Package: stemmatology (via r-universe)

October 13, 2026

Type Package

Title Stemmatological Analysis of Textual Traditions

Version 0.3.2

Date 2019-03-28

Author Jean-Baptiste Camps ; Florian Cafiero

Maintainer Jean-Baptiste Camps <jbcamps@hotmail.com>

Description Explore and analyse the genealogy of textual or musical traditions, from their variants, with various stemmatological methods, mainly the disagreement-based algorithms suggested by Camps and Cafiero (2015) <doi:10.1484/M.LECTIO-EB.5.102565>.

BugReports https://github.com/Jean-Baptiste-Camps/stemmatology/issues

Imports graphics, stats, cluster, igraph, xml2

Suggests testthat, knitr, rmarkdown, covr

License GPL-3 | file LICENSE

Encoding UTF-8

NeedsCompilation no

URL https://github.com/Jean-Baptiste-Camps/stemmatology

VignetteBuilder knitr

RoxygenNote 6.0.1

Config/pak/sysreqs libglpk-dev libxml2-dev

Repository https://jean-baptiste-camps.r-universe.dev

Date/Publication 2019-04-10 19:27:32 UTC

RemoteUrl https://github.com/jean-baptiste-camps/stemmatology

RemoteRef HEAD

RemoteSha 3ba4bc6580f439e9ab2a2cf070ca27b72a700eaa

Contents

fournival . . . 2
heinrichi . . . 3
import.TEIApparatus . . . 5
layout_as_stemma . . . 6
notreBesoin . . . 7
otinel . . . 8
parzival . . . 9
PCC . . . 11
PCC.buildGroup . . . 13
PCC.conflicts . . . 14
PCC.contam . . . 16
PCC.disagreement . . . 18
PCC.elimination . . . 20
PCC.equipollent . . . 21
PCC.Exploratory . . . 23
PCC.overconflicting . . . 25
PCC.reconstructModel . . . 27
PCC.Stemma . . . 29
stemmatology . . . 31

Index **35**

`fournival` *Fournival Data Set*

Description

Data from the tradition of Richart de Fournival, *Bestiaire d'Amours*, from C. Segre's edition, limited to archetype *y* with only substantive readings selected.

Usage

```
data(fournival)
```

Format

A matrix with 292 observations on the following 10 variables.

`A` a numeric vector

`B` a numeric vector

`C` a numeric vector

`D` a numeric vector

`E` a numeric vector

`H` a numeric vector

`I` a numeric vector

`J` a numeric vector

`K` a numeric vector

`O` a numeric vector

Details

Only the manuscripts from archetype *y* have been retained, in order to have a tradition with limited contamination, and a 10% sample has been taken in the full text. The variant locations have been selected to retain only substantial readings. The data is presented here as used in Camps & Cafiero 2015, without further modifications or corrections. Readings have been converted to numeric codes (0 being omission, and NA an absence of value).

Source

Richart de Fornival. Li Bestiaires d'Amours di maistre Richart de Fornival e li response du bestiaire. edited by Cesare Segre, Milano & Napoli, 1957.

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Examples

```
data(fournival)
```

`heinrichi` *Heinrichi data set*

Description

Data from the artificial textual tradition *Heinrichi*

Usage

```
data(heinrichi)
```

Format

A matrix with 1208 observations on the following 37 variables.

`A` a numeric vector

`Ab` a numeric vector

`Ac` a numeric vector

`Ad` a numeric vector

`Ae` a numeric vector

`B` a numeric vector

`Ba` a numeric vector

`Bb` a numeric vector

`Bd` a numeric vector

`Be` a numeric vector

`C` a numeric vector

`Ca` a numeric vector

`Cb` a numeric vector

`Cc` a numeric vector

`Cd` a numeric vector

`Ce` a numeric vector

`Cf` a numeric vector

`Da` a numeric vector

`E` a numeric vector

`F` a numeric vector

`G` a numeric vector

`H` a numeric vector

`I` a numeric vector

`J` a numeric vector

`K` a numeric vector

`L` a numeric vector

`M` a numeric vector

`N` a numeric vector

`O` a numeric vector

`P` a numeric vector

`R` a numeric vector

`S` a numeric vector

`T` a numeric vector

`V` a numeric vector

`W` a numeric vector

`X` a numeric vector

`Z` a numeric vector

Details

The data comes from an artificial tradition, created under controlled circumstances. The data is presented here as used in Camps & Cafiero 2015, without further modifications or corrections. Readings have been converted to numeric codes (0 being omission, and NA an absence of value).

Source

Roos, Teemu, and Tuomas Heikkilä. 'Evaluating methods for computer-assisted stemmatology using artificial benchmark data sets'. *Literary and Linguistic Computing* 24/4 (2009), p. 417–433.

Roos, Teemu, Tuomas Heikkilä, and Petri Myllymäki. 'Computer-Assisted Stemmatology Challenge'. Helsinki, 2007, https://www.cs.helsinki.fi/u/ttonteri/casc/data.html.

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Examples

```
data(heinrichi)
```

---

import.TEIApparatus *Import TEI apparatus*

Description

Import a TEI encoded parallel-segmentation apparatus.

Usage

```
import.TEIApparatus(file = "", appTypes = c("substantive"))
```

Arguments

file	Path to a valid TEI file.
appTypes	a vector of strings giving the types of entries that should be retained (as per app/@type). Default: "substantive". To include all types and no type, set to NULL.

Details

This function takes in input the path to a TEI file, with a <listWit>, with <witness> identified by an @xml:id, and with <app> entries encoded in parallel-segmentation mode. Using the witness sigla, it then creates a database of variant locations, with witnesses in columns and variant locations in rows. The @types attributes of the <app> elements are used to assess if they should be included in the variant locations matrix (default: only 'substantive' app entries). The readings are identified either by a code reflecting their order in the file (1 ... n) and omissions by 0. If <app> entries have @xml:id, they are used as rownames. Otherwise, the index is used.

Value

Either a numeric matrix, with witnesses in columns, variant locations in rows, and readings coded by a number; or, if alternative readings were found at some point, a character matrix, with witnesses in columns, variant locations in rows, and, in each cell, one or several readings, coded by numbers and separated by a comma (e.g. '1,2,3', if the witness has three different readings). The output of this function can be used as input for the PCC functions.

Note

If you want more control over the conversion, you can use directly the stylesheets available at https://github.com/Jean-Baptiste-Camps/stemmatology-utils.

Author(s)

Jean-Baptiste Camps (<jbcamps@hotmail.com>) & Florian Cafiero

References

Jean-Baptiste Camps, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2), edited by Andrew U. Frank et al., 2018, pp. 65–74, https://hal.archives-ouvertes.fr/hal-01695903/document.

`layout_as_stemma` *layout_as_stemma*

Description

`layout_as_stemma` creates a tree-like layout from an edgelist, where nodes are placed horizontally according to a measure of distance from their parent node.

Usage

```
layout_as_stemma(x)
```

Arguments

x	an edgelist containing, as a third column, the distance between the two nodes.

Details

The distance between the nodes will usually correspond to the number of different readings (disagreements and omissions). If a node has several parents, the function will consider only the distance from the last parent in topological order.

Value

A layout, i.e. a matrix of two columns, giving x,y coordinates for each node.

Warning

This function is experimental. Horizontal overlapping may occur has a result.

Author(s)

Jean-Baptiste Camps

See Also

PCC.Stemma, PCC.reconstructModel.

Examples

```
edgelist = structure(
    c("{ABC}", "{ABC}", "{ABC}", "D", "A","A","G",
        "A", "B", "C", "E", "F","G","H",
        1,5,3,10,3,4,5), .Dim = c(7L, 3L)
  )
g = igraph::graph_from_edgelist(edgelist[,1:2], directed = TRUE)
layout = layout_as_stemma(edgelist)
plot(g, layout = layout)
```

---

notreBesoin *Notre Besoin data set*

Description

Data from the artificial tradition *Notre-Besoin*

Usage

```
data(notreBesoin)
```

Format

A matrix with 42 observations on the following 13 variables.

`nb1` a numeric vector

`nb2` a numeric vector

`nb3` a numeric vector

`nb4` a numeric vector

`nb5` a numeric vector

`nb6` a numeric vector

`nb7` a numeric vector

`nb8` a numeric vector

`nb9` a numeric vector

`nb10` a numeric vector

`nb11` a numeric vector

`nb12` a numeric vector

`nb13` a numeric vector

Details

The data comes from an artificial tradition, created under controlled circumstances. The variant locations have been selected to retain only substantial readings. The data is presented here as used in Camps & Cafiero 2015, without further modifications or corrections. Readings have been converted to numeric codes (0 being omission, and NA an absence of value).

Source

Baret, Philippe V., P. Robinson, and C. Macé. 'Testing methods on an artificially created textual tradition'. *Linguistica computazionale* 24 (2004), p. 1000–1029.

Roos, Teemu, Tuomas Heikkilä, and Petri Myllymäki. 'Computer-Assisted Stemmatology Challenge'. Helsinki, 2007, https://www.cs.helsinki.fi/u/ttonteri/casc/data.html.

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Examples

```
data(notreBesoin)
```

---

otinel *Otinel data set*

Description

Data for the tradition of the *Chanson d'Otinel*

Usage

```
data(otinel)
```

Format

A matrix with 36 observations on the following 8 variables.

`A` a numeric vector

`B` a numeric vector

`M` a numeric vector

`NOt` a numeric vector

`WOt` a numeric vector

`EOtA` a numeric vector

`EOtF` a numeric vector

`EOtT` a numeric vector

Details

A sample of 36 substantial readings, taken from the tradition (Old French and translations) of the *Chanson d'Otinel*. Data will be subject to updates. Readings have been converted to numeric codes (0 being omission, and NA an absence of value).

Source

Camps, Jean-Baptiste. *La 'Chanson d'Otinel': édition complète du corpus manuscrit et prolégomènes à l'édition critique*, PhD thesis, dir. Dominique Boutet, Paris-Sorbonne, 2016, http://www.theses.fr/2016PA040173.

Examples

```
data(otinel)
```

`parzival` *Parzival data set*

Description

Data from the artificial textual tradition *Parzival*

Usage

```
data(parzival)
```

Format

A matrix with 139 observations on the following 16 variables.

`p1` a numeric vector

`p2` a numeric vector

`p3` a numeric vector

`p4` a numeric vector

`p5` a numeric vector

`p6` a numeric vector

`p7` a numeric vector

`p8` a numeric vector

`p9` a numeric vector

`p10` a numeric vector

`p11` a numeric vector

`p12` a numeric vector

`p13` a numeric vector

`p14` a numeric vector

`p15` a numeric vector

`p16` a numeric vector

Details

The data comes from an artificial tradition, created under controlled circumstances. The variant locations have been selected to retain only substantial readings. The data is presented here as used in Camps & Cafiero 2015, without further modifications or corrections. Readings have been converted to numeric codes (0 being omission, and NA an absence of value).

Source

M. Spencer, E. A. Davidson, A. C. Barbrook, and C. J. Howe. 'Phylogenetics of artificial manuscripts'. *Journal of Theoretical Biology*, 227 (2004), p. 503–11, https://doi.org/10.1016/j.jtbi.2003.11.022.

Roos, Teemu, Tuomas Heikkilä, and Petri Myllymäki. 'Computer-Assisted Stemmatology Challenge'. Helsinki, 2007, https://www.cs.helsinki.fi/u/ttonteri/casc/data.html.

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Examples

```
data(parzival)
```

PCC *PCC (Poole-Camps-Cafiero) stemmatological method*

Description

Global shell for all the PCC functions, both exploratory and stemma-building. This command successively executes PCC.Exploratory and PCC.Stemma, while asking user for input when necessary.

Usage

```
PCC(x, omissionsAsReadings = FALSE, alternateReadings = FALSE, limit = 0,
    recoverNAs = TRUE, layout_as_stemma = FALSE, pauseAtPlot = FALSE,
    ask = TRUE, threshold = NULL, verbose = FALSE)
```

Arguments

x	if alternateReadings = FALSE (default), a numeric matrix, with witnesses in columns, variant locations in rows, and readings coded by a number; if alternateReadings = TRUE, a character matrix, with witnesses in columns, variant locations in rows, and, in each cell, one or several readings, coded by numbers and separated by a comma (e.g. '1,2,3', if the witness has three different readings).
omissionsAsReadings	logical; if TRUE, omissions are treated as variant readings. They are taken into account in determining conflicts between variant locations or in computing severe disagreements between witnesses). Default: FALSE.
alternateReadings	logical; if TRUE, a witness can have multiple variants for a single variant location (contaminated manuscripts, editio variorum, ...), encoded as comma-separated values. Default: FALSE.
limit	The maximum number of severe disagreements expected for witnesses to be in the same group. Default: 0.
recoverNAs	logical; if TRUE, when an actual witness or reconstructed subarchetype is identified to the reconstructed model of a group, every NA it has is recovered by taking the value of the reconstructed model; if FALSE, their NAs values are kept. Default: TRUE.
layout_as_stemma	logical; if TRUE, the witnesses will be placed vertically according to the distance from their parent, as per the function layout_as_stemma (experimental!) Default: FALSE
pauseAtPlot	logical; if TRUE, the algorithms stops at each plot during the execution of PCC.contam. Default: FALSE.
ask	logical; if FALSE, decisions will be made without asking the user for input. Default: TRUE
threshold	numeric; the centrality threshold above which variant locations are considered to be over-conflicting. Used only with ask = FALSE.
verbose	logical; if FALSE, the function will only return the results, without information on the operations. Default: FALSE.

Details

This function provides a single entry to all the algorithms used in the PCC method. It successively calls `PCC.Exploratory` and `PCC.Stemma`. The algorithmic principles of the PCC method are described in Camps & Cafiero 2015. It builds on the propositions of Poole 1974, 1979.

In a first stage, problematic configurations in the traditions (i.e. configurations that cannot be linked to a normal genealogy, without contamination or polygenesis) are identified by crossing every possible pair of variant locations, and are then plotted as a network. When the most unreliable variant locations (i.e. unreliable) are identified, different methods for eliminating them are offered.

In a second time, a stemma is iteratively built, using the variant locations selected in the first stage. At each step, witness with no severe disagreements (i.e. disagreements between two witnesses, on two readings both shared with at least one other witness, cf. *Trennfehler*, *errores separativi*) are grouped together. A model is then reconstructed for each group, and either identified to a witness of the group or to an hypothetical subarchetype.

The option `recoverNAs=TRUE` is a novelty not described in the original paper (Camps & Cafiero 2015).

For more information about the underlying principles behind the method applied here, particularly the distinction between severe and benign disagreement, the different status given to readings, omissions and *lacunae*, the notion of conflict between variant locations or the way the stemma is built, see the *references* section.

Value

The function returns either a single object of class `"pccStemma"`, or a list containing several objects of class `"pccStemma"` (if multiple stemmata were drawn); see `PCC.Stemma`.

Author(s)

Jean-Baptiste Camps (<jbcamps@hotmail.com>) & Florian Cafiero(<florian.cafiero@polytechnique.edu>)

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Poole, Eric. 'L'analyse stemmatique des textes documentaires'. *La pratique des ordinateurs dans la critique des textes*, Paris, 1979, p. 151-161.

Poole, Eric, 'The Computer in Determining Stemmatic Relationships'. *Computers and the Humanities*, 8-4 (1974), p. 207-16.

See Also

`PCC.Exploratory`, `PCC.Stemma`.

Examples

```
# Load data
data("fournival")
# or alternatively, import it
# fournival = import.TEIApparatus(file = "myFournival.xml",
#    appTypes = c("substantive"))

# Analyse it with the PCC functions

# Non interactive mode
PCC(fournival, ask = FALSE, threshold = 0.06)

## Not run:
# Interactive mode
PCC(fournival)

## End(Not run)
```

`PCC.buildGroup` *PCC.buildGroup: Group Witnesses in Clusters*

Description

`PCC.buildGroup` groups together witnesses in relevant clusters, based on the absence (or number inferior to a limit) of severe disagreements between them.

Usage

```
PCC.buildGroup(x, limit = 0, ask = TRUE)
```

Arguments

`x`	A PCC.disagreement object.
`limit`	The maximum number of severe disagreements allowed for two witnesses in the same group. Default (and advised) value: `0`.
`ask`	logical; if FALSE, decisions will be made without asking the user for input. Default: TRUE

Details

Witnesses a number of severe disagreements between them lesser than or equal to `limit` are grouped together. This disagreement-based method is described in Camps & Cafiero 2015.

Value

The function returns a list containing:

`database`	The original database.
`groups`	A list of the groups that were created, identified by their labels.

Author(s)

Jean-Baptiste Camps (<jbcamps@hotmail.com>) & Florian Cafiero

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Poole, Eric. 'L'analyse stemmatique des textes documentaires'. *La pratique des ordinateurs dans la critique des textes*, Paris, 1979, p. 151-161.

Poole, Eric, 'The Computer in Determining Stemmatic Relationships'. *Computers and the Humanities*, 8-4 (1974), p. 207-16.

See Also

PCC.Stemma, PCC.disagreement, PCC.reconstructModel.

Examples

```
# A fictional simple tradition
x = matrix(
    c(
      1,0,1,1,1,1,1,1,
      1,0,1,2,2,2,1,2,
      1,0,0,3,2,1,NA,3,
      2,0,1,4,NA,1,1,1,
      2,1,2,5,2,1,1,4
    ), nrow = 8, ncol = 5,
    dimnames = list(c("VL1","VL2","VL3","VL4","VL5","VL6","VL7","VL8"),
                    c("A","B","C","D","E")))
# Compute disagreement(s)
x = PCC.disagreement(x)
# And now build the groups
PCC.buildGroup(x)
```

PCC.conflicts *PCC Exploratory Methods: Conflicts between Variant Locations*

Description

Given a matrix of variant locations, this function compares them by pairs to identify conflicting genealogical information between them.

Usage

```
PCC.conflicts(x, omissionsAsReadings = FALSE, alternateReadings = FALSE)
```

Arguments

x	if alternateReadings = FALSE (default), a numeric matrix, with witnesses in columns, variant locations in rows, and readings coded by a number; if alternateReadings = TRUE, a character matrix, with witnesses in columns, variant locations in rows, and, in each cell, one or several readings, coded by numbers and separated by a comma (e.g. '1,2,3', if the witness has three different readings).
omissionsAsReadings	logical; if TRUE, omissions are treated as variant readings (and taken into account in determining contradictions between variant locations or in computing disagreements between witnesses). Default: FALSE.
alternateReadings	logical; if TRUE, a witness can have multiple variants for a single variant location (contaminated manuscripts, editio variorum, ...), encoded as comma-separated values. Default: FALSE.

Details

This function tries to identify conflicts between variant locations, understood as contradictions in the genealogical information they might contain. In order to do that, every possible pair of variant locations is analysed in order to see if both can denote at least one possible normal genealogy (i.e. a genealogy without contamination or polygenesis). If not, they are considered "conflicting".

A network representing all the conflicts between variant locations is drawn, and the total number of conflicts and centrality index by variant location is given, as an help to estimate which variant locations are unreliable. This output can be then passed to the function PCC.overconflicting. See Camps & Cafiero 2015 for more details.

Value

An object of class "pccConflicts", a list containing

edgelist	a two-column character matrix, giving the edges between variant locations in the network of conflicts (adjacency list)
conflictsTotal	a one-column numeric matrix, giving the total number of conflicts per variant location
database	the original database used for the calculations

Author(s)

Jean-Baptiste Camps (<jbcamps@hotmail.com>) & Florian Cafiero

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Poole, Eric. 'L'analyse stemmatique des textes documentaires'. *La pratique des ordinateurs dans la critique des textes*, Paris, 1979, p. 151-161.

Poole, Eric, 'The Computer in Determining Stemmatic Relationships'. *Computers and the Humanities*, 8-4 (1974), p. 207-16.

See Also

PCC.Exploratory, PCC.overconflicting.

Examples

```
# Load data
data(fournival)

# Analyse its conflicts
myConflicts = PCC.conflicts(fournival)
```

PCC.contam *PCC Exploratory Methods: Contamination Detection*

Description

Detect possible contaminations by assessing the role of each witness in conflicting information between variant locations.

Usage

```
PCC.contam(x, omissionsAsReadings = FALSE, alternateReadings = FALSE, pauseAtPlot = FALSE)
```

Arguments

x	if alternateReadings = FALSE (default), a numeric matrix, with witnesses in columns, variant locations in rows, and readings coded by a number; if alternateReadings = TRUE, a character matrix, with witnesses in columns, variant locations in rows, and, in each cell, one or several readings, coded by numbers and separated by a comma (e.g. '1,2,3', if the witness has three different readings); or an object of class "pccConflicts" or "pccElimination".

`omissionsAsReadings`
: logical; if `TRUE`, omissions are treated as variant readings (and taken into account in determining conflicts between variant locations or in computing severe disagreements between witnesses). Default: `FALSE`.

`alternateReadings`
: logical; if `TRUE`, a witness can have multiple variants for a single variant location (contaminated manuscripts, editio variorum, ...), encoded as comma-separated values. Default: `FALSE`.

`pauseAtPlot`
: logical; if `TRUE`, the algorithms stops at each plot during the execution of `PCC.contam` (by setting graphical parameter `ask = TRUE`). Default: `FALSE`.

Details

To help assess the role of each witness in the conflicting information between variant locations, this function computes the number of conflicting variant locations when removing one of the witnesses, for each witness. If removing a witness makes the number of conflicting variant locations significantly drop, then contamination can be considered as plausible. Be aware that this function will be most efficient for contaminations limited to a single manuscript.

Value

An object of class "pccContam", a list containing

`totalByMs`
: a numeric matrix, with, in rows, each variant locations, and, in columns, the number of conflicts and centrality in the full database, followed by the difference in total conflicts and centrality caused by the removal of each witness.

`conflictsDifferences`
: a one row numeric matrix, containing, for each witness, the total decrease in conflicts caused by its removal from the computations

`database`
: the original database used for the calculations

Warning

The execution of this command can be time-consuming for large databases.

Note

Additional contamination detection methods will be implemented in the future.

Author(s)

Jean-Baptiste Camps & Florian Cafiero

References

Camps, Jean-Baptiste. 'Detecting Contaminations in Textual Traditions Computer Assisted and Traditional Methods'. Leeds, International Medieval Congress, 2013, unpublished paper, https://www.academia.edu/3825633/Detecting_Contaminations_in_Textual_Traditions_Computer_Assisted_and_Traditional_Methods.

See Also

PCC.Exploratory, PCC.equipollent.

Examples

```
# load a data set
data("fournival")

# identify conflicts on a subset
x = PCC.conflicts(fournival)
# identify problematic variant locations
x = PCC.overconflicting(x, ask = FALSE, threshold = 0.06)
# eliminate them
x = PCC.elimination(x)
# examinate the rest of the problematic cases, to detect
# plausible contaminations
PCC.contam(x)
```

PCC.disagreement *PCC.disagreement: Find disagreements and agreements between witnesses*

Description

The PCC.disagreement function helps spotting disagreements (and agreements) between manuscripts. For a given numeric matrix, representing the variants in different manuscripts, it locates disagreements (benign or severe), agreements and omissions in common between manuscripts.

Usage

```
PCC.disagreement(x, omissionsAsReadings = FALSE)
```

Arguments

x	a numeric matrix, with manuscripts in columns, variants in rows, and readings coded by a number.
omissionsAsReadings	logical; if TRUE, omissions are considered as readings.

Details

A distinction is made between severe and benign disagreements (see Camps & Cafiero 2015). Severe disagreements are disagreements between witnesses on two readings that are each shared by at least two witnesses. They have stronger genealogical implications than benign disagreements, that involve at least one singular reading. This distinction is used by the methods of the PCC family.

This function also gives common omissions, and oriented omissions (i.e. omission present in one witness but not an other). No distinction is made between omission and addition, as this means establishing the orientation in genealogical relationship.

Agreements are given as well, mostly with an indicative value, as they cannot be taken as a direct measure of similarity.

Value

The function returns:

`database` The original database.

`severeDisagreement`
A list of the severe disagreements between manuscripts.

`benignDisagreement`
A list of the benign disagreements between manuscripts.

`agreements` A list of agreements between manuscripts.

`omissionsInCommon`
A list of all the omissions in common between manuscripts(if omissionsAsReadings is set to TRUE, this will be NA).

`omissionsOriented`
A list of the omissions present in a manuscript but not in another (if omissionsAsReadings is set to TRUE, this will be NA).

Author(s)

Jean-Baptiste Camps & Florian Cafiero

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Poole, Eric. 'L'analyse stemmatique des textes documentaires'. *La pratique des ordinateurs dans la critique des textes*, Paris, 1979, p. 151-161.

Poole, Eric, 'The Computer in Determining Stemmatic Relationships'. *Computers and the Humanities*, 8-4 (1974), p. 207-16.

See Also

`PCC.Stemma`, `PCC.buildGroup`, `PCC.reconstructModel`.

Examples

```
#Load a tradition
data("fournival")
#Option: explore the tradition to see problems in variant locations
#PCC.Exploratory(fournival)

#Calculate disagreements
PCC.disagreement(fournival)
```

`PCC.elimination` *PCC Exploratory Methods: Elimination of Over-Conflicting Variant Locations*

Description

This function removes from the database the variant locations labelled as over-conflicting by the `PCC.overconflicting` function.

Usage

```
PCC.elimination(x)
```

Arguments

`x`	an object of class pccOverconflicting.

Details

When PCC.overconflicting has been applied to a PCC.conflicts object, it returns a table where over-conflicting variants are labeled as such. The PCC.elimination function simply removes those variants.

Value

A numeric matrix, with witnesses in columns, variant locations in rows, and readings coded by a number, from which over-conflicting variant locations have been removed.

Note

The notion of using a centrality threshold for the identification of over-conflicting variant locations is found in Camps & Cafiero 2015. Other formulas for this centrality might be implemented in the future.

Author(s)

Jean-Baptiste Camps (<jbcamps@hotmail.com>) & Florian Cafiero

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

See Also

PCC.Exploratory, PCC.conflicts, PCC.overconflicting, PCC.contam.

Examples

```
# Load data
data("fournival")
# Analyse its conflicts
myConflicts = PCC.conflicts(fournival)
## Not run:
# Interactive mode: identify over-conflicting VL
PCC.overconflicting(myConflicts)

## End(Not run)
# Non interactive mode
myConflicts = PCC.overconflicting(myConflicts, ask = FALSE, threshold = 0.06)
# Create a new DB without problematic VL
myNewDB = PCC.elimination(myConflicts)
```

PCC.equipollent *PCC Exploratory Methods: Extracting Competing Genealogies*

Description

A single table of variant locations can sometimes reflect different competing genealogies, due to contamination, either for a single manuscript, or for the whole tradition. PCC.equipollent identifies the variant locations without internal conflicts, and allows to create separate databases for each internally consistent configuration.

Usage

```
PCC.equipollent(x, ask = TRUE, scope = NULL, wits = NULL, verbose = FALSE)
```

Arguments

x	A PCC.conflicts object.
ask	logical; if FALSE, decisions will be made without asking the user for input. Default: TRUE. With FALSE, it is mandatory to specify scope and wits.
scope	should the inconsistent variant locations be neutralised for the whole tradition ("T") or only some witnesses ("W")? Use only with ask = FALSE.
wits	a vector containing the names of the witnesses for which inconsistent variant locations should be neutralised. Use only with ask = FALSE and scope = 'W'.
verbose	logical; if FALSE, the function will only return the results, without information on the operations. Default: FALSE.

Details

Some over-conflicting variant locations can be algorithmically ruled out for the building of a stemma (see PCC.conflicts, PCC.overconflicting and PCC.elimination). Yet, in some cases, choosing between conflicting variables is algorithmically undecidable. This might be due sometimes to contamination (see PCC.contam). PCC.equipollent helps addressing such cases. It tries to assess, first, the sets of variant locations that are internally consistent (no conflict among themselves), and then, creates as many different databases as sets were found. In creating these new databases, the variant location that have conflicting information with the current set are either fully neutralised (scope = "T") or neutralised only for some witnesses (scope = "W").

Value

An object of class pccEquipollent, a list containing

databases	a list with all alternative databases that have been created, if any
notInConflict	a list with the set(s) of VL without internal conflicts

Warning

This function is still experimental, and will work optimally only for simple cases, where competing genealogies can be easily identified.

Author(s)

Jean-Baptiste Camps & Florian Cafiero

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste. 'Detecting Contaminations in Textual Traditions Computer Assisted and Traditional Methods'. Leeds, International Medieval Congress, 2013, unpublished paper, https://www.academia.edu/3825633/Detecting_Contaminations_in_Textual_Traditions_Computer_Assisted_and_Traditional_Methods.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

See Also

PCC.Exploratory, PCC.conflicts, PCC.overconflicting, PCC.elimination, PCC.contam.

Examples

```
# load data
data("fournival")

# look for conflicts
y = PCC.conflicts(fournival)
# identify and eliminate overconflicting VL
y = PCC.overconflicting(y, ask = FALSE, threshold = 0.06)
y = PCC.elimination(y)
# look for further conflicts
y = PCC.conflicts(y)


# and now, create configurations for competing genealogies
# for instance, for one witness
newDB = PCC.equipollent(y, ask = FALSE, scope = "W", wits = "D")

# Alternatively, you can create them for the whole tradition
newDB = PCC.equipollent(y, ask = FALSE, scope = "T")
# or for several witnesses
newDB = PCC.equipollent(y, ask = FALSE, scope = "W", wits = c("A", "D"))

# and then you proceed to create one or several stemmata, e.g.
# PCC.Stemma(newDB$databases[[1]], ask = FALSE)
```

PCC.Exploratory *PCC Exploratory methods*

Description

This is the global function for exploratory methods of the PCC family. It interactively makes use of the lower-level exploratory functions, to assess conflicts between variant locations, eliminate problematic configurations or identify likely contaminations.

Usage

```
PCC.Exploratory(x, omissionsAsReadings = FALSE, alternateReadings = FALSE,
    pauseAtPlot = FALSE, ask = TRUE, threshold = NULL, verbose = FALSE)
```

Arguments

x	if alternateReadings = FALSE (default), a numeric matrix, with witnesses in columns, variant locations in rows, and readings coded by a number; if alternateReadings = TRUE, a character matrix, with witnesses in columns, variant locations in rows, and, in each cell, one or several readings, coded by numbers and separated by a comma (e.g. '1,2,3', if the witness has three different readings).
omissionsAsReadings	logical; if TRUE, omissions are treated as variant readings (and taken into account in determining conflicts between variant locations or in computing severe disagreements between witnesses). Default: FALSE.
alternateReadings	logical; if TRUE, a witness can have multiple variants for a single variant location (contaminated manuscripts, editio variorum, ...), encoded as comma-separated values. Default: FALSE.
pauseAtPlot	logical; if TRUE, the algorithms stops at each plot during the execution of PCC.contam (by setting graphical parameter ask = TRUE). Default: FALSE.
ask	logical; if FALSE, decisions will be made without asking the user for input. Default: TRUE
threshold	numeric; the centrality threshold above which variant locations are considered to be over-conflicting. Used only with ask = FALSE.
verbose	logical; if FALSE, the function will only return the results, without information on the operations. Default: FALSE.

Details

This function is meant to guide the user through the process of assessing and eliminating unreliable variant locations and/or identify competing genealogies (i.e. contamination), as described in Camps & Cafiero 2015.

It starts by computing and plotting the network of conflicting variant locations (i.e. variant locations that present contradictory genealogical information), by calling PCC.conflicts, and then interactively aids the user in determining overconflicting variant locations (with PCC.overconflicting), eliminating problematic variant locations (PCC.elimination), detecting contamination (PCC.contam) or creating new databases reflecting competing genealogies (PCC.equipollent).

Value

According to the choices made by the user, the output can be an object belonging to one of the following classes: pccConflicts, pccOverconflicting, pccContam or pccEquipollent.

Author(s)

Jean-Baptiste Camps (<jbcamps@hotmail.com>) & Florian Cafiero

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based*

Research in the Humanities (CRH-2), edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Poole, Eric. 'L'analyse stemmatique des textes documentaires'. *La pratique des ordinateurs dans la critique des textes*, Paris, 1979, p. 151-161.

Poole, Eric, 'The Computer in Determining Stemmatic Relationships'. *Computers and the Humanities*, 8-4 (1974), p. 207-16.

See Also

PCC.conflicts, PCC.overconflicting, PCC.elimination, PCC.contam, PCC.equipollent.

Examples

```
# Load data
data(fournival)
## Not run:
# Interactive mode
# or alternatively, import it
#fournival = import.TEIApparatus(file = "myFournival.xml",
#                      appTypes = c("substantive"))
# Analyse it with the PCC functions
PCC.Exploratory(fournival)

## End(Not run)

# Non interactive mode
PCC.Exploratory(fournival, ask = FALSE, threshold = 0.06)
```

PCC.overconflicting *PCC Exploratory Methods: Identification of Over-Conflicting Variant Locations*

Description

Given a network of conflicts (contradictions) between variant locations, this function helps in assessing which are the problematic ones.

Usage

```
PCC.overconflicting(x, ask = TRUE, threshold = NULL)
```

Arguments

x	an object of class pccConflicts.
ask	logical; if FALSE, decisions will be made without asking the user for input. Default: TRUE
threshold	numeric; the centrality threshold above which variant locations are considered to be over-conflicting. Used only with ask = FALSE.

Details

This function is dedicated to the identification of problematic variant locations, as defined in Poole 1974 and Camps & Cafiero 2015. It helps the user defining a threshold, defined in terms of centrality index, above which variant locations are considered to be over-conflicting. This output can be then passed to the function PCC.elimination, to remove them from the database.

Value

An object of class "pccOverconflicting", a list containing the three same first objects as the "pccConflicts" input,

edgelist	a two-column character matrix, giving the edges between variant locations in the network of conflicts
conflictsTotal	a one-column numeric matrix, giving the total number of conflicts per variant location
database	the original database used for the calculations

and adding

vertexAttributes
: a two column character matrix, with a row per vertex of the network (i.e. variant location), giving its label and colour

Author(s)

Jean-Baptiste Camps (<jbcamps@hotmail.com>) & Florian Cafiero

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Poole, Eric. 'L'analyse stemmatique des textes documentaires'. *La pratique des ordinateurs dans la critique des textes*, Paris, 1979, p. 151-161.

Poole, Eric, 'The Computer in Determining Stemmatic Relationships'. *Computers and the Humanities*, 8-4 (1974), p. 207-16.

See Also

PCC.Exploratory, PCC.conflicts, PCC.elimination.

Examples

```
# Load data
data("fournival")

# Analyse its conflicts
myConflicts = PCC.conflicts(fournival)
## Not run:
# Interactive mode: identify over-conflicting VL
PCC.overconflicting(myConflicts)

## End(Not run)
# Non interactive mode
PCC.overconflicting(myConflicts, ask = FALSE, threshold = 0.06)
```

PCC.reconstructModel *PCC.reconstructModel: Reconstruct the Model of Groups of Witnesses*

Description

PCC.reconstructModel examines coherent clusters of witnesses (PCC.buildGroup), to either identify their model in the tradition, either suggest a reconstructed model for the group.

Usage

```
PCC.reconstructModel(x, omissionsAsReadings = FALSE, recoverNAs = TRUE,
                     ask = TRUE, verbose = FALSE)
```

Arguments

x	The output of PCC.buildGroup.
omissionsAsReadings	logical; if TRUE, omissions are treated as variant readings. They are taken into account in determining conflicts between variant locations or in computing severe disagreements between witnesses). Default: FALSE.
recoverNAs	logical; if TRUE, when an actual witness or reconstructed subarchetype is identified to the reconstructed model of a group, every NA it has is recovered by taking the value of the reconstructed model; if FALSE, their NAs values are kept. Default: TRUE.
ask	logical; if FALSE, decisions will be made without asking the user for input. Default: TRUE
verbose	logical; if FALSE, the function will only return the results, without information on the operations. Default: FALSE

Details

This function takes PCC.buildGroup objects as input. It assesses the characteristics of the model of each group, and compares it to the existing witnesses. If a witness has the same characteristics as the computed model, it is identified as the model for the group. If no witness seems to be a good fit, the function adds a reconstructed model to the tradition.

Value

The function returns a list containing

fullDatabase	The full database, with the new reconstructed models and recovered NAs (if applicable).
database	The same with the *descripti* removed.
edgelist	An edgelist expressing the relations between the witnesses of each group with, as a third column, the distances between witnesses.
models	A list containing the database of readings for each model at the time of their reconstruction (i.e., before they are compared to extant witnesses).
modelsByGroup	A matrix with, in columns the groups, and a single row containing the label of their model.

Author(s)

Jean-Baptiste Camps (<jbcamps@hotmail.com>) & Florian Cafiero

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Poole, Eric. 'L'analyse stemmatique des textes documentaires'. *La pratique des ordinateurs dans la critique des textes*, Paris, 1979, p. 151-161.

Poole, Eric, 'The Computer in Determining Stemmatic Relationships'. *Computers and the Humanities*, 8-4 (1974), p. 207-16.

See Also

PCC.Stemma, PCC.disagreement, PCC.buildGroup.

Examples

```
#A fictional simple tradition
x = list(database = matrix(
    c(
      1,0,1,1,1,1,1,1,
      1,0,1,2,2,2,1,2,
      1,0,0,3,2,1,NA,3,
      2,0,1,4,NA,1,1,1,
      2,1,2,5,2,1,1,4
    ), nrow = 8, ncol = 5,
    dimnames = list(c("VL1","VL2","VL3","VL4","VL5","VL6","VL7","VL8"),
                    c("A","B","C","D","E"))),
    groups = list(c("A", "B", "C"), c("D", "E")))
#And now, reconstruct the groups
PCC.reconstructModel(x)
```

PCC.Stemma *PCC.Stemma: Building the Stemma Codicum*

Description

Builds a *stemma codicum* of the tradition, following the Poole-Camps-Cafiero method.

Usage

```
PCC.Stemma(x, omissionsAsReadings = FALSE, limit = 0, recoverNAs= TRUE,
           layout_as_stemma = FALSE, ask = TRUE, verbose = FALSE)
```

Arguments

x	a numeric matrix, with witnesses in columns, variant locations in rows, and readings coded by a number; or a character matrix, with witnesses in columns, variant locations in rows, and, in each cell, one or several readings, coded by numbers and separated by a comma (e.g. '1,2,3', if the witness has three different readings).
omissionsAsReadings	logical; if TRUE, omissions are treated as variant readings. They are taken into account in determining conflicts between variant locations or in computing severe disagreements between witnesses). Default: FALSE.
limit	The maximum number of severe disagreements expected for witnesses to be in the same group. Default: 0.
recoverNAs	logical; if TRUE, when an actual witness or reconstructed subarchetype is identified to the reconstructed model of a group, every NA it has is recovered by taking the value of the reconstructed model; if FALSE, their NAs values are kept. Default: TRUE.

`layout_as_stemma`
: logical; if TRUE, the witnesses will be placed vertically according to the distance from their parent, as per the function `layout_as_stemma` (experimental!) Default: FALSE

`ask`
: logical; if `FALSE`, decisions will be made without asking the user for input. Default: `TRUE`

`verbose`
: logical; if `FALSE`, the function will only return the results, without information on the operations. Default: `FALSE`.

Details

The `PCC.Stemma` function calls successively the functions `PCC.disagreement`, `PCC.buildGroup` and `PCC.reconstructModel` to build a stemma codicum of the tradition studied. By default, it stops when less than four witnesses are to be compared, as the possibility of errors becomes high. The user is however able to ask the algorithm its final answer for those last witnesses.

Value

The function returns either a single list, or a list containing several lists (if multiple stemmata were drawn). Each list contains:

`fullDatabase`
: The full database, with the new reconstructed models and recovered NAs (if applicable).

`database`
: The same with the *descripti* removed.

`edgelist`
: An edgelist expressing the relations between the witnesses of each group with, as a third column, the distances between witnesses.

`models`
: A list containing the database of readings for each model at the time of their reconstruction (i.e., before they are compared to extant witnesses).

`modelsByGroup`
: A matrix with, in columns the groups, and a single row containing the label of their model.

Author(s)

Jean-Baptiste Camps (<jbcamps@hotmail.com>) & Florian Cafiero (<florian.cafiero@polytechnique.edu>)

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

Poole, Eric. 'L'analyse stemmatique des textes documentaires'. *La pratique des ordinateurs dans la critique des textes*, Paris, 1979, p. 151-161.

Poole, Eric, 'The Computer in Determining Stemmatic Relationships'. *Computers and the Humanities*, 8-4 (1974), p. 207-16.

See Also

PCC.disagreement, PCC.buildGroup, PCC.reconstructModel, layout_as_stemma.

Examples

```
# Load data
data("parzival")
# or alternatively, import it
#fournival = import.TEIApparatus(file = "myFournival.xml",
#                      appTypes = c("substantive"))
## Not run:
# Interactive mode
# Analyse it with the PCC functions
myDBs = PCC.Exploratory(parzival)
# draw a stemma
PCC.Stemma(myDBs$databases[[2]])

## End(Not run)

# Non interactive mode
myDBs = PCC.Exploratory(parzival, ask = FALSE, threshold = 0.06)
PCC.Stemma(myDBs$databases[[2]], ask = FALSE)
```

---

stemmatology *An R Stemmatology Package*

Description

Build and analyse the genealogy of textual or musical traditions.

Details

Package:	stemmatology
Type:	Package
Version:	0.3.0
Date:	2018-05-20
License:	GPL-3

Stemmatology is the name of the field dedicated to studying text genealogies and establishing genealogical tree-like graphs known as stemma codicum.

This package includes various functions for stemmatological analysis.

Input:

Most of the functions take, as input a *numeric matrix*, with witnesses in columns, variant locations in rows, and readings coded by a number, e.g.

	A	B	C	D	E	H	I	J	K	O
1	0	1	1	1	NA	1	1	NA	1	1
2	1	1	1	1	NA	1	1	NA	1	1
3	1	1	1	1	NA	1	1	NA	1	1
4	1	1	1	2	NA	1	1	NA	1	1
5	1	1	1	2	NA	1	1	NA	1	1
6	1	1	1	1	NA	1	1	NA	1	1

where A, B, . . . , O are the various witnesses in columns, 1. . . 6 the various variant locations, in rows, and the different readings are coded either 0 (omission), 1, 2, . . . , *n*. `NA` is used for the lack of information (physical lacuna, absence of observation, variant location not applicable to a given witness, etc.).

Alternatively , if `alternateReadings = TRUE`, the input can be a *character matrix*, with witnesses in columns, variant locations in rows, and, in each cell, one or several readings, coded by numbers and separated by a comma (e.g. '1,2,3', if the witness has three different readings), e.g.

	A	D	F	T	P
1	"1"	"2"	"2"	"2"	"1,2"
2	"1"	"2"	"1,2"	"2"	"1"
3	"1"	"1"	"1"	"1"	"2"
4	"1,3"	"1,2"	"1"	"2"	"3"

Notice how a witness can bear several readings (e.g., P at VL 1).

Create or import data: Data can be created inside R or imported. They can be imported by reading a csv file, for instance (e.g. with `read.csv`). They can also be imported from a TEI encoded apparatus in parallel-segmentation, either by using an XSL stylesheet, or the built-in function `import.TEIApparatus`.

The function `import.TEIApparatus` allows to import a TEI P5 encoded apparatus into a stemmatological matrix usable with other functions. It has some parameters to refine the import (variant types, . . .), and can read either from disk or from an URL.

PCC Method:

Functions are made available for the PCC method (See Camps and Cafiero 2014 or `PCC` for more details). The most important are

`PCC`: global shell for the PCC functions

`PCC.Exploratory`: global function for exploratory methods of the PCC family

`PCC.Stemma`: Building the Stemma Codicum.

Other functions: The package contains also various other functions, particularly aimed at detecting contamination. It contains for instance the function `PCC.contam`.

The package aims at making available various other stemmatological methods, including further functions for contamination detection, or for theoretical stemmatology.

Note

Please report issues with this package to https://github.com/Jean-Baptiste-Camps/stemmatology.

Author(s)

Jean-Baptiste Camps (École nationale des chartes | Université PSL).

Florian Cafiero.

Maintainer: Jean-Baptiste Camps <<jbcamps@hotmail.com>>.

References

Camps, Jean-Baptiste, and Florian Cafiero. 'Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions'. *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities (CRH-2)*, edited by Andrew U. Frank et al., 2018, pp. 65–74, https://halshs.archives-ouvertes.fr/hal-01695903v1.

Camps, Jean-Baptiste, and Florian Cafiero. 'Genealogical Variant Locations and Simplified Stemma: A Test Case'. *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, Brepols, 2015, pp. 69–93, https://halshs.archives-ouvertes.fr/halshs-01435633, DOI: 10.1484/M.LECTIO-EB.5.102565.

See Also

PCC, PCC.Exploratory, PCC.Stemma

Examples

```
# Load data
data(fournival)
# or alternatively, import it
#fournival = import.TEIApparatus(file = "myFournival.xml",
#    appTypes = c("substantive"))

## Not run:
# Interactive mode

# Analyse it with the PCC functions
PCC(fournival)

## End(Not run)

# Complete step-by-step non interactive use
data("fournival")

# look for conflicts
myConflicts = PCC.conflicts(fournival)
# remove conflicting VL
myConflicts = PCC.overconflicting(myConflicts, ask = FALSE, threshold = 0.06)
myNewData = PCC.elimination(myConflicts)
# look for competing genealogies
myConflicts = PCC.conflicts(myNewData)
```

```
myNewData = PCC.equipollent(myConflicts, ask = FALSE, scope = "W", wits = "D")
# build a stemma
PCC.Stemma(myNewData$databases[[1]], ask = FALSE)
```

Index

∗ **datasets**
  fournival, 2
  heinrichi, 3
  notreBesoin, 7
  otinel, 8
  parzival, 9
∗ **graphs**
  layout_as_stemma, 6
∗ **manip**
  import.TEIApparatus, 5
  PCC.equipollent, 21
∗ **stemmatology**
  import.TEIApparatus, 5
  layout_as_stemma, 6
  PCC, 11
  PCC.buildGroup, 13
  PCC.conflicts, 14
  PCC.contam, 16
  PCC.disagreement, 18
  PCC.elimination, 20
  PCC.equipollent, 21
  PCC.Exploratory, 23
  PCC.overconflicting, 25
  PCC.Stemma, 29
  stemmatology, 31

fournival, 2

heinrichi, 3

import.TEIApparatus, 5, *32*

layout_as_stemma, 6, *11*, *30, 31*

notreBesoin, 7

otinel, 8

parzival, 9
PCC, *6*, 11, *18*, *32, 33*
PCC.buildGroup, 13, *19*, *27, 28*, *30, 31*
PCC.conflicts, 14, *21–25*, *27*
PCC.contam, 16, *21–25*, *32*
PCC.disagreement, *14*, 18, *28*, *30, 31*
PCC.elimination, 20, *22–27*
PCC.equipollent, *18*, 21, *24, 25*
PCC.Exploratory, *12*, *16*, *18*, *21*, 23, *23*, *27*, *32, 33*
PCC.overconflicting, *15, 16*, *21–24*, 25, *25*
PCC.reconstructModel, *7*, *14*, *19*, 27, *30, 31*
PCC.Stemma, *7*, *12*, *14*, *19*, *28*, 29, *32, 33*

read.csv, *32*

stemmatology, 31
stemmatology-package *(*stemmatology*)*, 31