

# Package ‘SillyPutty’

May 7, 2026

**Version** 0.4.2

**Date** 2025-04-07

**Title** Silly Putty Clustering

**Description** Implements a simple, novel clustering algorithm based on optimizing the silhouette width. See <[doi:10.1101/2023.11.07.566055](https://doi.org/10.1101/2023.11.07.566055)> for details.

**Depends** R (>= 4.4)

**Imports** methods, cluster, Thresher, oompaBase, Polychrome (>= 1.2)

**Suggests** knitr, rmarkdown, Mercator, Umpire, mclust

**VignetteBuilder** knitr

**License** Apache License (== 2.0)

**biocViews** Clustering

**URL** <http://oompa.r-forge.r-project.org/>

**NeedsCompilation** no

**Author** Dwayne Tally [aut],  
Kevin R. Coombes [aut, cre]

**Maintainer** Kevin R. Coombes <[krc@silicovore.com](mailto:krc@silicovore.com)>

**Repository** CRAN

**Date/Publication** 2025-04-08 12:40:06 UTC

## Contents

euclust . . . . .	2
findClusterNumber . . . . .	2
HCSP . . . . .	4
RandomSillyPutty-class . . . . .	5
SillyPutty-class . . . . .	6

<b>Index</b>	<b>9</b>
--------------	----------

---

`euclust`*An example Euclidean distance matrix*

---

**Description**

The Euclidean distance matrix between 300150 objects, used to illustrate the SillyPutty algorithms.

**Usage**

```
data(euclust)
```

**Format**

The binary R data file contains two objects. First, a `dist` object representing Euclidean distances between 150 samples. Second, a vector of the known (simulated) true groups to which each sample belongs.

**Details**

This data set was generated in the SillyPutty vignette from tools in the Umpire R package. The simulated data was intended to have five different clusters, all of approximately the same size. Noise was added to make the clusters somewhat harder for most algorithms to distinguish. The same data set is used in most of the examples in the man pages.

**Source**

This data set was generated in the SillyPutty vignette from the tools in the Umpire R package, and saved using code that is now hidden and disabled in the vignette source.

**Examples**

```
data(euclust)
class(euclust)
attr(euclust, "Size")
```

---

`findClusterNumber`*Using SillyPutty to find the number of clusters*

---

**Description**

A function that is designed to find an approximation of the true number,  $K$ , of clusters in a dataset. The `findClusterNumber` function calls `RandomSillyPutty` for each value of  $K$  in the range from `start` to `end`, performing `N` random starts each time.

NOTE: `start` must be  $> 1$ , and the function can be slow depending on how complex the dataset is and the number of `N` iterations.

**Usage**

```
findClusterNumber(distobj, start,end, N = 100,  
                  method = c("SillyPutty", "HCSP"), ...)
```

**Arguments**

<code>distobj</code>	An object of class <code>dist</code> representing a distance matrix.
<code>start</code>	The minimum cluster number for the range of clusters
<code>end</code>	The maximum cluster number for the range of clusters
<code>N</code>	Number of iterations
<code>method</code>	whether to use the full <code>RandomSillyPutty</code> algorithm or use the hybrid method of hierarchical clustering followed by <code>SillyPutty</code> .
<code>...</code>	Extra arguments to the <code>SillyPutty</code> function.

**Details**

The `findClusterNumber` function processes one distance matrix at a time, through `N` iterations. It returns a list. The `list` is a list of the maximum silhouette width values obtained from `N` iterations with their associated cluster number.

**Value**

A list containing the maximum silhouette width values per `K` clusters for each `K` in the range of possible cluster numbers.

**Author(s)**

Kevin R. Coombes <krc@silicovore.com>, Dwayne G. Tally <dtally110@hotmail.com>

**References**

Pending.

**Examples**

```
data(eucdist)  
set.seed(12)  
y <- findClusterNumber(eucdist, start = 3, end = 7, method = "HCSP")  
plot(names(y), y, xlab = "K", ylab = "Mean Silhouette Width",  
      type = "b", lwd = 2, pch = 16)
```

**Description**

Our simulations revealed that the fastest and most accurate clustering algorithm for modest-sized continuous data sets is the combination of hierarchical clustering (with Ward's linkage rule) followed by SillyPutty. The function HCSP implements this combination.

**Usage**

```
HCSP(dis, K, method = "ward.D2", ...)
```

**Arguments**

<code>dis</code>	An object of class <code>dist</code> representing a distance matrix.
<code>K</code>	The desired number of clusters.
<code>method</code>	Sane as the corresponding argument for <code>hclust</code> . We recommend not changing it.
<code>...</code>	Extra arguments to the SillyPutty function.

**Details**

The HCSP function that first runs hierarchical clustering, then applies the SillyPutty algorithm.

**Value**

A list containing two items: `hc`, the results of hierarchical clustering, and `sp`, a SillyPutty object by applying the algorithm to the result of cutting the dendrogram to produce `K` clusters.

**Author(s)**

Kevin R. Coombes <krc@silicovore.com>

**References**

Polina Bombina, Dwayne Tally, Zachary B. Abrams, Kevin R. Coombes. SillyPutty: Improved clustering by optimizing the silhouette width, bioRxiv 2023.11.07.566055; doi: <https://doi.org/10.1101/2023.11.07.566055>

**Examples**

```
data(eucdist)
set.seed(1234)
twostep <- HCSP(eucdist, K=5)
sw <- cluster::silhouette(twostep$sp@cluster, eucdist)
plot(sw)
```

---

 RandomSillyPutty-class

*Running SillyPutty From Multiple Random Initial Clusterings*


---

## Description

A function to perform cluster assignments on a distance matrix based on optimizing silhouette width. The cluster assignments are based on maximum and minimum silhouette width scores obtained from N iterations.

## Usage

```
RandomSillyPutty(distobj, K, N = 100, verbose = FALSE, ...)
## S4 method for signature 'RandomSillyPutty,matrix'
plot(x, y, distobj, col = NULL, ...)
## S4 method for signature 'RandomSillyPutty,missing'
plot(x, y, ...)
## S4 method for signature 'RandomSillyPutty'
summary(object, ...)
## S4 method for signature 'RSPSummary,missing'
plot(x, y, ...)
```

## Arguments

distobj	An object of class <code>dist</code> .
K	The number of clusters.
N	The number of iterations you want to run.
verbose	A logical value; should you print info while working
...	Extra arguments to the SillyPutty function or to generic methods.
x	An object of the RandomSillyPutty or RSPSummary class.
object	An object of the RandomSillyPutty class.
y	A layout matrix.
col	A character vector containing color names.

## Details

The RandomSillyPutty function reads and processes one distance matrix at a time, with a precomputed cluster number, and a number N iterations. RandomSillyPutty returns an s4 object. The MX component of this structure contains an integer vector that has a cluster assignment based on the best scoring silhouette width score from N iterations. The MN contains an integer vector that has a cluster assignment based on the worst scoring silhouette score from N iterations. The ave contains the average silhouette width of all N iteration. The labels is a dataframe containing the cluster assignment of the best scoring silhouette width score per iteration. The minItemSW is a list containing the silhouette width score of all N iterations.

**Value**

The constructor function, `RandomSillyPutty`, returns an object of the `RandomSillyPutty` class.

**Slots**

**MX:** An integer vector containing cluster assignment that had the best silhouette width from running the iterations

**MN:** An integer vector containing cluster assignment that had the worst silhouette width from running the iterations

**ave:** An integer vector of average distribution of the silhouette width scores throughout N iterations

**labels:** A data frame of the cluster assignments of the best silhouette width score.

**minItemSW:** A list of silhouette width scores of all N iterations

**Methods**

**plot** signature(x = "RandomSillyPutty", y = "matrix"): Plot the clusterings with the maximum and minimum global silhouette widths.

**summary** signature(x = "RandomSillyPutty"): .

**Author(s)**

Kevin R. Coombes <krc@silicovore.com>, Dwayne G. Tally <dtally110@hotmail.com>

**References**

Pending.

**Examples**

```
data(eucdist)
# 'eucdist' is the Euclidean distance matrix (i.e., 'dist' object) from
# a simulated data set with 500 elements and 5 groups.
set.seed(12)
y <- RandomSillyPutty(eucdist, 6, N = 100)
summary(y)
```

---

SillyPutty-class

*Running SillyPutty*

---

**Description**

A function that takes in an already existing starting location based on unsupervised clustering attempts. I.G. Kmeans or Hierarchical cluster assignment. SillyPutty optimizes the pre-existing cluster assignments based on the best silhouette width score.

**Usage**

```
SillyPutty(labels, dissim, maxIter = 1000, loopSize = 15, verbose = FALSE)
```

**Arguments**

labels	A numeric vector containing pre-computed cluster labels
dissim	An object of class <code>dist</code> ; that is, a distance matrix.
maxIter	A numeric vector of length one; the maximum number of individual steps, each of which reclassifies only one object
loopSize	How many steps to retain in memory to test if you have entered an infinite loop while rearranging objects.
verbose	A logical vector of length one; should you output a lot of information while running?

**Details**

The `SillyPutty` function processes a pre-computed cluster assignment along with a distance metric and returns a `s4` object. The `cluster` component is a list of the new cluster assignments based on best silhouette width score. The `silhouette` is a dataframe containing the silhouette width score calculated by `SillyPutty`. The `minSW` contains a positive and negative version of the minimum silhouette width score. The `meanSW` is a double vector that shows the average silhouette width score after applying `SillyPutty` to the cluster assignment.

**Value**

The constructor function `SillyPutty`, returns an object of the `SillyPutty` class.

**Slots**

`cluster`: A list containing the adjusted cluster assignment that had the best silhouette width.

`silhouette`: A dataframe containing the silhouette width scores.

`minSW`: A silhouette double vector that contains the positive and negative version of the minimum silhouette width value.

`meanSW`: A double vector that contains the average silhouette width value.

**Author(s)**

Kevin R. Coombes <krc@silicovore.com>, Dwayne G. Tally <dtally11@hotmail.com>

**References**

Pending

**Examples**

```
data(eucdist)
set.seed(12)
hc <- hclust(eucdist, "ward.D2")
clues <- cutree(hc, k = 5)
hcSilly <- SillyPutty(clues, eucdist)
```

# Index

- \* **cluster**
  - findClusterNumber, 2
  - HCSP, 4
  - RandomSillyPutty-class, 5
  - SillyPutty-class, 6
- \* **datasets**
  - euclust, 2
- \*
  - findClusterNumber, 2
  - HCSP, 4
  - RandomSillyPutty-class, 5
  - SillyPutty-class, 6
- dist, 5
- euclust, 2
- findClusterNumber, 2
- hclust, 4
- HCSP, 4
- plot, RandomSillyPutty, matrix-method  
(RandomSillyPutty-class), 5
- plot, RandomSillyPutty, missing-method  
(RandomSillyPutty-class), 5
- plot, RSPSummary, matrix-method  
(RandomSillyPutty-class), 5
- plot, RSPSummary, missing-method  
(RandomSillyPutty-class), 5
- RandomSillyPutty  
(RandomSillyPutty-class), 5
- RandomSillyPutty-class, 5
- SillyPutty (SillyPutty-class), 6
- SillyPutty-class, 6
- summary, RandomSillyPutty-method  
(RandomSillyPutty-class), 5
- trueGroups (euclust), 2