Package ‘riemstats’

November 17, 2025

Title Riemannian ANOVA Statistics

Version 0.2.0

Description Provides statistical methods for analyzing samples of symmetric positive definite (SPD) matrices, particularly functional connectivity matrices from neuroimaging data. Implements Fréchet ANOVA (Dubey and Müller (2019) <doi:10.1093/biomet/asz052>) for testing differences between groups in metric spaces, and Riemannian ANOVA methods that leverage tangent space geometry with classic multivariate test statistics including Wilks' Lambda and Pillai's trace. Also includes harmonization techniques for removing batch effects in multi-site studies: ComBat-based harmonization (Honnorat et al. (2024) <doi:10.1016/j.media.2023.103043>) and rigid harmonization (Simeon et al. (2022) <doi:10.3389/fninf.2022.769274>). Builds on 'riemtan' package infrastructure for efficient computation with multiple Riemannian metrics.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.3.0)

Imports Matrix, CovTools, sva, purrr, riemtan, methods

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

URL `https://nicoesve.github.io/riemstats/`

BugReports `https://github.com/nicoesve/riemstats/issues`

Config/testthat/edition 3

Maintainer Nicolas Escobar <`nescoba@iu.edu`>

NeedsCompilation no

Author Nicolas Escobar [aut, cre] (ORCID: <`https://orcid.org/0009-0006-0800-5692`>), Jaroslaw Harezlak [ths]

Repository CRAN

Date/Publication 2025-11-17 21:00:02 UTC

Contents

`combat_harmonization` *Harmonize Vector Images Using ComBat*

Description

Applies the ComBat batch effect correction to vector images in a `CSuperSample` object and reconstructs harmonized samples.

Usage

```
combat_harmonization(super_sample)
```

Arguments

`super_sample`	A `CSuperSample` object containing samples to harmonize.

Value

A new `CSuperSample` object with harmonized vector images.

`format_matr` *Format a Matrix as a Packed dpoMatrix*

Description

Converts a matrix to a packed symmetric positive definite matrix (`dpoMatrix`) using the Matrix package.

Usage

```
format_matr(x)
```

Arguments

`x`	A numeric matrix.

Value

A packed `dpoMatrix` object.

`frechet_anova` *Frechet ANOVA Test Statistic*

Description

Computes the Frechet ANOVA test statistic and its p-value for a given super-sample object.

Usage

```
frechet_anova(super_sample)
```

Arguments

`super_sample`	An object of class CSuperSample (see riemtan package)

Value

A list with two elements:

statistic The Frechet ANOVA test statistic

p_value The p-value from the chi-squared test

`log_wilks_lambda` *Compute the Log Wilks' Lambda Statistic*

Description

Calculates the log of Wilks' lambda statistic for a given `super_sample` object. This function ensures that the within-group and total covariance matrices are computed, then computes the difference of their log determinants.

Usage

```
log_wilks_lambda(super_sample)
```

Arguments

`super_sample`	An object of class CSuperSample

Details

Wilks' lambda is a test statistic for the ANOVA test decribed in (to appear)

Value

A numeric value representing the log Wilks' lambda statistic.

`normalization` *Normalize Rows of a Matrix*

Description

Centers and scales each row of a matrix to have zero mean and unit norm.

Usage

```
normalization(si)
```

Arguments

`si`	A numeric matrix.

Value

A matrix with each row centered and scaled.

`one_permutation` *Permutation Statistic for a Super Sample*

Description

Randomly shuffles all observations across groups while preserving group sample sizes, creating a new super sample under the null hypothesis of no group differences, and computes a specified statistic on the resulting permuted super sample.

Usage

```
one_permutation(x, stat_fun)
```

Arguments

`x`	An object of class `CSuperSample`, representing the original super sample.
`stat_fun`	A function to compute a statistic on the resulting `CSuperSample` object.

Details

This function performs a permutation test by:

1. Extracting all data points from all groups
2. Randomly shuffling the data
3. Reassigning data to groups with the same sample sizes as the original
4. Computing the test statistic on the permuted data

This approach tests the null hypothesis that group labels are exchangeable, which is natural for testing whether sub-populations differ.

Value

The value returned by `stat_fun` when applied to the permuted super sample.

`pillais_trace` *Compute Pillai's Trace Statistic*

Description

Calculates Pillai's trace statistic for a given `super_sample` object. This function ensures that the within-group and total covariance matrices are computed, then computes the sum of the eigenvalues of the matrix (Total - Within) %*% solve(Total).

Usage

```
pillais_trace(super_sample)
```

Arguments

`super_sample`	An object of class CSuperSample

Details

Pillai's trace is a test statistic for the ANOVA test decribed in (to appear).

Value

A numeric value representing Pillai's trace statistic.

`riem_anova` *Compute p-values using permutation test*

Description

Computes a permutation-based p-value for a given super sample. The statistic used for the permutation test can be specified via the `stat_fun` argument.

Usage

```
riem_anova(ss, stat_fun = log_wilks_lambda, nperm = 1000)
```

Arguments

`ss`	An object of class `CSuperSample`.
`stat_fun`	A function to compute a statistic on the `CSuperSample` object (default: `log_wilks_lambda`).
`nperm`	The number of permutations to generate for estimating the p-value (default: 1000).

Details

The function computes the statistic on the observed data and compares it to the distribution of statistics computed on permuted samples. Under the null hypothesis that group labels are exchangeable, this provides an exact test (subject to Monte Carlo error).

The permutation test:

1. Computes the test statistic on the observed data
2. Randomly shuffles group assignments while preserving sample sizes
3. Recomputes the test statistic on each permuted dataset
4. Calculates the p-value as the proportion of permuted statistics that exceed the observed statistic

This approach is computationally efficient and does not require parameter estimation or synthetic data generation.

Value

numeric A permutation-based p-value.

`rigid_harmonization` *Harmonize Tangent Images Across Batches Using Rigid Correction*

Description

Applies a rigid harmonization procedure to tangent images in a `CSuperSample` object. First, batch means are subtracted from each sample's tangent images (batch correction), then the overall mean is added back (global correction). The harmonized tangent images are used to reconstruct new `CSample` objects, which are collected into a new `CSuperSample`.

Usage

```
rigid_harmonization(super_sample)
```

Arguments

`super_sample`	A `CSuperSample` object containing samples to harmonize.

Details

This function performs harmonization in two steps:

1. **Batch Correction:** For each batch, the mean of its tangent images is subtracted from each tangent image in the batch.
2. **Global Correction:** The overall mean (across all batches) of tangent images is added back to each tangent image.

The harmonized tangent images are then used to reconstruct samples using the reference point and metric from the original `CSuperSample`.

Value

A new `CSuperSample` object with harmonized tangent images.

`ts2corr` *Compute OAS-Shrunk Correlation Matrix from Time Series*

Description

Normalizes a time series matrix and computes its OAS-shrunk correlation matrix.

Usage

```
ts2corr(ts)
```

Arguments

`ts`	A numeric matrix representing time series data.

Value

A shrunk correlation matrix.

Index