Guide for the janno R package v1.0.0

Contents

1 Read janno files

You can read `.janno` files with

```
my_janno_object <- janno::read_janno(
  path = "path/to/my/janno_file.janno",
  to_janno = TRUE,
  validate = TRUE
)
```

The path argument takes one or multiple file paths or directory paths. `read_janno()` searches recursively for `.janno` files in the directory paths.

Before loading the `.janno` files are validated with `janno::validate_janno()`. You can avoid this potentially time consuming step with `validate = FALSE`.

Usually the `.janno` files are loaded as normal `.tsv` files with every column type set to `character` and then the columns are transformed to the intended types. This transformation can be turned off with `to_janno = FALSE`.

`read_janno()` returns an object of class `janno`. `janno` objects are derived tibbles, so all tidyverse operations can be applied to them. As long as the data layout does not change, they will remain `janno` objects and not be transformed to default tibbles.

2 Validate janno files

You can validate `.janno` files with

```
my_janno_issues <- janno::validate_janno("path/to/my/janno_file.janno")
```

`validate_janno` returns a tibble with issues in the respective `.janno` files.

3 Write janno objects back to .janno files

janno objects usually contain list columns, that can not directly be written to a flat text file like the .janno file. The function write_janno solves that. It employs a helper function flatten_janno, which translates list columns to the string list format in .janno files (so: multiple values for one cell separated by ;). This only works for vector list columns, so when each cell contains a vector of values. If a list column cotains other data structures, e.g. data.frames, they will be dropped and replaced with the NULL value n/a in the resulting .janno file.

```
janno::write_janno(
  my_janno_object,
  path = "path/to/my/new/janno_file.janno"
)
```

4 Process age information in janno objects

.janno files contain age information in multiple different columns. The function janno::process_age() works with this age information to calculate different derived columns, which are then added to the input janno object.

You can run it with

```
janno::process_age(
  my_janno_object,
  choices = c("Date_BC_AD_Prob", "Date_BC_AD_Median_Derived", "Date_BC_AD_Sample"),
  n = 100
)
```

The choices argument contains the list of columns that should be calculated and added. n is the number of samples that should be drawn for Date_BC_AD_Sample.

4.1 Output column Date_BC_AD_Prob

Date_BC_AD_Prob is a list column with a data.frame for each janno row ("samples"). This data.frame stores a density distribution (sum_dens) over a set of years (age) with the information of a given year is within two standard deviations (two_sigma) from the median age (center).

age	sum_dens	two_sigma	center
-1506	0.00000456	FALSE	FALSE
-1505	0.00000622	FALSE	FALSE
-1504	0.00000907	FALSE	FALSE
…	…	…	…

The density distributions are either the result of (sum) calibration on radiocarbon dates or – for samples that are only contextually dated – a uniform distribution over the archaeologically determined age.

4.2 Output column Date_BC_AD_Median_Derived

Date_BC_AD_Median_Derived is a simple integer column with the median age (in years) as determined for Date_BC_AD_Prob.

4.3 Output column Date_BC_AD_Sample

Date_BC_AD_Sample is again a list column with a vector of n ages (in years) for each sample. These ages are drawn with sample(prob = ...) considering the probability distribution calculated for Date_BC_AD_Prob.

5 General helper functions

When you're preparing a `.janno` file and want to determine the entries for the columns `Date_BC_AD_Median`, `Date_BC_AD_Start` and `Date_BC_AD_Stop` from radiocarbon dates, then `janno::quickcalibrate()` might come in handy.

```
janno::quickcalibrate(ages, sds)
```

`ages` takes a list of uncalibrated ages BP and `sds` a list of standard deviations. If multiple ages are provided for one sample, then the function automatically performs a sum calibration.

`quickcalibrate(list(1000, c(2000, 2200)), list(20, c(30, 40)))` for example returns a data.frame like this:

Date_BC_AD_Median	Date_BC_AD_Start	Date_BC_AD_Stop
1029	996	1144
-88	-364	98

This output can be copied to the new `.janno` file.