

# Package ‘inphr’

September 1, 2025

**Title** Statistical Inference for Persistence Homology Data

**Version** 0.0.1

**Description** A set of functions for performing null hypothesis testing on samples of persistence diagrams using the theory of permutations. Currently, only two-sample testing is implemented. Inputs can be either samples of persistence diagrams themselves or vectorizations. In the former case, they are embedded in a metric space using either the Bottleneck or Wasserstein distance. In the former case, persistence data becomes functional data and inference is performed using tools available in the 'fdatest' package. Main reference for the interval-wise testing method: Pini A., Vantini S. (2017) ``Interval-wise testing for functional data" <doi:10.1080/10485252.2017.1306627>. Main reference for inference on populations of networks: Lovato, I., Pini, A., Stamm, A., & Vantini, S. (2020) ``Model-free two-sample test for network-valued data" <doi:10.1016/j.csda.2019.106896>.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**URL** <https://github.com/tdaverse/inphr>,  
<https://tdaverse.github.io/inphr/>

**BugReports** <https://github.com/tdaverse/inphr/issues>

**Imports** cli, fdatest, flipr, phutil, rlang, TDAvec

**Depends** R (>= 3.5)

**LazyData** true

**Suggests** tinytest

**NeedsCompilation** no

**Author** Aymeric Stamm [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-8725-3654>>)

**Maintainer** Aymeric Stamm <aymeric.stamm@cnrs.fr>

**Repository** CRAN

**Date/Publication** 2025-09-01 09:50:07 UTC

## Contents

archspirals . . . . .	2
trefoils1 . . . . .	2
trefoils2 . . . . .	3
two_sample_diagram_test . . . . .	3
two_sample_functional_test . . . . .	5

<b>Index</b>	<b>8</b>
--------------	----------

---

archspirals	<i>Persistence diagrams from Archimedean spiral samples</i>
-------------	---

---

### Description

A set of 24 persistence diagrams computed from noisy samples of 2-armed Archimedean spirals. Each sample consists of 120 points sampled from an Archimedean spiral, embedded in 3D with a zero z-coordinate, then Gaussian noise (sd = 0.05) added. Vietoris-Rips persistence was computed up to dimension 2 with maximum scale 6 using `TDA::ripsDiag()`. Generated with seed 28415.

### Usage

```
archspirals
```

### Format

An object of class `persistence_set` containing 24 objects of class `phutil::persistence`.

---

trefoils1	<i>Persistence diagrams from trefoil knot samples (first set)</i>
-----------	---

---

### Description

A set of 24 persistence diagrams computed from noisy samples of trefoil knots. Each sample consists of 120 points sampled from a trefoil knot with Gaussian noise (sd = 0.05) added. Vietoris-Rips persistence was computed up to dimension 2 with maximum scale 6 using `TDA::ripsDiag()`. Generated with seed 28415.

### Usage

```
trefoils1
```

### Format

An object of class `persistence_set` containing 24 objects of class `phutil::persistence`.

---

trefoils2

*Persistence diagrams from trefoil knot samples (second set)*


---

### Description

A set of 24 persistence diagrams computed from noisy samples of trefoil knots. Each sample consists of 120 points sampled from a trefoil knot with Gaussian noise ( $sd = 0.05$ ) added. Vietoris-Rips persistence was computed up to dimension 2 with maximum scale 6 using `TDA::ripsDiag()`. Generated with seed 28415.

### Usage

```
trefoils2
```

### Format

An object of class `persistence_set` containing 24 objects of class `phutil::persistence`.

---

two\_sample\_diagram\_test

*Two-sample test for diagram representation of persistence homology data*


---

### Description

This function performs a two-sample test for persistence homology data using the theory of permutation hypothesis testing to test the null hypothesis that the two samples come from the same distribution. The inference is performed using test statistics that only involve distances between persistence diagrams. Hence, the input data can be either a persistence set or a precomputed distance matrix.

### Usage

```
two_sample_diagram_test(
  x,
  y,
  dimension = 0L,
  p = 2L,
  ncores = 1L,
  B = 1000L,
  stat_functions = list(flipr::stat_t_ip, flipr::stat_f_ip),
  npc = "tippett",
  seed = NULL,
  verbose = FALSE,
  keep_null_distribution = FALSE,
  keep_permutations = FALSE
)
```

**Arguments**

x	An object of class <code>persistence_set</code> typically produced by <code>phutil::as_persistence_set()</code> or of class <code>dist</code> typically produced by <code>phutil::bottleneck_pairwise_distances()</code> or <code>phutil::wasserstein_pairwise_distances()</code> . If <code>x</code> is a persistence set, then <code>y</code> must be either a vector of two integers (sample sizes) or another persistence set. If <code>x</code> is a distance matrix, then <code>y</code> must be a vector of two integers (sample sizes).
y	An object of class <code>persistence_set</code> typically produced by <code>phutil::as_persistence_set()</code> or a vector of two integers. If <code>x</code> is a persistence set, then <code>y</code> must be either a vector of two integers (sample sizes) or another persistence set. If <code>x</code> is a distance matrix, then <code>y</code> must be a vector of two integers (sample sizes).
dimension	An integer value specifying the homology dimension to use. Defaults to $0L$ , which corresponds to the 0-dimensional homology.
p	An integer value specifying the p-norm to use for the Wasserstein distance. Defaults to $2L$ , which corresponds to the Euclidean distance. If <code>p</code> is set to <code>Inf</code> , then the Bottleneck distance is used.
ncores	An integer value specifying the number of cores to use when computing the pairwise distance matrix between all combined persistence diagrams. Defaults to $1L$ , which means that the computation is done sequentially.
B	An integer value specifying the number of permutations to use for the permutation hypothesis test. Defaults to $1000L$ .
stat_functions	A list of functions that compute test statistics to be used for solving the inference problem. These functions must take two arguments: first, an object of class <code>dist</code> representing a distance matrix and second, an integer vector specifying the indices of the data points belonging to the first sample. Defaults to <code>list(flipr::stat_t_ip, flipr::stat_f_ip)</code> which are distance-based statistics equivalent to Student's and Fisher's statistics respectively.
npc	A string specifying the non-parametric combination method to use. Choices are either "tippett" (default) or "fisher". The former corresponds to the Tippett's method, while the latter corresponds to Fisher's method.
seed	An integer value specifying the seed for random number generation. Defaults to <code>NULL</code> which uses current time.
verbose	A boolean value indicating whether to print some information about the progress of the computation. Defaults to <code>FALSE</code> .
keep_null_distribution	A boolean specifying whether the empirical permutation null distribution should be returned as well. Defaults to <code>FALSE</code> .
keep_permutations	A boolean specifying whether the list of sampled permutations used to compute the empirical permutation null distribution should be returned as well. Defaults to <code>FALSE</code> .

**Value**

A numeric value storing the p-value from the two-sample test where the null hypothesis is that the two samples come from the same distribution. If one of `keep_null_distribution` or `keep_permutations`

is set to TRUE, then the output will be a list containing the p-value and the null distribution (if keep\_null\_distribution is set to TRUE) and the list of sampled permutations (if keep\_permutations is set to TRUE).

## References

Lovato, I., Pini, A., Stamm, A., & Vantini, S. (2020). Model-free two-sample test for network-valued data. *Computational Statistics & Data Analysis*, 144, 106896.

## Examples

```
two_sample_diagram_test(trefoils1[1:5], trefoils2[1:5], B = 100L)
two_sample_diagram_test(trefoils1[1:5], archspirals[1:5], B = 100L)
```

---

```
two_sample_functional_test
```

*Two-sample test for functional representations of persistence homology data*

---

## Description

This function performs a two-sample test for persistence homology data using the theory of permutation hypothesis testing to test the null hypothesis that the two samples come from the same distribution. The input data must be objects of class `persistence_set` typically produced by `phutil::as_persistence_set()`.

## Usage

```
two_sample_functional_test(
  x,
  y,
  dimension = 0L,
  scale_size = 100L,
  representation = c("betti", "euler", "life", "silhouette", "entropy"),
  mu = 0,
  order = 2L,
  nknots = scale_size,
  B = 1000L,
  paired = FALSE
)
```

## Arguments

- x An object of class `persistence_set` typically produced by `phutil::as_persistence_set()` specifying the first sample.
- y An object of class `persistence_set` typically produced by `phutil::as_persistence_set()` specifying the second sample.

dimension	An integer value specifying the homology dimension to use. Defaults to $0L$ , which corresponds to the 0-dimensional homology.
scale_size	An integer value specifying the number of scale values to use for the functional representation. Defaults to $100L$ .
representation	A string specifying the functional representation to use. Choices are "betti", "euler", "life", "silhouette", and "entropy". Defaults to "betti".
mu	The difference between the first functional population and the second functional population under the null hypothesis. Either a constant (in this case, a constant function is used) or a J-dimensional vector containing the evaluations on the same grid which data are evaluated. The default is $\mu=0$ .
order	Order of the B-spline basis expansion. The default is $\text{order}=2$ .
nknots	An integer value specifying the number of knots to use for the B-spline representation. Defaults to <code>scale_size</code> .
B	An integer value specifying the number of permutations to use for the permutation hypothesis test. Defaults to $1000L$ .
paired	A logical indicating whether the test is paired. The default is FALSE.

### Value

A length-4 list containing the following objects:

- `xfd`: A numeric matrix of shape  $n_1 \times p$  storing the representation of the first sample on a uniform grid.
- `yfd`: A numeric matrix of shape  $n_2 \times p$  storing the representation of the second sample on a uniform grid.
- `scale_seq`: A numeric vector of shape  $p$  storing the scale sequence used for the functional representation.
- `iwt`: An object of class ITP2 which is a list containing at least the following components:
  - `basis`: A string indicating the basis used for the first phase of the algorithm. In this case, equals to "B-spline".
  - `test`: A string indicating the type of test performed. In this case, equals to "2pop".
  - `mu`: The difference between the mean of the first and second populations under the null hypothesis (as entered by the user).
  - `paired`: A boolean value indicating whether the two samples are paired or not (as entered by the user).
  - `coeff`: A numeric matrix of shape  $n \times p$  of the  $p$  coefficients of the B-spline basis expansion, with  $n = n_1 + n_2$ . Rows are associated to units and columns to the basis index. The first  $n_1$  rows report the coefficients of the first population units and the following  $n_2$  rows report the coefficients of the second population units.
  - `pval`: A numeric vector of shape  $p$  storing the **uncorrected** p-values for each coefficient of the B-spline basis expansion.
  - `pval.matrix`: A numeric matrix of shape  $p \times p$  of the p-values of the multivariate tests. The element  $(i, j)$  of the `pval.matrix` matrix contains the p-value of the joint NPC test of the components  $(j, j + 1, \dots, j + (p - i))$ .

- `corrected.pval`: A numeric vector of shape  $p$  storing the **corrected** p-values for each coefficient of the B-spline basis expansion.
- `labels`: A character vector of shape  $n$  storing the membership of each unit to the first or second population.
- `data.eval`: A numeric matrix of shape  $n \times p$  storing the evaluation of the functional data on a uniform grid.
- `heatmap.matrix`: A numeric matrix storing the p-values. Used only for plots.

## References

Pini, A., & Vantini, S. (2017). Interval-wise testing for functional data. *Journal of Nonparametric Statistics*, 29(2), 407-424.

## Examples

```
tref1 <- trefoils1[1:5]
archsp <- archspirals[1:5]
out <- two_sample_functional_test(tref1, archsp, B = 10L, scale_size = 20L)
plot(out$wt, xrange = range(out$scale_seq))
matplot(
  out$scale_seq[-1],
  t(rbind(out$xfd, out$yfd)),
  type = "l",
  col = c(rep(1, length(tref1)), rep(2, length(archsp)))
)
```

# Index

## \* datasets

archspirals, 2

trefoils1, 2

trefoils2, 3

archspirals, 2

phutil::as\_persistence\_set(), 4, 5

phutil::bottleneck\_pairwise\_distances(),  
4

phutil::persistence, 2, 3

phutil::wasserstein\_pairwise\_distances(),  
4

TDA::ripsDiag(), 2, 3

trefoils1, 2

trefoils2, 3

two\_sample\_diagram\_test, 3

two\_sample\_functional\_test, 5