

Package ‘gogamer’

September 8, 2016

Type Package

Title Go Game Data Parser

Version 0.4.3

Maintainer Kota Mori <kmori05@gmail.com>

Description Easy and flexible interface for manipulating go game (weiqi, baduk) data.
The package features a reader function for SGF (smart go format) text files,
and a set of plotting functions that draw go board images.

License MIT + file LICENSE

Depends R (>= 3.1)

Imports dplyr, ggplot2, graphics, gridExtra, magrittr, scales, stats,
stringr, Rcpp, utils,

LazyData TRUE

RoxygenNote 5.0.1

Suggests testthat

URL <https://github.com/kota7/gogamer>

BugReports <https://github.com/kota7/gogamer/issues>

LinkingTo Rcpp

NeedsCompilation yes

Author Kota Mori [aut, cre]

Repository CRAN

Date/Publication 2016-09-08 07:33:47

R topics documented:

addlabels	2
addmarkers	3
addstones	4
addterritory	4
ggkifu	5

ggoban	6
gogame	6
gogamer	8
gogame_graphics	8
gokifu	10
gostate	10
is.ggkifu	11
is.ggoban	12
is.gogame	12
kifu	13
kifunote	13
mimiaka	14
parse_sgf	14
plot.gokifu	15
plot.gostate	15
prune_sgf	16
read_sgf	16
saikoyo	17
set_gamepath	17
star_position	18
stateat	18
suggested_size	19

Index 20

addlabels	<i>Add text label on board</i>
-----------	--------------------------------

Description

Add text label on board

Usage

```
addlabels(gg, x, y, label, color = NULL, ...)
```

Arguments

gg	ggplot object
x, y	integer vectors of stone locations
label	character vector of labels
color	integer vector of stone colors
...	graphic parameters

Value

[ggoban](#) object

See Also

[ggoban](#), [graphic_parameters](#)

Examples

```
ggoban(19) %>%  
  addstones(c(16, 4), c(16, 3), c(1, 2)) %>%  
  addlabels(c(4, 3), c(17, 16), c("a", "b"))
```

addmarkers

Add markers

Description

Add markers

Usage

```
addmarkers(gg, x, y, color, marker = 17, ...)
```

Arguments

gg	ggplot object
x, y	integer vectors of stone locations
color	integer vector of stone colors
marker	scalar integer indicating the shape of marker
...	graphic parameters

Value

[ggoban](#) object

See Also

[ggoban](#), [graphic_parameters](#)

Examples

```
ggoban(19) %>%  
  addstones(10, 10, 1) %>%  
  addmarkers(10, 10, 1)
```

addstones *Add stones to go board*

Description

Add stones to go board

Usage

```
addstones(gg, x, y, color, number = NULL, ...)
```

Arguments

gg	ggplot object
x, y	integer vectors of stone locations
color	integer vector of stone colors
number	integer vector of numbers on stones
...	graphic parameters

Value

[ggoban](#) object

See Also

[ggoban](#), [graphic_parameters](#)

Examples

```
ggoban(19) %>%  
  addstones(c(10, 11), c(10, 10), c(1, 2), c(10, 11))
```

addterritory *Add territory markers*

Description

Add territory markers

Usage

```
addterritory(gg, x, y, color, ...)
```

Arguments

gg	ggplot object
x, y	integer vectors of stone locations
color	integer vector of stone colors
...	graphic paramters

Value

ggoban object

See Also

[ggoban](#), [graphic_parameters](#)

ggkifu

One-page kifu image

Description

One-page kifu image

Usage

```
ggkifu(board, note, boardsize, heights = NULL, savesize = NULL)
```

Arguments

board	ggplot object of board
note	ggplot object of outside note
boardsize	integer of board size
heights	Numeric vector of size two, indicates the vertical size ratio between board and note
savesize	Numeric vector of size two, indicating appropriate pair of width and height when saving

Value

ggkifu object

ggoban

Draw go board

Description

Draw go board

Usage

```
ggoban(boardsize, ...)
```

Arguments

boardsize	integer of boardsize
...	graphic parameters

Value

Object of class ggoban, which inherits ggplot

See Also

[graphic_parameters](#)

Examples

```
ggoban(19)
```

gogame

Go game object

Description

gogame class object capsulizes a go game record. It keeps stores such game plays, outcomes as well as other information such as player names and date. The object supports various methods for creating board images and Kifu documents.

Usage

```
gogame(properties, gametree)
```

Arguments

properties	a list of game properties
gametree	a list

Details

This is a constructor of gogame class object. It is mainly designed to be called from `parse_sgf` function, which interprets text of sgf format and creates the corresponding gogame object.

The object can produce two kinds of game images. The first is a board snapshot at an arbitrary timing. `stateat` is for computing the board configuration and `plotat` is for drawing the board image. The second kind of images is the kifu document that summarize a range of moves in a page. `kifu` is prepared for computing the move sequence and `kifuplot` is for drawing the image.

The constructor takes two mandatory arguments: `properties` and `gametree`. `properties` is a named list of game meta information. Currently, following names are recognized (variable type in parentheses). One may provide additional information so long as it does not contradict names of other information. Check `str(mimiaka)` to see a list of variable names already in use.

`boardsize` board size (integer)

`whitename, whiterank, blackname, blackrank` player names and ranks (character)

`komi` Komi (numeric)

`handicap` number of handicap stones (integer)

`result` game outcome (character)

`date, place` game date and location (character)

`event, round` competition or event name and round (character)

`rule` rule (character)

`gametree` stores game plays, setups, comments, and territory counts. To accommodate the cases of games with branches, the object employs a tree data structure. `gametree` should contain potentially four data sets. Parentheses list the variable names to be contained.

`transition` Transition of board configuration. a positive value means a stone is added, negative means removed. Absolute values of value indicate the stone color (1: black, 2: white) (x, y, value, move, nodeid)

`move` Moves and setups. `ismove` indicates whether it is a move or setup (x, y, color, ismove, move, nodeid)

`point` Territories. (x, y, color, move, nodeid)

`comment` Comments made during the game. (comment, move, nodeid)

`nodeid` variable in each data indicates the tree node to which the row belongs to.

`gametree` also include three variables that describe tree structure.

`parent` integer vector that points parent of each node. The first node is always assumed to be the root node and its parent is zero

`children` list of integer vector that points children of each node

`leaf` integer vector of leaf nodes (i.e. no children)

For example, when there is only one node, `nodeid` variable of all data should be 1, and `parent` is 0 (there is only the root node), `children` has unique entry and it is integer vector of length zero, and `leaf` is 1. When some of the three components are missing, then the function tries the best to recover from the supplied information

When `transition` is missing, then the function tries to compute it using `move` and tree structure information.

Value

gogame object

See Also

[read_sgf](#), [parse_sgf](#), [plotat](#), [stateat](#), [kifu](#), [kifuplot](#)

gogamer

gogamer: Go Game Data Parser

Description

Easy and flexible interface for manipulating go game (weiqi, baduk) data. The package features a reader function for SGF (smart go format) text files, and a set of plotting functions that draw go board images.

gogame_graphics

Graphic parameters for go game images

Description

Customize go images

Details

Go game images can be customized flexibly by passing appropriate graphic parameters to functions that create images. The complete list of available parameters are given below (default values in parentheses).

`targetwidth` Width to be used when exporting the image. Other size parameters are adjusted in accordance with this parameter (5)

`colortheme` Predefined color template. Currently supports: "standard", "bw", "pastel", "dark" and "crystal" (NULL)

`boardsize` Size of board (19)

`xlim, ylim` Use to draw a partial board image (NULL, NULL)

`boardcolor` Color of board ("ecf0b7")

`boardalpha` Transparency of board (1)

`gridcolor` Color of grid lines ("262626")

`starcolor` Color of stars on board ("262626")

`starsize` Size of stars on board (1.5)

`axislabelcolor` Color of axis labels ("262626")

`axislabelsize` (Size of axis labels 3.5)

xlabel, ylabel Axis labels (LETTERS[-9], as.character(1:25))
 axislabels If true, axis labels are added (TRUE)
 blackcolor, whitecolor Stone colors on the board ("#111111", "#f5f5f5")
 stonealpha Transparency of stones (1)
 stonesize Size of stones (6)
 stonelinecolor Color of stone line ("#101010")
 stonelinewidth Width of stone line (0.7)
 blacknumbercolor, whitenumcolor Color of numbers on stones ("#f0f0f0", "#0f0f0f")
 numbersize Size of numbers on stones (3)
 blacklabelcolor, whitelabelcolor, emptylabelcolor Colors of labels put on stones and empty points ("#f0f0f0", "#0f0f0f", "#262626")
 labelsize Size of labels (3.5)
 emptylabelshadowsize Size of background for labels put on empty points (5)
 blackmarkercolor, whitemarkercolor, emptymarkercolor Colors of markers put on stones and empty points ("#f0f0f0", "#0f0f0f", "#262626")
 markersize Size of markers (2.5)
 lastmovemarker Shape of marker indicating the last move (3)
 territorysize Size of territory indicators (2)
 territoryshape Shape of territory indicators. Recommended to use an integer between 21 and 25 (21)
 territorylinecolor Color of lines of territory indicators ("#262626")
 territorylinewidth Line width of territory indicators (0.5)
 blackmark, whitemark, emptymark Letters used to indicate stones and empty points when printing on console ("@", "0", "+")
 notebackcolor Background color of outside note for kifu ("#eff7df")
 notetextcolor Color of text used in outside note for kifu ("#101010")
 notestonesize Size of stones in outside note for kifu (5)
 notenumbersize Size of numbers on stones in outside note for kifu (2.5)
 notetextsize Size of texts in outside note for kifu (3)
 moveperrow Number of moves shown for each line of outside note for kifu (8)
 adjustorigin If true, move numbers are deducted by a multiple of hundred when appropriate (TRUE)
 adjustsizeonboard Adjust all sizes for go board image in accordance with the board size. Not recommended to change (TRUE)
 adjustsizeonnote Adjust all sizes for outside note in accordance with the board size (FALSE)

Examples

```

ggoban(19, boardcolor = "yellow", gridcolor = "blue") %>%
  addstones(c(10, 11), c(10, 10), c(1, 2), c(1, 2),
            blackcolor = "green", whitecolor = "red",
            blacknumbercolor = "black",
            whitenumcolor = "black")

```

gokifu	<i>Kifu (go game record) for a range of moves</i>
--------	---

Description

Kifu (go game record) for a range of moves

Usage

```
gokifu(unnumbered, numbered, noted, comment, boardsize)
```

Arguments

unnumbered	stones to be shown with no number on the board. These typically are stones put before the move range of the kifu. It also includes setup stones in the range (data.frame)
numbered	moves to be numberd on the board (data.frame)
noted	moves to be listed outside the board (data.frame)
comment	comments during the move range of the game (data.frame)
boardsize	board size (integer)

Value

gokifu object

gostate	<i>Go game state</i>
---------	----------------------

Description

gostate object stores a go game state including stone configuration on the board and the numbers of prisoners.

Usage

```
gostate(board, boardsize, b_captured, w_captured, movenumber = NULL,
        lastmove = NULL, points = NULL, comment = NULL)
```

Arguments

board	a data.frame representing stone allocation. It must have variables (x, y, color)
boardsize	board size (integer)
b_captured, w_captured	numbers of captured stone (integer)
movenumber	integer of move number
lastmove	integer vector of length three that indicates the last move location and color in the order of (x, y, color)
points	data.frame of territory locations
comment	character vector of comments

Value

gostate object

Examples

```
gostate(data.frame(x = 4, y = 4, color = 1), 19, 0, 0)
```

is.ggkifu	<i>Check if the object is ggkifu class</i>
-----------	--

Description

Check if the object is ggkifu class

Usage

```
is.ggkifu(x)
```

Arguments

x	R object
---	----------

Value

logical

`is.ggoban`*Check if object is ggoban class*

Description

Check if object is ggoban class

Usage

```
is.ggoban(x)
```

Arguments

x R object

Value

Logical. True if and only if obj inherits ggoban class.

`is.gogame`*Check if the object is gogame class*

Description

Check if the object is gogame class

Usage

```
is.gogame(x)
```

Arguments

x R object

Value

logical

kifu *Kifu for certain move range*

Description

Kifu for certain move range

Usage

```
kifu(x, from = 1L, to = 99L, restart = NA_integer_)
```

```
kifuplot(x, from = 1L, to = 100L, restart = NA_integer_, ...)
```

Arguments

x	gogame object
from, to	Positive integers. Range of moves
restart	Positive integer. If supplied, this number is used as the smallest move number in the range. If not supplied, original move numbers are used as they are.
...	graphic parameters

Value

kifu returns a [gokifu](#) object

kifuplot returns a [ggkifu](#) object

Examples

```
kifu(saikoyo)
kifuplot(mimiaka, 127, 150)
```

kifunote *Draw outside note of kifu*

Description

Draw outside note of kifu

Usage

```
kifunote(x, ...)
```

Arguments

x	kifu object
...	graphic parameter

Value

ggplot object

mimiaka	<i>Mimiaka (ear-reddening) game</i>
---------	-------------------------------------

Description

Go game known as "Mimiaka (ear-reddening) game", between Genan Inseki and Kuwabara Shusaku

Usage

mimiaka

Format

gogame object

Source

- Power, John. *Invincible: The games of Shusaku*. Kiseido Publishing Company 1982.
- <http://senseis.xmp.net/?EarReddeningMove>

parse_sgf	<i>Parse text of the smart go format.</i>
-----------	---

Description

Parse text of the smart go format.

Usage

parse_sgf(sgf)

Arguments

sgf scalar character of sgf text.

Value

gogame object

plot.gokifu	<i>Draw kifu</i>
-------------	------------------

Description

Draw kifu

Usage

```
## S3 method for class 'gokifu'  
plot(x, y, ...)
```

Arguments

x	kifu object
y	not in use (just for argument consistency with generic function)
...	graphical paramters

Value

ggkifu object

plot.gostate	<i>Draw go board state as graphic</i>
--------------	---------------------------------------

Description

Draw go board state as graphic

Usage

```
## S3 method for class 'gostate'  
plot(x, y, marklast = TRUE, markpoints = FALSE, ...)
```

Arguments

x	gostate object
y	not in use
marklast	logical indicating if last move should be marked
markpoints	logical indicating if territories should be marked
...	graphic parameters

Value

ggoban object, which inherits ggplot class

See Also[gogame_graphics](#)**Examples**

```
stateat(saikoyo, 116) %>% plot()
```

prune_sgf	<i>Remove branches from SGF text</i>
-----------	--------------------------------------

Description

Remove branches from SGF text

Usage

```
prune_sgf(sgf, keepfirst = TRUE)
```

Arguments

sgf	Scalar character formatted as SGF
keepfirst	Logical. If TRUE, keep the first branch. Otherwise, keep the last branch.

Value

Scalar character of SGF text where branches are removed

read_sgf	<i>Read and parse a SGF file</i>
----------	----------------------------------

Description

Read and parse a SGF file

Usage

```
read_sgf(path, ...)
```

Arguments

path	character string of the path to a smart go format (SGF) file. Can be local or online
...	arguments passed to readLines

Value

[gogame](#) object

saikoyo	<i>sai vs toya koyo</i>
---------	-------------------------

Description

Go game between Sai and Toya Koyo played online in the comic "Hikaru no Go"

Usage

saikoyo

Format

gogame object

Details

The game played between Sai and Toya Koyo on internet in the comic "Hikaru no Go", Episode 112-115. The game is originally played by Japanese professionals Yoda Notimoto and Rin Kaiho on May 1st, 1997. In the comic, Toya Koyo resigns when he realizes that the game would eventually end with his loss by 0.5 point.

Source

Hotta, Yumi, Obata, Takeshi, and Umezawa, Yukari. Hikaru no Go: Gojasu Characters Guide. Shueisha 2002.

set_gamepath	<i>Switch path of go game</i>
--------------	-------------------------------

Description

Switch path of a go game. Paths are indexed by integers starting at one. If pathid exceeds the number of paths stored in the game, the function throws an error.

Usage

```
set_gamepath(x, pathid = 1L)
```

Arguments

x	gogame object
pathid	integer

Value

gogame object

star_position	<i>Get location of stars</i>
---------------	------------------------------

Description

Get location of stars

Usage

```
star_position(boardsize)
```

Arguments

boardsize	integer of boardsize
-----------	----------------------

Value

data.frame with variables x and y

stateat	<i>Go board status at a move number</i>
---------	---

Description

This function obtains the board state at the move number. The result is stored in a [gostate](#) object.

Usage

```
stateat(x, at)
```

```
plotat(x, at, ...)
```

Arguments

x	gogame object
at	integer of the move number
...	arguments passed to plot.gostate

Value

stateat returns a [gostate](#) object

plotat returns a ggplot object

Examples

```
stateat(saikoyo, 116)
plotat(mimiaka, 127)
```

suggested_size	<i>Suggested size used for saving</i>
----------------	---------------------------------------

Description

Suggested size used for saving

Usage

suggested_size(obj)

Arguments

obj ggoban or ggkifu object

Value

Numeric vector of size two

Index

*Topic **datasets**

mimiaka, [14](#)
saikoyo, [17](#)

addlabels, [2](#)
addmarkers, [3](#)
addstones, [4](#)
addterritory, [4](#)

ggkifu, [5](#), [13](#), [15](#)
ggoban, [2–5](#), [6](#)
gogame, [6](#), [14](#), [16](#)
gogame_graphics, [8](#), [16](#)
gogamer, [8](#)
gogamer-package (gogamer), [8](#)
gokifu, [10](#), [13](#)
gostate, [10](#), [18](#)
graphic_parameters, [3–6](#)
graphic_parameters (gogame_graphics), [8](#)

is.ggkifu, [11](#)
is.ggoban, [12](#)
is.gogame, [12](#)

kifu, [7](#), [8](#), [13](#)
kifunote, [13](#)
kifuplot, [7](#), [8](#)
kifuplot (kifu), [13](#)

mimiaka, [14](#)

parse_sgf, [7](#), [8](#), [14](#)
plot.gokifu, [15](#)
plot.gostate, [15](#), [18](#)
plotat, [7](#), [8](#)
plotat (stateat), [18](#)
prune_sgf, [16](#)

read_sgf, [8](#), [16](#)
readLines, [16](#)

saikoyo, [17](#)
set_gamepath, [17](#)
star_position, [18](#)
stateat, [7](#), [8](#), [18](#)
suggested_size, [19](#)