Package: shikakusphere (via r-universe)

October 30, 2026

Title Miscellaneous Functions for Japanese Mahjong

Version 0.1.4

Description A collection of miscellaneous functions for Japanese mahjong that wraps C++ sources of 'shanten-number' <https://github.com/tomohxx/shanten-number> and 'cmajiang' <https://github.com/TadaoYamaoka/cmajiang>.

License GPL (>= 3)

URL https://paithiov909.github.io/shikakusphere/

BugReports https://github.com/paithiov909/shikakusphere/issues

Depends R (>= 4.1)

Imports generics, magick, methods, purrr, Rcpp, rlang, stringi, tibble, vctrs, withr

Suggests dplyr, patrick, rsvg, testthat (>= 3.0.0)

LinkingTo Rcpp

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs libmagick++-dev gsfonts libicu-dev libssl-dev

Repository https://paithiov909.r-universe.dev

Date/Publication 2026-02-02 02:45:24 UTC

RemoteUrl https://github.com/paithiov909/shikakusphere

RemoteRef HEAD

RemoteSha 4f6fb0a215b98e81e1bdea3d2ae79d37665fe86d

Contents

calc_defen . . . 2
calc_xiangting . . . 3
collect_tingpai . . . 4
convert-from-mjai . . . 5
default_rule . . . 6
detect-tiles . . . 6
hand2img . . . 7
hupai . . . 7
int2tile . . . 8
is_valid_tile . . . 8
lineup . . . 9
lipai . . . 9
modify_rule . . . 10
paistr . . . 10
parse_hupai . . . 12
proceed . . . 12
rand_hands . . . 13
rand_qipai . . . 14
ranked_game . . . 14
testdat_xiangting . . . 15
tiles . . . 15
trans_tile . . . 16

Index **17**

`calc_defen` *Calculate score of hand*

Description

Calculate score of hand

Usage

```
calc_defen(
  pai,
  baopai,
  libaopai = "",
  rongpai = "",
  rule = default_rule(),
  zhuangfeng = c("z1", "z2", "z3", "z4"),
  menfeng = c("z2", "z3", "z4", "z1"),
  lizhi = c("none", "lizhi", "double-lizhi"),
  yifa = FALSE,
  qianggang = FALSE,
  lingshang = FALSE,
```

```
  haidi = c("none", "haidimoyue", "hedilaoyu"),
  tianhe = c("none", "tianhe", "dihe"),
  changbang = 0L,
  lizhibang = 0L
)
```

Arguments

pai	A string scalar. This param is not vectorized.
baopai	A character vector. "Dora" indicators.
libaopai	A character vector. "Ura-dora" indicators. Leave empty if there is no libaopai.
rongpai	A string scalar such as "m1=". Leave empty if there is no rongpai.
rule	A list; a rule set. Defaults to default_rule().
zhuangfeng	A string scalar; "ba-kaze" tile.
menfeng	A string scalar; "ji-kaze" tile.
lizhi	A string scalar. Either "none", "lizhi", or "double-lizhi".
yifa	A logical scalar; flag for "ippatsu".
qianggang	A logical scalar; flag for "chankan".
lingshang	A logical scalar; flag for "rinshan-kaihou".
haidi	A string scalar. Either "none", "haidimoyue", or "hedilaoyu".
tianhe	A string scalar; Either "none", "tianhe", or "dihe".
changbang	An integer scalar; the number of counter sticks that indicates "honba".
lizhibang	An integer scalar; the number of 1,000-point sticks on the table.

Value

A data frame.

Examples

```
calc_defen("m345567p234s3378", baopai = "z1", rongpai = "s9=")
```

---

calc_xiangting *Calculate xiangting number of hands*

Description

Calculate xiangting number of hands

Usage

```
calc_xiangting(pai)
```

Arguments

pai	A character vector.

Value

A tibble.

Examples

```
calc_xiangting(c("m345567p234s3378", "p222345z1234567"))
```

---

collect_tingpai *Collect tingpais for hands*

Description

Collect tiles that can decrease the xiangting number of hands if they are drawn. In case there are no tingpais for the hand, the function warns about it and returns an empty character vector.

Usage

```
collect_tingpai(pai)
```

Arguments

pai	A character vector.

Value

A list of character vectors.

Examples

```
collect_tingpai(c("m345567p234s3378", "p222345z1234567"))
```

convert-from-mjai *Convert symbols from 'mjai' format*

Description

Converts symbols from 'mjai' format into 'cmajiang' format. To read 'mjai' format logs, use `convlog::read_tenhou6()`.

Usage

```
mjai_target(perspective, target)

mjai_conv(type, pai, consumed, target)
```

Arguments

`perspective`	An integer vector that indicates the plyaer's perspective.
`target`	An integer vector that represents the target of 'mjai' events.
`type`	A character vector that represents the type of 'mjai' events.
`pai`	A character vector of tiles to be converted from 'mjai' format into 'cmajiang' format.
`consumed`	A list of character vectors that represent the 'consumed' tiles in 'mjai' format.

Value

A character vector.

Examples

```
if (requireNamespace("dplyr", quietly = TRUE)) {
  data("ranked_game")
  ranked_game[["paifu"]] |>
    dplyr::filter(
      type %in% c("tsumo", "dahai", "chi", "pon", "daiminkan", "kakan", "ankan")
    ) |>
    dplyr::mutate(
      target = mjai_target(actor, target),
      pai = trans_tile(pai)
    ) |>
    dplyr::group_by(game_id, round_id, actor) |>
    dplyr::mutate(pai = mjai_conv(type, pai, consumed, target)) |>
    dplyr::ungroup()
}
```

default_rule *Create default rule set*

Description

Create default rule set

Usage

```
default_rule()
```

Value

A list.

See Also

https://github.com/kobalab/majiang-core/wiki/%E3%83%AB%E3%83%BC%E3%83%AB

detect-tiles *Detect specific tiles*

Description

Detect specific tiles

Usage

```
tile2rank(x)

tile2suit(x)

is_fulou(x)

is_tsumogiri(x)

is_lizhi(x)

is_suit(x, suit = c("m", "p", "s", "z"))

is_yaojiu(x, negate = FALSE)

is_zhongzhang(x)

is_suzhi(x, suzhi = c("23", "34", "45", "56", "67", "78"))
```

Arguments

x	A character vector.
suit	A string scalar. Either "m", "p", "s", or "z".
negate	A logical scalar.
suzhi	A string scalar. Either "23", "34", "45", "56", "67", or "78".

Value

- For `tile2rank` and `tile2suit`: a character vector (an empty string for "_").
- For other functions: a logical vector.

`hand2img` *Plot a player's hand as an image*

Description

This function is a short hand for `paistr(pai) |> plot()`

Usage

```
hand2img(pai, ...)
```

Arguments

pai	A character vector.
...	Other arguments passed to `plot()`.

See Also

`paistr`

`hupai` *List of hupai*

Description

List of hupai

Usage

```
hupai
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 54 rows and 3 columns.

Details

A tibble that contains the list of hupai.

The list includes the following columns:

- `id`: The id of the hupai.
- `en`: The English name of the hupai.
- `jp`: The Japanese name of the hupai.

`int2tile` *Convert integers to tiles*

Description

Convert integers to tiles

Usage

```
int2tile(x = seq_len(38) - 1, origin = c("zero", "one"))
```

Arguments

`x`	An integer vector.
`origin`	A string scalar. Either "zero" or "one".

Value

A factor.

Examples

```
int2tile(c(0, 1, 25, 37))
```

`is_valid_tile` *Check if each element is a valid tile*

Description

Check if each element is a valid tile

Usage

```
is_valid_tile(x)
```

Arguments

x	A character vector to be checked.

Value

A logical vector.

---

lineup *Line up tiles*

Description

Arrange all tiles from a data frame.

Usage

```
lineup(x)
```

Arguments

x	A data frame with columns `id`, `tile`, and `n`.

Value

An object of `x$tile` class.

Examples

```
rand_hands()(3) |>
  paistr() |>
  tidy() |>
  lineup()
```

---

lipai *Compose hands from character vectors*

Description

Compose hands from character vectors while ignoring invalid tiles. This function can handle any number of tiles in each hand, but cannot more than 5 identical tiles. If there are more than 5 identical tiles, arises an error.

Usage

```
lipai(x)
```

Arguments

x	A list of character vectors or a character vector.

Value

A character vector.

Examples

```
lipai(list(c("m1", "m2", "m3"), c("p1", "p2", "p3")))
```

---

modify_rule *Modify the default rule set by patch*

Description

Modify the default rule set by patch

Usage

```
modify_rule(patch = list())
```

Arguments

patch	A list.

Value

A list.

---

paistr paistr *vector*

Description

Create a `paistr` vector from a character vector that represents player's hands.

Usage

```
paistr(x = character())

is_paistr(x)

## S3 method for class 'skksph_paistr'
calculate(x, ...)

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

## S3 method for class 'skksph_paistr'
tidy(x, ...)
```

Arguments

x	• For `paistr()`: A character vector. • For `is_paistr()`: An object to test. • For `calculate()`: An object to summarize. • For `plot()`: An object to plot as an image. • For `tidy()`: An object to tidy up.
...	Other arguments for `calculate()`, `plot()` or `tidy()`.
y	• For `plot()`: Not used.

Details

Note that the validation of this function is not so strict. For example, `paistr("z0")` still produces a valid `paistr` vector even though "z0" is not a tile that actually exists. These `paistr` are simply ignored by the 'cmajiang' function wrapper.

The number of tiles displayed when `print()` is not always accurate, so if you need to count the actual number of tiles, use `tidy()`.

Value

- For `paistr()`: An object of class `skksph_paistr`.
- For `is_paistr()`: A logical scalar.
- For `calculate()`: A tibble.
- For `plot()`: A bitmap image that internally converted by `magick::image_read_svg()` is invisibly returned.
- For `tidy()`: A tibble.

Examples

```
pai <- paistr(c("m055z7z7,m78-9,z5555,z666=", "m123s789z1117*,p5550"))
print(pai)
is_paistr(pai)
```

```
calculate(pai)
tidy(pai)
```

`parse_hupai` *Parse chains of hupai ids*

Description

Parse comma separated chains of hupai ids into a list of factors.

Usage

```
parse_hupai(str, lang = c("en", "jp"))
```

Arguments

`str`	A character vector.
`lang`	A string scalar. Either "en" or "jp".

Value

A list of factors.

`proceed` *Proceed state of hands*

Description

Proceed state of hands

Usage

```
proceed(qipai, zimo, dapai, accumulate = FALSE)
```

Arguments

`qipai`	A list of character vectors where each element represents a player's qipai.
`zimo`	A list of character vectors where each element represents a players's zimo.
`dapai`	A list of character vectors where each element represents a players's dapai.
`accumulate`	Logical. If true, returns accumulated result.

Value

A (list of) `paistr` vector(s).

Examples

```
if (requireNamespace("dplyr", quietly = TRUE)) {
  qipai <-
    paistr("m123456789p1s1z11") |>
    tidy() |>
    dplyr::mutate(tile = as.character(tile)) |>
    lineup()
  zimo <- list(c("p2", "p3"))
  dapai <- list(c("z1", "z1"))
  proceed(qipai, zimo, dapai)
}
```

---

rand_hands *Create a function to randomly generate hands*

Description

Create a function to randomly generate hands

Usage

```
rand_hands(
  hupai = c("pinghe", "zhuangfeng", "menfeng", "fanpai", "duanyaojiu", "yibeikou",
     "sansetongshun", "yiqitongguan", "hunquandaiyaojiu", "qiduizi", "duiduihu",
    "sananke", "sangangzi", "sansetongke", "hunlaotou", "xiaosanyuan", "hunyise",
   "chunquandaiyaojiu", "erbeikou", "qingyise", "guoshiwushuang", "sianke", "dasanyuan",
   "xiaosixi", "dasixi", "ziyise", "lvyise", "qinglaotou", "sigangzi", "jiulianbaodeng"),
  zhuangfeng = c("z1", "z2", "z3", "z4"),
  menfeng = c("z2", "z3", "z4", "z1"),
  rule = default_rule(),
  seed = sample.int(10000, 1)
)
```

Arguments

hupai	A string scalar.
zhuangfeng	A string scalar; "ba-kaze" tile.
menfeng	A string scalar; "ji-kaze" tile.
rule	A list; a rule set. Defaults to `default_rule()`.
seed	An integer scalar. Used for initializing the random number generator.

Value

A function inheriting class `purrr_function_compose` that takes single arguments `n`: number of hands to be generated.

Examples

```
rand_hands()(3)
```

`rand_qipai` *Generate hands at random*

Description

Generate hands at random

Usage

```
rand_qipai(nrow, rule = default_rule(), seed = sample.int(10000, 1))
```

Arguments

`nrow`	An integer scalar.
`rule`	A list; a rule set. Defaults to `default_rule()`.
`seed`	An integer scalar. Used for initializing the random number generator.

Value

A tibble that consists of 4 columns of `paistr` vector.

Examples

```
rand_qipai(3)
```

`ranked_game` *Sample data of mahjong game log*

Description

Sample data of mahjong game log

Usage

```
ranked_game
```

Format

An object of class `list` of length 3.

Details

A named list that contains sample data of mahjong game log. The data was coverted from 'tenhou.net/6' format using `read_tenhou6()` function in convlog package.

Source

https://github.com/Equim-chan/mjai-reviewer/blob/master/convlog/tests/testdata/ranked_game.json

---

testdat_xiangting *Test cases for* `calc_xiangting()`

Description

Test cases for `calc_xiangting()`

Usage

```
testdat_xiangting
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 40000 rows and 4 columns.

Details

A tibble that contains test cases for `calc_xiangting()`. Contents may vary depending on package version.

Source

https://github.com/TadaoYamaoka/cmajiang/tree/main/tests_cpp/data

---

tiles *Lookup table for tiles conversion*

Description

Lookup table for tiles conversion

Usage

```
tiles
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 38 rows and 4 columns.

Details

A tibble that contains the lookup table for tiles conversion.

`trans_tile` *Translate tiles from one format to another*

Description

Translate tiles from one format to another

Usage

```
trans_tile(
  x,
  from = c("mjai", "tenhou_int", "id", "cmajiang"),
  to = c("cmajiang", "mjai", "tenhou_int", "id")
)
```

Arguments

`x`	A vector of tiles to be translated.
`from`	A string scalar. Either "mjai", "tenhou_int", "id", or "cmajiang".
`to`	A string scalar. Either "cmajiang", "mjai", "tenhou_int", or "id".

Value

- For `to = "cmajiang"` and `to = "mjai"`: A character vector.
- For `to = "tenhou_int"`: An integer vector.

Examples

```
trans_tile(c("m0", "p1", "z1", "_"), from = "cmajiang", to = "mjai")
trans_tile(c("5mr", "1p", "E", "?"), from = "mjai", to = "cmajiang")
trans_tile(c(51, 21, 41, 0), from = "tenhou_int", to = "mjai")
```

Index

∗ **datasets**
    hupai, 7
    ranked_game, 14
    testdat_xiangting, 15
    tiles, 15

calc_defen, 2
calc_xiangting, 3
calculate.skksph_paistr *(*paistr*)*, 10
collect_tingpai, 4
convert-from-mjai, 5

default_rule, 6
detect-tiles, 6

hand2img, 7
hupai, 7

int2tile, 8
is_fulou *(*detect-tiles*)*, 6
is_lizhi *(*detect-tiles*)*, 6
is_paistr *(*paistr*)*, 10
is_suit *(*detect-tiles*)*, 6
is_suzhi *(*detect-tiles*)*, 6
is_tsumogiri *(*detect-tiles*)*, 6
is_valid_tile, 8
is_yaojiu *(*detect-tiles*)*, 6
is_zhongzhang *(*detect-tiles*)*, 6

lineup, 9
lipai, 9

mjai_conv *(*convert-from-mjai*)*, 5
mjai_target *(*convert-from-mjai*)*, 5
modify_rule, 10

paistr, *7*, 10
parse_hupai, 12
plot.skksph_paistr *(*paistr*)*, 10
proceed, 12

rand_hands, 13
rand_qipai, 14
ranked_game, 14

testdat_xiangting, 15
tidy.skksph_paistr *(*paistr*)*, 10
tile2rank *(*detect-tiles*)*, 6
tile2suit *(*detect-tiles*)*, 6
tiles, 15
trans_tile, 16