'individuals').

get.data* signature(x = "obkData"): search for a matching fields in the object and returns the corresponding values; in the absence of match, NULL is returned. Several values can be provided; they can be names of the slots, or any variable stored within the data.frames samples or individuals.

Thibaut Jombart, Simon Frost, Lulla Opatowski, Paul Birrell, Anne Cori, Marc Baguelin, Caroline Colijn... add your name/email here

## EMPTY OBJECT ## new("obkData")

## INDIVIDUAL INFO ONLY ## new("obkData", individuals=data.frame("individualID"=letters)) new("obkData", individuals=data.frame("individualID"=letters, age=1:26, 1:26))

samp <- data.frame(individualID=c('toto','toto','titi'), sampleID=c(1,3,2), date=c("2001-02-13","2001-03-01","2001-05-25"), swab=c("+","-","+"))

## SAMPLE INFO ONLY ## new("obkData", sample=samp) new("obkData", sample=samp[,c(1:3)]) new("obkData", sample=samp[,c(1:3,4,4,4)])

## SAMPLE & INDIV INFO - MISSING INDIV ## new("obkData", sample=samp[,c(1:3,4,4,4)] , individuals=data.frame("individualID"=letters, age=1:26))

## SAMPLE & INDIV INFO ## ind <- data.frame("individualID"=c("toto","John Doe", "titi"), age=c(20,18,67), sex=c("m","m","?")) new("obkData", sample=samp, ind=ind)

## DNA INFO, NOTHING ELSE ## library(ape) data(woodmouse) dat.dna <- as.list(woodmouse)

new("obkData", dna=dat.dna) # should be empty

## SAMP + DNA INFO ## samp <- data.frame(individualID=c('toto','toto','titi'), sampleID=c(1,3,2), date=c("2001-02-13","2001-03-01","2001-05-25"), swab=c("+","-","+"))

samp <- cbind.data.frame(samp, sequenceID=c(1,2,3))

## sequences given as indices new("obkData", samples=samp, dna=dat.dna) # (note the nice sample ordering)

## sequences given as IDs samp$sequenceID <- c("No304","No306","No305") new("obkData", samples=samp, dna=dat.dna) # (note the nice sample ordering)

## sequences given as IDs, with wrong IDs ## samp$sequenceID <- c("No304","No306","Arrrhhh")
# this would generate a meaningful error ## new("obkData", samples=samp, dna=dat.dna) # (note
the nice sample ordering)

## multiple sequences per individual samp$sequenceID <- c("No304","No306","No305") samp <- samp[c(1,1,2,2,2,3),] samp$sequenceID <- 1:6 new("obkData", samples=samp, dna=dat.dna)

## multiple sequences per individual, locus information samp$locus <- c("gene1","gene2")[c(1,1,1,2,1,2)] new("obkData", samples=samp, dna=dat.dna)

## TEST ACCESSORS ## x <- new("obkData", samples=samp, dna=dat.dna, individuals=ind)

get.locus(x) get.individuals(x) get.individuals(x, "individuals") get.dna(x, locus="gene2") get.nsequences(x) get.data(x, "samples") get.data(x, "individuals") get.data(x, c("date","swab")) table(get.data(x, c("swab", "locus")))

classes

Index