

# Package ‘caviarpd’

March 8, 2026

**Type** Package

**Title** Cluster Analysis via Random Partition Distributions

**Version** 0.3.22

**Description** Cluster analysis is performed using pairwise distance information and a random partition distribution. The method is implemented for two random partition distributions. It draws samples and then obtains and plots clustering estimates.

An implementation of a selection algorithm is provided for the mass parameter of the partition distribution. Since pairwise distances are the principal input to this procedure, it is most comparable to the hierarchical and k-medoids clustering methods. The method is Dahl, Andros, Carter (2022+) <[doi:10.1002/sam.11602](https://doi.org/10.1002/sam.11602)>.

**License** MIT + file LICENSE | Apache License 2.0

**URL** <https://github.com/dbdahl/caviarpd-package>

**BugReports** <https://github.com/dbdahl/caviarpd-package/issues>

**Depends** R (>= 4.3.0)

**Suggests** salso (>= 0.3.0)

**SystemRequirements** Cargo (Rust's package manager), rustc

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Config/Roxido/TemplateVersion** 26.03.04

**Config/Roxido/MinCLIVersion** 0.8.0

**NeedsCompilation** yes

**Author** David B. Dahl [aut, cre] (ORCID:

<<https://orcid.org/0000-0002-8173-1547>>),

R. Jacob Andros [aut] (ORCID: <<https://orcid.org/0000-0002-1289-385X>>),

J. Brandon Carter [aut] (ORCID:

<<https://orcid.org/0000-0003-1687-0564>>),

Authors of the dependency Rust crates [ctb] (see inst/AUTHORS file for details)

**Maintainer** David B. Dahl <dahl@stat.byu.edu>

**Repository** CRAN

**Date/Publication** 2026-03-08 04:50:02 UTC

## Contents

caviarpd . . . . .	2
<b>Index</b>	<b>4</b>

---

caviarpd	<i>Cluster Analysis via Random Partition Distributions</i>
----------	--

---

## Description

Returns a clustering estimate given pairwise distances using the CaviarPD method.

## Usage

```
caviarpd(
  distance,
  nClusters,
  mass = NULL,
  nSamples = 200,
  gridLength = 5,
  loss = "binder",
  temperature = 100,
  similarity = c("exponential", "reciprocal")[1],
  maxNClusters = 0,
  nRuns = 4,
  nCores = nRuns
)
```

## Arguments

distance	An object of class 'dist' or a pairwise distance matrix.
nClusters	A numeric vector that specifies the range for the number of clusters to consider in the search for a clustering estimate.
mass	The mass value to use for sampling. If NULL, the mass value is found by inverting values from nClusters.
nSamples	The number of samples drawn per candidate estimate.
gridLength	The number of candidate estimates to consider. The final estimate is obtained from nSamples × gridLength total samples.
loss	The SALSO method (Dahl, Johnson, Müller, 2021) tries to minimize this expected loss when searching the partition space for an optimal estimate. This must be either "binder" or "VI".

temperature	A positive number that accentuates or dampens distance between observations.
similarity	Either "exponential" or "reciprocal" to indicate the desired similarity function.
maxNClusters	The maximum number of clusters that can be considered by the SALSO method.
nRuns	The number of runs of the SALSO algorithm.
nCores	The number of CPU cores to use. A value of zero indicates to use all cores on the system.

### Details

A range for the number of clusters to be considered is supplied using the `nClusters` argument.

### Value

A object of class `salso.estimate`, which provides a clustering estimate (a vector of cluster labels) that can be displayed and plotted.

### References

D. B. Dahl, J. Andros, J. B. Carter (2023), Cluster Analysis via Random Partition Distributions, *Statistical Analysis and Data Mining*, doi:[10.1002/sam.11602](https://doi.org/10.1002/sam.11602).

D. B. Dahl, D. J. Johnson, and P. Müller (2022), Search Algorithms and Loss Functions for Bayesian Clustering, *Journal of Computational and Graphical Statistics*, 31(4), 1189-1201, doi:[10.1080/10618600.2022.2069779](https://doi.org/10.1080/10618600.2022.2069779).

### Examples

```
# To reduce load on CRAN servers, limit the number of samples, grid length, and CPU cores.
set.seed(34)
iris.dis <- dist(iris[,-5])
est <- caviarpd(distance=iris.dis, nClusters=c(2,4), nSamples=20, nCores=1)
if ( require("salso") ) {
  summ <- summary(est, orderingMethod=2)
  plot(summ, type="heatmap")
  plot(summ, type="mds")
}
```

# Index

caviarpd, [2](#)