Package ‘kstIO’

January 26, 2023

Version 0.4-0

Date 2023-01-26

Title Knowledge Space Theory Input/Output

Description Knowledge space theory by Doignon and Falmagne (1999) <doi:10.1007/978-3-642-58625-5> is a set- and order-theoretical framework which proposes mathematical formalisms to operationalize knowledge structures in a particular domain. The 'kstIO' package provides basic functionalities to read and write KST data from/to files to be used together with the 'kst', 'kstMatrix', 'pks' or 'DAKS' packages.

License GPL (>= 3)

Depends R (>= 3.5.0), pks (>= 0.4-0), MASS, stringr, sets, relations

Imports kstMatrix

Suggests kst (>= 0.5-1)

Author Cord Hockemeyer [aut, cre]

Maintainer Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

NeedsCompilation no

Repository CRAN

Encoding UTF-8

Date/Publication 2023-01-26 16:20:02 UTC

R topics documented:

`kstIO-package` `kstIO` *File formats*

Description

Knowledge space theory by Doignon and Falmagne (1985, 1999) is a set- and order-theoretical framework, which proposes mathematical formalisms to operationalize knowledge structures in a particular domain. The 'kstIO' package provides basic functionalities to read and write KST data from/to files.

Details

This page focuses on the different file formats that can be used with the `kstIO` functions.

File Formats

Over time and in different research groups with knowledge space theory, different file formats have evolved.

Matrix Format: The probably simplest and most direct approach is to store the information in a binary ASCII matrix where a "1" in row i and column j means that item j is element of state/response pattern i.

There is no separating character between the columns, and there should be no trailing whitespace at the end of the line. The last line of the matrix must carry an EndOfLine - in most editors (except vi) this means an empty line after the matrix.

KST Tools Format: This format (Hockemeyer, 2001) extends the matrix format by two preceding header lines containing the number of items and the number of states/response patterns, respectively.

SRBT Tools Format: This format (Poetzi & Wesiak, 2001) extends the KST tools format by yet another preceding header line with format and content metadata. This new header line has the format

```
#SRBT v2.0 <struct> ASCII <comment>
```

where `<struct>` specifies the type of data stored in the file and `<comment>` is an optional arbitrary comment.

Furthermore, SRBT files may contain an arbitrary number of comment lines after the number of states/patterns.

The following data types are supported by the respective `kstIO` functions:

- basis
- data
- relation
- space
- structure

For kbase and surmise relation files, the encoding information "ASCII" is missing because these files are always in ASCII format.

CSV Format: CSV (comma separated values) is a standard file format for data tables. Within a row, the different columns are separated by commas. Please note that in some European countries and for some programs, rows are instead separated by semicolon, and the comma replaces the decimal point.

CSV files written/read by kstIO functions start with a head row containing the item IDs. The subsequent rows build the matrix as described n the previous format descriptions.

Special File/Data Types

Base Files: Base files are not available in KST tools format.

Their matrix part differs from the other files in that it contains "0", "1", and "2". A "1" means that the state is minimal for the item and a "2" means that it is not (but contains the item). A "0" stands (as always) for the state not containing the item.

Surmise Relation Files: Surmise relation files are not available in KST Tools format, either.

Their matrices are somewhat transposed in comparison to all the other formats. Row i and column j is equal to "1" if knowing i can be surmised from knowing j, and equal to "0" otherwise. Thus, column j describes the minimal state for item j.

Surmise Function Files: Surmise function files are available only in CSV format.

The matrix is here preceded by a row which denotes the item for which the respective state is a clause.

Example

```
#SRBT v2.0 structure ASCII
3
5
# Some comment or item information
000
100
110
101
111
```

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Doignon, J.-P. & Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies, 23,* 175–196.

Doignon, J.-P. & Falmagne, J.-C. (1999). *Knowledge Spaces.* Springer Verlag, Berlin.

Hockemeyer, C. (2001). KST Tools User Manual (2nd ed.). `https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf`.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. `https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf`.

See Also

`kbase` `space_property` `kstructure`

`read_kbase` *Read base file*

Description

Read a base from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_kbase(filename, format = "auto", as.letters = TRUE)
```

Arguments

`filename` A character string specifying the name of the base file.

`format` Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", or "auto" (default).

`as.letters` logical, should the elements of the sets be letters or numbers?

Details

The `format` values `"SRBT"`, `"KST"`, and `"matrix"` refer to the different generations of file formats described in kstIO-package. The value `"auto"` (default) requests an automatic detection of the format by thje `read_XXX` function.

If `as.letters` is `TRUE` the elements of the sets are letters, otherwise numbers.

Value

A list with the following elements:

`matrix` the read structure/data as binary matrix

`sets` the read structure as object of class `kbase`

Note

In automatic format detection, the distinction between "matrix" and "KST" formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kbase, kstIO-package

Examples

```
# Produce a base file
library(kst)
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
b <- kbase(kspace(kstructure(as.pattern(DoignonFalmagne7$K, as.set=TRUE))))
write_kbase(b, "DF7.bas", "KST") # (Old) KST format
# Read file
read_kbase("DF7.bas") # Automatic format detection
read_kbase("DF7.bas", "KST") # Explicit format specification
setwd(d)
```

---

read_kdata *Read a response patterns file*

Description

Read a set of response patterns from a file. The file formats are described in the kstIO-package information page.

Usage

```
  read_kdata(filename, format = "auto", as.letters = TRUE)
```

Arguments

`filename` A character string specifying the name of the data file.

`format` Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", or "auto" (default).

`as.letters` logical, should the elements of the sets be letters or numbers?

Details

The `format` values `"SRBT"`, `"KST"`, and `"matrix"` refer to the different generations of file formats described in kstIO-package. The value `"auto"` (default) requests an automatic detection of the format by thje `read_XXX` function.

If `as.letters` is `TRUE` the elements of the sets are letters, otherwise numbers.

Value

A binary matrix with the response patterns.

Note

In automatic format detection, the distinction between `"matrix"` and `"KST"` formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

Author(s)

Cord Hockemeyer <`cord.hockemeyer@uni-graz.at`>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). `https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf`.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. `https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf`

See Also

kstIO-package

Examples

```
# Produce a data file
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
df <- as.binmat(DoignonFalmagne7$N.R)
write_kdata(df, "DF7.dat", "matrix") # matrix format (without any headers)
# Read file
read_kdata("DF7.dat") # Automatic format detection (default)
read_kdata("DF7.dat", "matrix") # Explicit ormat specification
setwd(d)
```

read_kfamset — *Read a family of sets from file*

Description

Read a family of sets from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_kfamset(filename, format = "auto", as.letters = TRUE)
```

Arguments

`filename`	A character string specifying the name of the space file.
`format`	Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", or "auto" (default).
`as.letters`	logical, should the elements of the sets be letters or numbers?

Details

The `format` values `"SRBT"`, `"KST"`, and `"matrix"` refer to the different generations of file formats described in kstIO-package. The value `"auto"` (default) requests an automatic detection of the format by thje `read_XXX` function.

`read_kfamset()` reads any knowledge space file (space, structure, basis) and ignores any file type infor in SRBT file headers.

If `as.letters` is `TRUE` the elements of the sets are letters, otherwise numbers.

Value

A list with the following elements:

`matrix`	the read structure/data as binary matrix
`sets`	the read structure as `kspace`.

Note

In automatic format detection, the distinction between `"matrix"` and `"KST"` formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

There exists no `write_kfamset` function on purpose.

Author(s)

Cord Hockemeyer <`cord.hockemeyer@uni-graz.at`>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

`space_property`, kstIO-package

Examples

```
# Produce a space file
library(kst)
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
ksp <- kspace(kstructure(as.pattern(DoignonFalmagne7$K, as.set=TRUE)))
write_kspace(ksp, "DF7.spc") # Write in (default) SRBT format
# Read file
read_kfamset("DF7.spc") # Automatic format detection (default)
read_kfamset("DF7.spc", "SRBT") # Explicit format specification
setwd(d)
```

`read_kspace` *Read a knowledge space file*

Description

Read a knowledge space from a file. The file formats are described in the kstIO-package information page.

Usage

```
  read_kspace(filename, format = "auto",
      as.letters = TRUE, close = FALSE)
```

Arguments

`filename`	A character string specifying the name of the space file.
`format`	Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", or "auto" (default).
`as.letters`	logical, should the elements of the sets be letters or numbers?
`close`	logical, determines whether a closure under union is performed on the read structure.

Details

The format values "SRBT", "KST", and "matrix" refer to the different generations of file formats described in kstIO-package. The value "auto" (default) requests an automatic detection of the format by thje read_XXX function.

If as.letters is TRUE the elements of the sets are letters, otherwise numbers.

If close is TRUE, a closure under union is computed to ensure that the returned knowledge space really is one.

Value

A list with the following elements:

matrix	the read structure/data as binary matrix
sets	the read structure as kspace.

Note

In automatic format detection, the distinction between "matrix" and "KST" formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

space_property, kstIO-package

Examples

```
# Produce a space file
library(kst)
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
ksp <- kspace(kstructure(as.pattern(DoignonFalmagne7$K, as.set=TRUE)))
write_kspace(ksp, "DF7.spc") # Write in (default) SRBT format
# Read file
read_kspace("DF7.spc") # Automatic format detection (default)
read_kspace("DF7.spc", "SRBT") # Explicit format specification
setwd(d)
```

read_kstructure *Read a knowledge structure file*

Description

Read a knowledge structure from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_kstructure(filename, format = "auto", as.letters = TRUE)
```

Arguments

filename	A character string specifying the name of the structure file.
format	Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", or "auto" (default).
as.letters	logical, should the elements of the sets be letters or numbers?

Details

The format values "SRBT", "KST", and "matrix" refer to the different generations of file formats described in kstIO-package. The value "auto" (default) requests an automatic detection of the format by thje read_XXX function.

If as.letters is TRUE the elements of the sets are letters, otherwise numbers.

Value

A list with the following elements:

matrix	the read structure/data as binary matrix
sets	the read structure as object of class kstructure.

Note

In automatic format detection, the distinction between "matrix" and "KST" formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kstructure, kstIO-package

Examples

```
# Produce a structure file
library(kst)
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
kst <- kstructure(as.pattern(DoignonFalmagne7$K, as.set=TRUE))
write_kstructure(kst, "DF7.struct") # Write in (default) SRBT format
# Read file
read_kstructure("DF7.struct") # Automatic format detection (default)
read_kstructure("DF7.struct", "SRBT") # Explicit format specification
setwd(d)
```

---

read_surmisefunction   *Read surmise function file*

Description

Read a surmise function from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_surmisefunction(filename)
```

Arguments

filename	A character string specifying the name of the base file.

Details

Surmise function files exist (so far) only in CSV format.

Value

A list with two elements:

relation	The surmise relation as object of class relation.
matrix	The incidence matrix of the surmise relation.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

See Also

kstIO-package

---

read_surmiserelation *Read surmise relation file*

Description

Read a surmise relation from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_surmiserelation(filename, format = "auto", as.letters = TRUE, close = FALSE)
```

Arguments

`filename`	A character string specifying the name of the base file.
`format`	Specification of the files format. Can be "SRBT", "matrix", "CSV", or "auto" (default).
`as.letters`	logical, should the elements of the sets be letters or numbers? Defaults to TRUE.
`close`	logical, should the relation be closed under reflexivity and transitivity? Defaults to FALSE

Details

The `format` values `"SRBT"` and `"matrix"` refer to the different generations of file formats described in kstIO-package. The value `"auto"` (default) requests an automatic detection of the format by the `read_XXX` function.

If `as.letters` is `TRUE` (default), the elements of the sets are letters, otherwise numbers.

If `close` is `TRUE` (default is `FALSE`), the relation is closed under reflexivity and transitivity, otherwise it is returned as is.

Value

A list with two elements:

`relation`	The surmise relation as object of class `relation`.
`matrix`	The incidence matrix of the surmise relation.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kstIO-package

Examples

```
# Produce a relation file
library(kst)
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
r <- as.relation(kspace(kstructure(as.pattern(DoignonFalmagne7$K, as.set=TRUE))))
write_surmiserelation(r, "DF7.rel", "SRBT") # SRBT format
# Read file
read_surmiserelation("DF7.rel") # Automatic format detection
read_surmiserelation("DF7.rel", "SRBT") # Explicit format specification
setwd(d)
```

write_kbase *Write a base file*

Description

Write a base to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_kbase(x, filename, format = "SRBT")
```

Arguments

x	The data to be written, either a binary matrix or an object of kbase class.
filename	A character string specifying the name of the base file.
format	Specification of the files format. Can be "SRBT" (default), "KST", "CSV", or "matrix".

Details

The format values "SRBT", "KST", and "matrix" refer to the different generations of file formats described in kstIO-package.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kbase, kstIO-package

Examples

```
# Obtain data to write from the 'pks' package
library(kst)
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
b <- kbase(kspace(kstructure(as.pattern(DoignonFalmagne7$K, as.set=TRUE))))
# Write base to file
write_kbase(b, "DF7.bas") # Write in (default) SRBT format
write_kbase(b, "DF7.bas", "KST") # (Old) KST format
setwd(d)
```

write_kdata *Write a knowledge space theory file*

Description

Write a data set to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_kdata(x, filename, format = "SRBT")
```

Arguments

x	The data to be written, as a binary matrix.
filename	A character string specifying the name of the data file.
format	Specification of the files format. Can be "SRBT" (default), "KST", "CSV", or "matrix".

Details

The `format` values `"SRBT"`, `"KST"`, and `"matrix"` refer to the different generations of file formats described in kstIO-package.

Author(s)

Cord Hockemeyer `<cord.hockemeyer@uni-graz.at>`

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). `https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf`.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* `https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf`

See Also

kstIO-package

Examples

```
# Obtain data to write from the 'pks' package
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
df <- as.binmat(DoignonFalmagne7$N.R)
# Write data to file
write_kdata(df, "DF7.dat") # Write in (default) SRBT format
write_kdata(DoignonFalmagne7$K, "DF7.dat", "matrix") # matrix format (without any headers)
setwd(d)
```

`write_kspace` *Write a knowledge space file*

Description

Write a knowledge space to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_kspace(x, filename, format = "SRBT")
```

Arguments

`x` The data to be written, either a binary matrix or an object of `kspace` class.

`filename` A character string specifying the name of the base file.

`format` Specification of the files format. Can be "SRBT" (default), "KST", "CSV", or "matrix".

Details

The format values "SRBT", "KST", and "matrix" refer to the different generations of file formats described in kstIO-package.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

space_property, kstIO-package

Examples

```
# Obtain data to write from the 'pks' package
library(kst)
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
ksp <- kspace(kstructure(as.pattern(DoignonFalmagne7$K, as.set=TRUE)))
# Write space to file
write_kspace(ksp, "DF7.spc") # Write in (default) SRBT format
write_kspace(DoignonFalmagne7$K, "DF7.spc", "KST") # Write the matrix directly in (old) KST format
setwd(d)
```

---

write_kstructure *Write a knowledge structure file*

Description

Write a knowledge structure to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_kstructure(x, filename, format = "SRBT")
```

Arguments

`x` The data to be written, either a binary matrix or an object of `kstructure` class.

`filename` A character string specifying the name of the base file.

`format` Specification of the files format. Can be "SRBT" (default), "KST", "CSV", or "matrix".

Details

The `format` values `"SRBT"`, `"KST"`, and `"matrix"` refer to the different generations of file formats described in kstIO-package.

Author(s)

Cord Hockemeyer `<cord.hockemeyer@uni-graz.at>`

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). `https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf`.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* `https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf`

See Also

`kstructure`, kstIO-package

Examples

```
# Obtain data to write from the 'pks' package
library(kst)
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
kst <- kstructure(as.pattern(DoignonFalmagne7$K, as.set=TRUE))
# Write structure to file
write_kstructure(kst, "DF7.struct") # Write in (default) SRBT format
# Write the matrix directly in (old) KST format
write_kstructure(DoignonFalmagne7$K, "DF7.struct", "KST")
setwd(d)
```

write_surmisefunction *Write a surmise function file*

Description

Write a surmise function to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_surmisefunction(x, filename)
```

Arguments

x	The data to be written, either a quadratic binary matrix or an object of `relation` class.
filename	A character string specifying the name of the base file.

Details

SUrmise function can (so far) be stored only in CSV format.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

See Also

kstIO-package

write_surmiserelation *Write a (surmise) relation file*

Description

Write a surmise relation to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_surmiserelation(x, filename, format = "SRBT")
```

Arguments

`x` The data to be written, either a quadratic binary matrix or an object of `relation` class.

`filename` A character string specifying the name of the base file.

`format` Specification of the files format. Can be "SRBT" (default), "CSV", or "matrix".

Details

The `format` values `"SRBT"` and `"matrix"` refer to the different generations of file formats described in kstIO-package.

Author(s)

Cord Hockemeyer `<cord.hockemeyer@uni-graz.at>`

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). `https://kst.hockemeyer.at/techreports/KST-Tools_TechRep_FWF01.pdf`.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* `https://kst.hockemeyer.at/techreports/SRBT-Tools_TechRep_FWF01.pdf`

See Also

kstIO-package

Examples

```
# Obtain data to write from the 'pks' package
library(kst)
library(pks)
d <- getwd()
setwd(tempdir())
data(DoignonFalmagne7)
r <- as.relation(kspace(kstructure(as.pattern(DoignonFalmagne7$K, as.set=TRUE))))
# Write surmise relation to file
write_surmiserelation(r, "DF7.bas") # Write in (default) SRBT format
write_surmiserelation(r, "DF7.bas", "matrix") # matrix format
setwd(d)
```

Index