

# Package ‘isocat’

July 3, 2020

**Type** Package

**Title** Isotope Origin Clustering and Assignment Tools

**Version** 0.2.6

**Description** This resource provides tools to create, compare, and post-process spatial isotope assignment models of animal origin. It generates probability-of-origin maps for individuals based on user-provided tissue and environment isotope values (e.g., as generated by IsoMAP, Bowen et al. [2013] <doi:10.1111/2041-210X.12147>) using the framework established in Bowen et al. (2010) <doi:10.1146/annurev-earth-040809-152429>). The package 'isocat' can then quantitatively compare and cluster these maps to group individuals by similar origin. It also includes techniques for applying four approaches (cumulative sum, odds ratio, quantile only, and quantile simulation) with which users can summarize geographic origins and probable distance traveled by individuals. Campbell et al. [2020] establishes several of the functions included in this package <doi:10.1515/ami-2020-0004>.

**License** CC0

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**RoxygenNote** 7.1.1

**Depends** R (>= 2.10), utils, raster

**Imports** stats, plyr, sp, magrittr, foreach

**Suggests** dendextend, doParallel, dplyr, ggplot2, gridExtra, kableExtra, knitr, parallel, pvclust, rmarkdown, rasterVis, viridisLite

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Caitlin Campbell [aut, cre] (<<https://orcid.org/0000-0002-8199-7775>>)

**Maintainer** Caitlin Campbell <[caitjcampbell@gmail.com](mailto:caitjcampbell@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-07-03 16:30:12 UTC

**R topics documented:**

clusterSimmatrix	2
cumsumAtSamplingLocation	3
cumsumbelow	4
getPrecisionPar	5
isoscape	5
isoscape_sd	6
isotopeAssignmentModel	7
makecumsumSurface	8
makeOddsSurfaces	10
makeQuantileSimulationSurface	11
makeQuantileSurfaces	12
meanAggregateClusterProbability	13
oddsAtSamplingLocation	14
projectSummaryMaxSurface	15
quantileAtSamplingLocation	16
schoenersD	17
simmatrixMaker	18

<b>Index</b>	<b>20</b>
--------------	-----------

---

clusterSimmatrix	<i>Hierarchical clustering analysis of similarity matrix</i>
------------------	--

---

**Description**

Function applies hierarchical clustering analysis to similarity matrix, such as one output by 'simmatrixMaker' function. Just a wrapper for pvclust. Output is a pvclust object.

**Usage**

```
clusterSimmatrix(
  simmatrix,
  dist_mthd = "correlation",
  hclust_mthd = "average",
  nBoot = 1000,
  nClusters = FALSE,
  r = seq(0.7, 1.4, by = 0.1)
)
```

**Arguments**

simmatrix	symmetric similarity matrix object.
dist_mthd	Distance measure to be used. Defaults to "correlation". See help(pvclust).
hclust_mthd	Method of clustering. Defaults to "average". See help(pvclust).
nBoot	number of bootstrap replications. Defaults to 1000. See help(pvclust).

`nClusters`      number of clusters to run in parallel using 'doParallel'. Defaults to FALSE (non-parallel).

`r`                Relative size of bootstrap replications.

## Examples

```
# Create probability-of-origin maps to compare.
myiso <- rasterFromXYZ(isoscape)
raster::plot(myiso)
myiso_sd <- rasterFromXYZ(isoscape_sd)
n <- 5
set.seed(42)
df <- data.frame(
  ID = LETTERS[1:n],
  isotopeValue = sample(-120:-40, n),
  SD_indv = rep(5, n)
)
assignmentModels <- isotopeAssignmentModel(
  ID = df$ID,
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)
raster::plot(assignmentModels)
# Compare maps with simmatrixMaker.
mymatrix <- simmatrixMaker(assignmentModels, nClusters = FALSE, csvSavePath = FALSE)
# Cluster similarity matrix.
clust_results <- clusterSimmatrix(mymatrix, dist_mthd = "correlation",
  hclust_mthd = "average", nBoot = 1000, nClusters = FALSE,
  r = seq(.7,1.4,by=.1) )
clust_results
```

---

cumsumAtSamplingLocation

*Cumulative sum at coordinates*

---

## Description

Function estimates cumulative sum of all values in a surface below the value at a specified longitude and latitude.

## Usage

```
cumsumAtSamplingLocation(indivraster, Lat, Lon)
```

**Arguments**

indivraster	RasterLayer representing normalized probability of origin surface
Lat	Integer latitude
Lon	Integer longitude

**See Also**

[makecumsumSurface](#)

**Examples**

```
# Generate example probability surface.
myiso <- raster::rasterFromXYZ(isoscape)
myiso_sd <- rasterFromXYZ(isoscape_sd)
exampleSurface <- isotopeAssignmentModel(
  ID = "A",
  isotopeValue = -100,
  SD_indv = 5,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)
# Calculate odds ratio at specific point.
set.seed(1)
x <- sample( which( !is.na(exampleSurface[]) ), size = 1)
pt <- raster::xyFromCell(exampleSurface, x)
cumsumAtSamplingLocation(indivraster = exampleSurface, Lat = pt[2], Lon = pt[1])
```

---

cumsumbelow

*Cumulative sum below value*

---

**Description**

Function that calculates the cumulative sum of values less than or equal to a given value.

**Usage**

```
cumsumbelow(vals)
```

**Arguments**

vals	Object of numeric or integer class.
------	-------------------------------------

**Value**

Returns list of values representing cumulative sum of ‘val’ values less than or equal to the input.

**Examples**

```
vals <- 1:10
cumsumbelow(vals)
```

---

getPrecisionPar	<i>Calculate model precision at given threshold values (in parallel).</i>
-----------------	---

---

**Description**

Function that counts cells (number and proportion) above given values.

**Usage**

```
getPrecisionPar(rasterstack, checkVals, method = FALSE, nCluster = 20)
```

**Arguments**

rasterstack	RasterStack of probability surfaces
checkVals	vector of numeric 'threshold' values against which to calculate precision
method	is FALSE by default. If character vector, appends a column recording 'method' used.
nCluster	is a numeric object specifying how many clusters to form and run in parallel.

**Value**

Returns a dataframe of precision values at given threshold.

---

isoscape	<i>Example isoscape data</i>
----------	------------------------------

---

**Description**

Example isoscape data

**Usage**

```
isoscape
```

**Format**

A dataframe containing 1800 observations, derived from a cropped raster object.

## References

Campbell C. J. (2018) IsoMAP job 66100, Isoscapes Modeling, Analysis and Prediction (version 1.0). The IsoMAP Project. <http://isomap.org>

Bowen G. J., West J.B., Miller C. C., Zhao L. and Zhang T. (2018) IsoMAP: Isoscapes Modeling, Analysis and Prediction (version 1.0). IsoMAP job 66100, Caitlin J. Campbell. Isoscapes Modeling, Analysis and Prediction (version 1.0). The IsoMAP Project. <http://isomap.org>

## Examples

```
iso <- rasterFromXYZ(isoscape)
```

---

isoscape\_sd

*Example isoscape standard deviation data*

---

## Description

Example isoscape standard deviation data

## Usage

```
isoscape_sd
```

## Format

A dataframe containing 1800 observations, derived from a cropped raster object.

## References

Campbell C. J. (2018) IsoMAP job 66100, Isoscapes Modeling, Analysis and Prediction (version 1.0). The IsoMAP Project. <http://isomap.org>

Bowen G. J., West J.B., Miller C. C., Zhao L. and Zhang T. (2018) IsoMAP: Isoscapes Modeling, Analysis and Prediction (version 1.0). IsoMAP job 66100, Caitlin J. Campbell. Isoscapes Modeling, Analysis and Prediction (version 1.0). The IsoMAP Project. <http://isomap.org>

## Examples

```
iso_sd <- rasterFromXYZ(isoscape_sd)
```

---

 isotopeAssignmentModel

*Isotope assignment model function*


---

### Description

Creates isotope assignment models projections of probable origin. Results returned as a Raster-Stack, with layer names corresponding to individual ID.

### Usage

```
isotopeAssignmentModel(
  ID,
  isotopeValue,
  SD_indv = 0,
  precip_raster,
  precip_SD_raster,
  additionalModels = FALSE,
  additionalModel_name = "CombinedIsotope-OtherModelAssignments",
  savePath = FALSE,
  nClusters = FALSE
)
```

### Arguments

ID	ID value or vector of values (for naming assignment model layers). If missing, will count from 1.
isotopeValue	Isotope precipitation value or vector of values.
SD_indv	error associated with transfer function fit. Value or vector of values. If missing, will assume value of 0.
precip_raster	precipitation isoscape raster.
precip_SD_raster	precipitation isoscape standard deviation raster.
additionalModels	optional additional model raster object (e.g. an SDM, rasterized range map, or stack thereof). If specified, function will return isotope assignment rasters and the product of these additionalModels and each assignmentRaster.
additionalModel_name	optional filename for additionalModel .grd savepath
savePath	If specified, function will save results to this path as a '.grd' file.
nClusters	integer of cores to run in parallel with doParallel. Default FALSE.

## Examples

```

myiso <- rasterFromXYZ(isoscape)
raster::plot(myiso)
myiso_sd <- rasterFromXYZ(isoscape_sd)
df <- data.frame(
  ID = paste0("Example.", 1:3),
  isotopeValue = c(-100, -80, -50),
  SD_indv = rep(5, 3)
)
assignmentModels <- isotopeAssignmentModel(
  ID = df$ID,
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd
)
raster::plot(assignmentModels)

# Add additionalModels:
range_raster <- myiso
range_raster[] <- as.numeric( 1:ncell(myiso) %% 60 >= 10)
plot(range_raster)

sdm_raster <- myiso
sdm_raster[] <- (1:ncell(sdm_raster))^2
sdm_raster <- sdm_raster / raster::cellStats(sdm_raster, "sum")
plot(sdm_raster)

extraModels <- raster::stack(range_raster, sdm_raster)
assignmentModels <- isotopeAssignmentModel(
  ID = paste0("Combo.",df$ID),
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  additionalModels = extraModels
)
raster::plot(assignmentModels)

```

---

makecumsumSurface

*Create cumulative sum probability surface*

---

## Description

Converts normalized probability surface (e.g. one layer output of `isotopeAssignmentModel` function) to cumulative sum surfaces, i.e., one where the new value of a given cell is equal to the sum of all old values less than or equal to the old value of the cell.



**Usage**

```
makecumsumSurface(indivraster, rescale = FALSE, rename = FALSE)
```

**Arguments**

indivraster	Normalized probability surface RasterLayer
rescale	Rescale between 0 and 1? Defaults to FALSE.
rename	Character value to append to raster name (e.g. "_odds"). Defaults to FALSE.

**Value**

Returns RasterLayer rescaled to Cumulative Sum values.

**See Also**

[cumsumAtSamplingLocation](#)

**Examples**

```
# Generate example probability surfaces.
myiso <- rasterFromXYZ(isoscape)
myiso_sd <- rasterFromXYZ(isoscape_sd)
df <- data.frame(
  ID = c(-100, -80, -50),
  isotopeValue = c(-100, -80, -50),
  SD_indv = rep(5, 3)
)
assignmentModels <- isotopeAssignmentModel(
  ID = df$ID,
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)

# Convert to cumulative sum surface.
cumulative_sum_surface <- stack(
  lapply( unstack( assignmentModels ), makecumsumSurface )
)
plot(cumulative_sum_surface)
```

---

makeOddsSurfaces	<i>Convert probability surface to odds-ratio surface</i>
------------------	--

---

### Description

Converts normalized probability surface (e.g. one layer output of `isotopeAssignmentModel` function) to Odds Ratio surfaces.

### Usage

```
makeOddsSurfaces(probabilitySurface, rename = FALSE)
```

### Arguments

probabilitySurface	Normalized probability surface RasterLayer
rename	Character value to append to raster name (e.g. "_odds"). Defaults to FALSE.

### Value

Returns RasterLayer rescaled to Odds Ratio values.

### See Also

[oddsAtSamplingLocation](#)

### Examples

```
# Generate example probability surfaces.
myiso <- rasterFromXYZ(isoscape)
myiso_sd <- rasterFromXYZ(isoscape_sd)
df <- data.frame(
  ID = c(-100, -80, -50),
  isotopeValue = c(-100, -80, -50),
  SD_indv = rep(5, 3)
)
assignmentModels <- isotopeAssignmentModel(
  ID = df$ID,
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)

# Convert to odds ratio surfaces.
odds_ratio_surface <- stack(
  lapply( unstack(assignmentModels), makeOddsSurfaces )
)
```

```
plot(odds_ratio_surface)
```

---

```
makeQuantileSimulationSurface
```

*Create quantile-simulation surfaces*

---

## Description

Converts normalized probability surfaces (e.g. one layer output of `isotopeAssignmentModel` function) to quantile surfaces.

## Usage

```
makeQuantileSimulationSurface(  
  probabilitySurface,  
  ValidationQuantiles,  
  rename = FALSE,  
  rescale = TRUE  
)
```

## Arguments

<code>probabilitySurface</code>	Normalized probability surface <code>RasterLayer</code> .
<code>ValidationQuantiles</code>	Vector of quantile values from known-origin individuals, against which to compare each value within the probability surface. Each value must be between 0 and 1.
<code>rename</code>	Character value to append to raster name (e.g. <code>"_quantileSimulation"</code> ). Defaults to <code>FALSE</code> .
<code>rescale</code>	If <code>rescale = TRUE</code> , returns surface showing proportion of times each surface cell value fell within the validation quantiles distribution. If <code>rescale = FALSE</code> , returns discrete number of times the cell fell within the distribution.

## Value

Returns `RasterLayer` rescaled to quantile values.

## Examples

```
# Generate example probability surfaces.  
library(isocat)  
myiso <- raster::rasterFromXYZ(isoscape)  
myiso_sd <- raster::rasterFromXYZ(isoscape_sd)  
df <- data.frame(  
  ID = c(-100, -80, -50),
```

```

        isotopeValue = c(-100, -80, -50),
        SD_indv = rep(5, 3)
    )
assignmentModels <- isotopeAssignmentModel(
  ID = df$ID,
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd
)

# Example known-origin quantile data.
q <- rweibull(20000, 6, .98)
q <- sample( q[ q >=0 & q <= 1 ], 10000, replace = TRUE)
hist(q)

# Convert to quantile surfaces.
quantileSimulation_surface <- raster::stack(
  lapply(
    unstack(assignmentModels),
    makeQuantileSimulationSurface,
    ValidationQuantiles = q
  )
)
plot(quantileSimulation_surface)

```

---

makeQuantileSurfaces *Convert probability surface to probability-quantile surface*

---

### Description

Converts normalized probability surface (e.g. one layer output of isotopeAssignmentModel function) to quantile surfaces.

### Usage

```
makeQuantileSurfaces(probabilitySurface, rename = FALSE)
```

### Arguments

probabilitySurface  
Normalized probability surface RasterLayer

rename  
Character value to append to raster name (e.g. "\_quantile"). Defaults to FALSE.

### Value

Returns RasterLayer rescaled to quantile values.

**See Also**

[quantileAtSamplingLocation](#)

**Examples**

```
# Generate example probability surfaces.
myiso <- rasterFromXYZ(isoscape)
myiso_sd <- rasterFromXYZ(isoscape_sd)
df <- data.frame(
  ID = c(-100, -80, -50),
  isotopeValue = c(-100, -80, -50),
  SD_indv = rep(5, 3)
)
assignmentModels <- isotopeAssignmentModel(
  ID = df$ID,
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)

# Convert to quantile surfaces.
quantile_surface <- raster::stack( lapply( unstack(assignmentModels), makeQuantileSurfaces) )
plot(quantile_surface)
```

---

meanAggregateClusterProbability

*Create mean aggregate probability-of-origin surfaces for each cluster.*

---

**Description**

Subset probability-of-origin surfaces by cluster assignment and find mean aggregate probability-of-origin surface for each clustered group.

**Usage**

```
meanAggregateClusterProbability(indivIDs, clusters, surfaces, nClust = FALSE)
```

**Arguments**

indivIDs	Vector of individual ID variables corresponding to surface names.
clusters	Vector of cluster IDs, in an order corresponding to 'indivIDs'.
surfaces	Stack of probability-of-origin surfaces for all individuals. Object of class 'RasterStack.'
nClust	Create and apply a multi-core cluster for faster processing using 'raster' and 'parallel' packages. Defaults to 'FALSE' (i.e., no clustering).

**Examples**

```

# Create and cluster example assignment surfaces.
myiso <- rasterFromXYZ(isoscape)
myiso_sd <- rasterFromXYZ(isoscape_sd)
df <- data.frame(
  ID = LETTERS[1:9],
  isotopeValue = seq(-120,-25,length.out = 9),
  SD_indv = rep(5, 9)
)
assignmentModels <- isotopeAssignmentModel(
  ID = df$ID,
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)
mySimilarityMatrix <- simmatrixMaker(assignmentModels)
cS <- clusterSimmatrix(
  simmatrix = mySimilarityMatrix,
  r = seq(.7,1.4,by=.1)
)
# Cut clusters.
myheight <- 0.25
df$cluster <- dendextend::cutree(cS$hclust, h = myheight)
# Create mean aggregate surfaces.r p
meanSurfaces <- meanAggregateClusterProbability(
  indivIDs = df$ID,
  clusters = df$cluster,
  surfaces = assignmentModels,
  nClust = FALSE
)

```

---

oddsAtSamplingLocation

*Odds ratio at coordinates function*

---

**Description**

Function estimates percentile of each non-NA value within a RasterLayer using the empirical cumulative distribution function, and extracts value at location specified. For more information, see `help(ecdf)`.

**Usage**

```
oddsAtSamplingLocation(indivraster, Lat, Lon)
```

**Arguments**

indivraster	RasterLayer representing normalized probability of origin surface
Lat	Integer latitude
Lon	Integer longitude

**See Also**

[makeOddsSurfaces](#)

**Examples**

```
# Generate example probability surface.
myiso <- raster::rasterFromXYZ(isoscape)
myiso_sd <- rasterFromXYZ(isoscape_sd)
exampleSurface <- isotopeAssignmentModel(
  ID = "A",
  isotopeValue = -100,
  SD_indv = 5,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)
# Calculate odds ratio at specific point.
set.seed(1)
x <- sample( which( !is.na(exampleSurface[]) ), size = 1)
pt <- raster::xyFromCell(exampleSurface, x)
oddsAtSamplingLocation(exampleSurface, Lat = pt[2], Lon = pt[1])
```

---

projectSummaryMaxSurface

*Project probability-of-origin surfaces into one summary surface.*

---

**Description**

Create a summary surface showing which RasterLayer in a Stack has the highest value at a given location. For each cell in a RasterStack, this function returns the identity of the RasterLayer with the highest value at that cell. This surface is intended as a visual summary of common origins, not a basis for quantitative analysis.

**Usage**

```
projectSummaryMaxSurface(surfaces, nClust = FALSE)
```

**Arguments**

surfaces	Object of class "RasterStack", where each layer represents a probability-of-origin surface
nClust	Create and apply a multi-core cluster for faster processing using 'raster' and 'parallel' packages. Defaults to 'FALSE' (i.e., no clustering).

**Examples**

```
# Create and cluster example assignment surfaces.
myiso <- rasterFromXYZ(isoscape)
myiso_sd <- rasterFromXYZ(isoscape_sd)
assignmentModels <- isotopeAssignmentModel(
  ID = LETTERS[1:4],
  isotopeValue = seq(-120,-25,length.out = 4),
  SD_indv = rep(5,4),
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)
raster::plot(assignmentModels)

# Project mean aggregate surfaces into space.
summaryMap <- projectSummaryMaxSurface(
  surfaces = assignmentModels,
  nClust = FALSE
)
raster::plot(summaryMap)
```

---

quantileAtSamplingLocation

*Quantile at coordinates function*

---

**Description**

Function estimates percentile of each non-NA value within a RasterLayer using the empirical cumulative distribution function, and extracts value at location specified. For more information, see `help(ecdf)`.

**Usage**

```
quantileAtSamplingLocation(indivraster, Lat, Lon)
```

**Arguments**

indivraster	A RasterLayer representing normalized probability of origin surface
Lat	Integer latitude
Lon	Integer longitude



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

```
# Generate example probability surface.
myiso <- raster::rasterFromXYZ(isoscape)
myiso_sd <- rasterFromXYZ(isoscape_sd)
exampleSurface <- isotopeAssignmentModel(
  ID = "A",
  isotopeValue = -100,
  SD_indv = 5,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)
# Calculate quantile probability value at specific point.
set.seed(1)
x <- sample( which( !is.na(exampleSurface[]) ), size = 1)
pt <- raster::xyFromCell(exampleSurface, x)
quantileAtSamplingLocation(exampleSurface, Lat = pt[2], Lon = pt[1])
```

---

schoenersD

*Calculates Schoener's D-value between two RasterLayers.*

---

**Description**

RasterLayers must have identical resolutions and extents.

**Usage**

```
schoenersD(rast1, rast2)
```

**Arguments**

rast1	Input RasterLayer
rast2	Input RasterLayer 2

**Details**

Calculates similarity value of two RasterLayers using Schoener's D-metric.

## Examples

```
# Generate example probability surfaces.
myiso <- raster::rasterFromXYZ(isoscape)
myiso_sd <- raster::rasterFromXYZ(isoscape_sd)
df <- data.frame(
  ID = c(-100, -80, -50),
  isotopeValue = c(-100, -80, -50),
  SD_indv = rep(5, 3)
)
assignmentModels <- isotopeAssignmentModel(
  ID = df$ID,
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)

# Calculate Schoener's D-metric of spatial similarity between two of the
# example probability surfaces.
schoenersD(assignmentModels[[1]], assignmentModels[[2]])
## 0.969156
```

---

simmatrixMaker	<i>Generates similarity matrix for RasterStack</i>
----------------	--

---

## Description

Applies pairwise comparisons of Schoener's D-metric between each RasterLayer in a RasterStack to populate a similarity matrix.

## Usage

```
simmatrixMaker(assignmentRasters, nClusters = FALSE, csvSavePath = FALSE)
```

## Arguments

assignmentRasters	Input RasterStack
nClusters	Clusters to create run in parallel using 'doParallel'. Defaults to FALSE.
csvSavePath	Optional savepath to write similarity matrix to csv file. Defaults to FALSE, will not create csv.

**Examples**

```
# Create probability-of-origin maps to compare.
myiso <- rasterFromXYZ(isoscape)
raster::plot(myiso)
myiso_sd <- rasterFromXYZ(isoscape_sd)
n <- 5
set.seed(42)
df <- data.frame(
  ID = LETTERS[1:n],
  isotopeValue = sample(-120:-40, n),
  SD_indv = rep(5, n)
)
assignmentModels <- isotopeAssignmentModel(
  ID = df$ID,
  isotopeValue = df$isotopeValue,
  SD_indv = df$SD_indv,
  precip_raster = myiso,
  precip_SD_raster = myiso_sd,
  nClusters = FALSE
)
raster::plot(assignmentModels)
# Compare maps with simmatrixMaker.
simmatrixMaker(assignmentModels, nClusters = FALSE, csvSavePath = FALSE)
```

# Index

## \* datasets

isoscapes, 5  
isoscapes\_sd, 6

clusterSimmatrix, 2  
cumsum\_at\_point  
    (cumsumAtSamplingLocation), 3  
cumsum\_surface (makecumsumSurface), 8  
cumsumAtSamplingLocation, 3, 9  
cumsumbelow, 4

getPrecisionPar, 5

isoscapes, 5  
isoscapes\_sd, 6  
isotopeAssignmentModel, 7

makecumsumSurface, 4, 8  
makeOddsSurfaces, 10, 15  
makeQuantileSimulationSurface, 11  
makeQuantileSurfaces, 12, 17  
meanAggregateClusterProbability, 13

odds\_at\_point (oddsAtSamplingLocation),  
    14  
odds\_surface (makeOddsSurfaces), 10  
oddsAtSamplingLocation, 10, 14

projectSummaryMaxSurface, 15

quantile\_at\_point  
    (quantileAtSamplingLocation),  
    16  
quantile\_surface  
    (makeQuantileSurfaces), 12  
quantileAtSamplingLocation, 13, 16

schoenersD, 17  
simmatrixMaker, 18