fasano.franceschini.test: An Implementation of a Multivariate KS Test in R

by Connor Puritz, Elan Ness-Cohn, and Rosemary Braun

Abstract The Kolmogorov–Smirnov (KS) test is a nonparametric statistical test used to test for differences between univariate probability distributions. The versatility of the KS test has made it a cornerstone of statistical analysis across many scientific disciplines. However, the test proposed by Kolmogorov and Smirnov does not easily extend to multivariate distributions. Here we present the fasano.franceschini.test package, an R implementation of a multivariate two-sample KS test described by Fasano and Franceschini (1987). The fasano.franceschini.test package provides a test that is computationally efficient, applicable to data of any dimension and type (continuous, discrete, or mixed), and that performs competitively with similar R packages.

1 Introduction

The Kolmogorov–Smirnov (KS) test is a nonparametric, univariate statistical test designed to assess whether a sample of data is consistent with a given probability distribution (or, in the two-sample case, whether the two samples came from the same underlying distribution). First described by Kolmogorov and Smirnov in a series of papers (Kolmogorov, 1933a,b; Smirnov, 1936, 1937, 1939, 1944, 1948), the KS test is a popular goodness-of-fit test that has found use across a wide variety of scientific disciplines, including neuroscience (Atasoy et al., 2017), climatology (Chiang et al., 2018), robotics (Hahne et al., 2018), epidemiology (Wong and Collins, 2020), and cell biology (Kaczanowska et al., 2021).

Due to its popularity, several multivariate extensions of the KS test have been described in literature. Justel et al. (1997) proposed a multivariate test based on Rosenblatt's transformation, which reduces to the KS test in the univariate case. While the test statistic is distribution-free, it is difficult to compute in more than two dimensions, and an approximate test with reduced power must be used instead. Furthermore, the test is only applicable in the one-sample case. Heuchenne and Mordant (2022) proposed to use the Hilbert space-filling curve to define an ordering in $\mathbb{R}^2$. The preimage of both samples is computed under the space-filling curve map, and the two-sample KS test is performed on the preimages. While it is theoretically possible to extend this approach to higher dimensions, the authors note that this would be computationally challenging and leave it as an open problem. Naaman (2021) derived a multivariate extension of the DKW inequality and used it to provide estimates of the tail properties of the asymptotic distribution of the KS test statistic in multiple dimensions. While an important theoretical result, it is of limited practical use absent a method for computing exact p-values.

Peacock (1983) proposed a test which addresses the fact that there are multiple ways to order points in higher dimensions, and thus multiple ways of defining a cumulative distribution function. In one dimension, probability density can be integrated from left to right, resulting in the canonical CDF $P(X < x)$; or from right to left, resulting in the survival function $P(X > x)$. However, since $P(X < x) = 1 - P(X > x)$ (for continuous random variables), the KS test statistic is independent of this choice. In two dimensions, there are four ways of ordering points, and thus four possible cumulative distribution functions: $P(X < x, Y < y)$, $P(X > x, Y < y)$, $P(X < x, Y > y)$, and $P(X > x, Y > y)$. Since any three of these are independent of one another, the KS test statistic will not be independent of which ordering is chosen. To address this, Peacock (1983) proposed to compute a KS statistic using each possible cumulative distribution function, and to take the test statistic to be the maximum of those.

Peacock (1983) suggested that for a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$, each of the four KS statistics should be maximized over the set of all coordinate-wise combinations $\{(X_i, Y_j) : 1 \leq i, j \leq n\}$. The complexity of computing Peacock's test statistic thus scales cubically with sample size, which is expensive and can become intractable for large sample sizes. Fasano and Franceschini (1987) proposed a simple change to Peacock's test: instead of maximizing each KS statistic over all coordinate-wise combinations of points in the sample, the statistics should be maximized over just the points in the sample itself. This slight change greatly reduces the computational complexity of the test while maintaining a similar power across a variety of alternatives (Fasano and Franceschini, 1987; Lopes et al., 2007). Fasano and Franceschini (1987) proposed both a one-sample and two-sample version of their test, although we focus on the two-sample test here.

In this article we present the **fasano.franceschini.test** package, an R implementation of the two-sample Fasano–Franceschini test. Our implementation can be applied to continuous, discrete, or

mixed datasets of any size and of any dimension. We first introduce the test by detailing how the test statistic is computed, how it can be computed efficiently, and how p-values can be computed. We then describe the package structure and provide several basic examples illustrating its usage. We conclude by comparing the package to three other CRAN packages implementing multivariate two-sample goodness-of-fit tests.

2 Fasano–Franceschini test

2.1 Two-sample test statistic

Let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_{n_1})$ and $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_2})$ be samples of i.i.d. d-dimensional random vectors drawn from unknown distributions F_1 and F_2, respectively. The two-sample Fasano–Franceschini test evaluates the null hypothesis

$$H_0 : F_1 = F_2$$

against the alternative

$$H_1 : F_1 \neq F_2.$$

In their original paper, Fasano and Franceschini (1987) only considered two- and three-dimensional random vectors, although their test naturally extends to arbitrary dimensions as follows.

For $\mathbf{x} \in \mathbb{R}^d$, we define the ith open orthant with origin $\mathbf{x}$ as

$$\mathcal{O}_i(\mathbf{x}) = \left\{ \mathbf{y} \in \mathbb{R}^d \,|\, \mathbf{e}_{ij}(\mathbf{y}_j - \mathbf{x}_j) > 0,\ j = 1, \ldots, d \right\}$$

where $\mathbf{e}_i \in \{-1, 1\}^d$ is a length d combination of ± 1. For example, in two dimensions, the four combinations $\mathbf{e}_1 = (1, 1)$, $\mathbf{e}_2 = (-1, 1)$, $\mathbf{e}_3 = (-1, -1)$, and $\mathbf{e}_4 = (1, -1)$ correspond to quadrants one through four in the plane, respectively. In general there are 2^d such combinations, corresponding to the 2^d orthants that divide $\mathbb{R}^d$. Using the indicator function

$$I_j(\mathbf{x} \,|\, \mathbf{y}) = \begin{cases} 1, & \mathbf{x} \in \mathcal{O}_j(\mathbf{y}) \\ 0, & \mathbf{x} \notin \mathcal{O}_j(\mathbf{y}) \end{cases}$$

we define

$$D(\mathbf{p} \,|\, \mathbf{X}, \mathbf{Y}) = \max_{1 \leq j \leq 2^d} \left| \frac{1}{n_1} \sum_{k=1}^{n_1} I_j\left(\mathbf{X}_k \,|\, \mathbf{p}\right) - \frac{1}{n_2} \sum_{k=1}^{n_2} I_j\left(\mathbf{Y}_k \,|\, \mathbf{p}\right) \right|. \tag{1}$$

This is similar to the distance used in the two-sample KS test, but takes into account all possible ways of ordering points in $\mathbb{R}^d$. Note that this function does not depend on the enumeration of the orthants. Maximizing D over each sample separately leads to the difference statistics

$$D_1(\mathbf{X}, \mathbf{Y}) = \max_{1 \leq i \leq n_1} D(\mathbf{X}_i \,|\, \mathbf{X}, \mathbf{Y})$$

and

$$D_2(\mathbf{X}, \mathbf{Y}) = \max_{1 \leq i \leq n_2} D(\mathbf{Y}_i \,|\, \mathbf{X}, \mathbf{Y}).$$

The two-sample Fasano–Franceschini test statistic, as originally defined by Fasano and Franceschini (1987), is the average of the difference statistics scaled by the sample sizes:

$$\mathcal{D}_0(\mathbf{X}, \mathbf{Y}) = \sqrt{\frac{n_1 n_2}{n_1 + n_2}} \left(\frac{D_1(\mathbf{X}, \mathbf{Y}) + D_2(\mathbf{X}, \mathbf{Y})}{2} \right). \tag{2}$$

This test statistic is discrete, but in general is not integer-valued. Note that

$$n_1 n_2 D(\mathbf{p} \,|\, \mathbf{X}, \mathbf{Y}) = \max_{1 \leq j \leq 2^d} \left| n_2 \sum_{k=1}^{n_1} I_j\left(\mathbf{X}_k \,|\, \mathbf{p}\right) - n_1 \sum_{k=1}^{n_2} I_j\left(\mathbf{Y}_k \,|\, \mathbf{p}\right) \right| \in \mathbb{Z},$$

and thus

$$n_1 n_2 D_i(\mathbf{X}, \mathbf{Y}) \in \mathbb{Z},\ i \in \{1, 2\}.$$

Let

$$C_{n_1, n_2} = 2\sqrt{n_1 n_2 (n_1 + n_2)}.$$

Then

$$C_{n_1,n_2}\mathcal{D}_0(\mathbf{X},\mathbf{Y}) = 2\sqrt{n_1 n_2 (n_1+n_2)}\sqrt{\frac{n_1 n_2}{n_1+n_2}}\left(\frac{D_1(\mathbf{X},\mathbf{Y})+D_2(\mathbf{X},\mathbf{Y})}{2}\right)$$
$$= n_1 n_2 (D_1(\mathbf{X},\mathbf{Y})+D_2(\mathbf{X},\mathbf{Y})) \in \mathbb{Z}.$$

To avoid comparing floating point numbers, it is preferable for the test statistic to be integer-valued, and thus we use

$$\mathcal{D}(\mathbf{X},\mathbf{Y}) = C_{n_1,n_2}\mathcal{D}_0(\mathbf{X},\mathbf{Y}) \tag{3}$$

as our test statistic. As will be shown, the p-value of the test is independent of scalar rescaling of the test statistic.

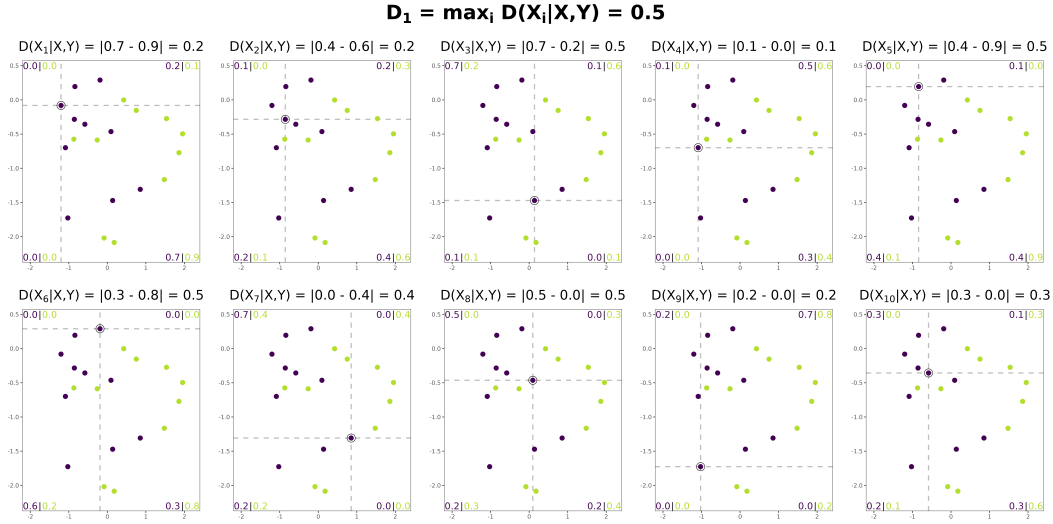

Figure 1: Illustration of the computation of the difference statistic D_1 in two dimensions. Each point in the first sample is used to divide the plane into four quadrants, and both samples are cumulated in each of the four quadrants. The fraction of each sample in each quadrant is shown in the corresponding plot corner, and the maximum difference over all four quadrants is shown above each plot. D_1 is taken as the maximum of these differences. To compute the test statistic, we would next compute D_2 by repeating the same procedure, but using points in the second sample to divide the plane instead.

2.2 Computational complexity

The bulk of the time required to compute the test statistic in (3) is spent evaluating sums of the form

$$\sum_{\mathbf{x}\in S} I_j(\mathbf{x}\,|\,\mathbf{y}),$$

which count the number of points in a set S that lie in a given d-dimensional region. The simplest approach to computing such sums is brute force, where every point $\mathbf{x} \in S$ is checked independently. The orthant a point lies in can be determined using d binary checks, resulting in a time complexity of $O(N^2)$, where $N = \max(n_1, n_2)$, to evaluate (3) for fixed d.

Alternatively, we can consider each sum as a single query rather than a sequence of independent ones. Specifically, both sums in (1) are orthogonal range counting queries, which ask how many points in a set $S \subset \mathbb{R}^d$ lie in an axis-aligned box $(x_1, x_1') \times \cdots \times (x_d, x_d')$. Range counting is an important problem in the field of computational geometry, and as such a variety of data structures have been described to provide efficient solutions (de Berg et al., 2008). One solution, first introduced by Bentley (1979), is a multi-layer binary search tree termed a range tree. Other slightly more efficient data structures have been proposed for range counting, but range trees are well suited for our purposes, particularly because their construction scales easily to arbitrary dimensions (Bentley, 1979; de Berg et al., 2008).

A range tree can be constructed on a set of n points in d-dimensional space using $O(n \log^{d-1} n)$ space in $O(n \log^{d-1} n)$ time. The number of points that lie in an axis-aligned box can be reported in $O(\log^d n)$ time, and this time can be further reduced to $O(\log^{d-1} n)$ when $d > 1$ using fractional cascading (de Berg et al., 2008). To compute (3), we construct one range tree for each of the two

samples, and then query each tree 2^d times. Thus the total time complexity to compute the test statistic using range trees for fixed d is $O(N \log^{d-1} N)$, where $N = \max(n_1, n_2)$.

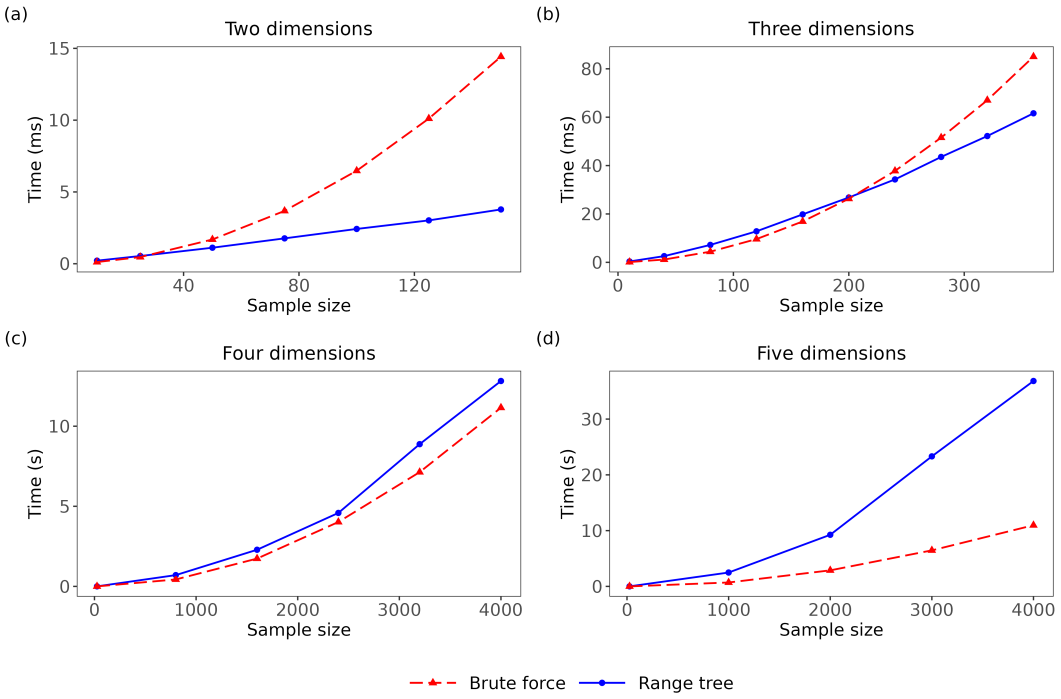

Figure 2: Time to compute the Fasano–Franceschini test statistic as a function of sample size, comparing the brute force and range tree methods for data of dimensions two through five. Points represent the mean time of 200 evaluations. Samples are taken to be of the same size and are drawn from multivariate standard normal distributions.

As the range tree method has a better asymptotic time complexity, we expect it to outperform the brute force method for larger sample sizes. However, for smaller sample sizes, the cost of building the range trees can outweigh the benefit gained by more efficient querying. As exact computation times can vary depending on the geometry of the samples, it is not possible to determine in general when one method will outperform the other. Despite this, we sought to establish rough benchmarks. Drawing equal sized samples from multivariate standard normal distributions, we sought to determine the sample size N^* at which the range tree method becomes more efficient than the brute force method (Figure 2). For $d = 2$, $N^* \approx 25$; for $d = 3$, $N^* \approx 200$; for $d = 4, 5$, and presumably all higher dimensions, $N^* > 4000$. Based on these benchmarking results, our package automatically selects which of the two methods is likely faster based on the dimension and samples sizes of the supplied data. If users are interested in performing more precise benchmarking for their specific dataset, the argument `nPermute` can be set equal to 0, which bypasses the permutation test and only computes the test statistic.

2.3 Significance testing

To the best of our knowledge, no results have been published concerning the distribution of the Fasano–Franceschini test statistic. Any analysis would likely be complicated by the fact that, unlike the KS test statistic, the Fasano–Franceschini test statistic is not distribution free (Fasano and Franceschini, 1987). In their original paper, Fasano and Franceschini (1987) did not attempt any analytical analysis and instead performed simulations to estimate critical values of their test statistic for various two- and three-dimensional distributions. By fitting a curve to their results, Press et al. (2007) proposed an explicit formula for p-values in the two-dimensional case. However, this formula is only approximate, and its accuracy degrades as sample sizes decrease or the true p-value becomes large (greater than 0.2). While this would still allow a simple rejection decision at any common significance level, it is sometimes useful to quantify large p-values more exactly (such as if one was to do a cross-study concordance analysis comparing p-values between studies as in Ness-Cohn et al. 2020). Effort could be made to improve this approximation, however it is still only valid in two dimensions, and thus an alternative method would be needed in higher dimensions.

To ensure the broadest applicability of the test, we assess significance using a permutation test. Let

$\mathbf{Z} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_N)$ be defined by

$$\mathbf{Z}_i = \begin{cases} \mathbf{X}_i, & 1 \leq i \leq n_1 \\ \mathbf{Y}_{i-n_1}, & n_1 + 1 \leq i \leq N \end{cases}$$

where $N = n_1 + n_2$. The test statistic in (3) can then be written as

$$\mathcal{D}(\mathbf{Z}) = \mathcal{D}(\mathbf{X}, \mathbf{Y}).$$

Denote the symmetric group on $\{1, \ldots, n\} \subset \mathbb{N}$ by S_n. For $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and $\sigma \in S_n$, define

$$\mathbf{x}_\sigma = (\mathbf{x}_{\sigma(1)}, \ldots, \mathbf{x}_{\sigma(n)}).$$

Under the null hypothesis, $\mathbf{X}$ and $\mathbf{Y}$ were drawn from the same distribution, and thus the elements of $\mathbf{Z}$ are exchangeable. We can therefore compute the permutation test p-value

$$p = \frac{\sum_{\sigma \in S_N} I(\mathcal{D}(\mathbf{Z}_\sigma) \geq \mathcal{D}(\mathbf{Z}))}{N!} \tag{4}$$

where I denotes the indicator function (Hemerik and Goeman, 2018; Ramdas et al., 2022). As it is generally infeasible to iterate over all $N!$ permutations, we can instead consider for $M \in \mathbb{N}$

$$p_M = \frac{1 + \sum_{m=1}^{M} I(\mathcal{D}(\mathbf{Z}_{\sigma_m}) \geq \mathcal{D}(\mathbf{Z}))}{1 + M} \tag{5}$$

where $\sigma_1, \ldots, \sigma_M$ are independent permutations drawn uniformly from S_N. This p-value is valid, as under the null hypothesis

$$\mathbb{P}(p_m \leq \alpha) \leq \alpha \ \forall \alpha \in [0, 1].$$

Moreover, $p_M \to p$ almost surely. These results hold for any sampling distributions (continuous, discrete, or mixed) and any valid test statistic (Hemerik and Goeman, 2018; Ramdas et al., 2022).

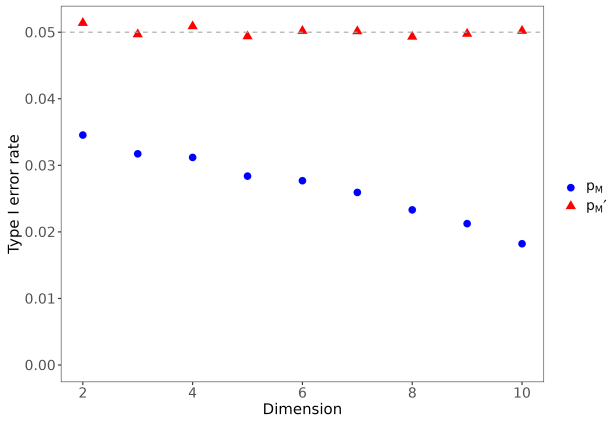

Figure 3: Type I error rate of the test using p_M and p'_M as dimension increases. Samples are both of size 10 and are drawn from standard multivariate normal distributions of the specified dimension. The number of permutations used is 100, and the error rate is estimated using 10^5 replications.

Under certain conditions (see Proposition 2 of Hemerik and Goeman 2018), the permutation test using the p-value p_M is exact, which is to say that under the null hypothesis

$$\mathbb{P}(p_m \leq \alpha) = \alpha \ \forall \alpha \in [0, 1].$$

However, when there is a nonzero probability of ties in the test statistic, this test can be quite conservative (Hemerik and Goeman, 2018). In such cases, the test can be made exact by instead using

$$p'_M = \frac{\sum_{m=1}^{M} I(\mathcal{D}(\mathbf{Z}_{\sigma_m}) > \mathcal{D}(\mathbf{Z}))}{1 + M} + U \frac{1 + \sum_{m=1}^{M} I(\mathcal{D}(\mathbf{Z}_{\sigma_m}) = \mathcal{D}(\mathbf{Z}))}{1 + M}, \tag{6}$$

where $U \sim \text{Unif}(0, 1)$ (Hoeffding, 1952; Hemerik and Goeman, 2018). The fact that p'_M is randomized is not inherently problematic, since it is already randomized due to the selection of random permutations (Hemerik and Goeman, 2018).

As the test statistic in (3) is discrete, ties are possible, and thus the test using p_M is generally conservative. In high dimensions, ties can become quite prevalent, leading the type I error rate to

decrease dramatically (Figure 3). We thus use p'_M as our p-value instead of p_M. As a final remark, we note that if the test statistic is scaled by a constant scalar, p'_M remains unchanged since both indicator functions are invariant under scalar rescaling of $\mathcal{D}$. Therefore, the outcome of the test does not depend on our choice to use the integer-valued test statistic $\mathcal{D}$ in (3) over Fasano and Franceschini's original test statistic $\mathcal{D}_0$ in (2).

3 Package overview

The **fasano.franceschini.test** package is written primarily in C++, and interfaces with R using **Rcpp** (Eddelbuettel et al., 2022). The C++ range tree class (Weihs, 2020) was based on the description in de Berg et al. (2008), including an implementation of fractional cascading. The permutation test is parallelized using **RcppParallel** (Allaire et al., 2022). The package consists of one function, `fasano.franceschini.test`, for performing the two-sample Fasano–Franceschini test. The arguments of this function are described below.

- `S1` and `S2`: the two samples to compare. Both should be either numeric `matrix` or `data.frame` objects with the same number of columns.
- `nPermute`: the number of permutations to use for performing the permutation test. The default is 100. If set equal to 0, the permutation test is bypassed and only the test statistic is computed.
- `threads`: the number of threads to use when performing the permutation test. The default is one thread. This parameter can also be set to `"auto"`, which uses the value returned by `RcppParallel::defaultNumThreads()`.
- `seed`: an optional seed for the pseudorandom number generator (PRNG) used during the permutation test.
- `verbose`: whether to display a progress bar while performing the permutation test. The default is `TRUE`. This functionality is only available when `threads = 1`.
- `method`: an optional `character` indicating which method to use to compute the test statistic. The two methods are `'r'` (range tree) and `'b'` (brute force). Both methods return the same results but may vary in computation speed. If this argument is not passed, the sample sizes and dimension of the data are used to infer which method is likely faster.

The output is an object of the class `htest`, and consists of the following components:

- `statistic`: the value of the test statistic.
- `p.value`: the permutation test p-value.
- `method`: the name of the test (i.e. `'Fasano-Franceschini Test'`).
- `data.name`: the names of the original data objects.

4 Examples

Here we demonstrate the basic usage and features of the **fasano.franceschini.test** package. We begin by loading the necessary libraries and setting a seed for reproducibility.

```
> library(fasano.franceschini.test)
> library(MASS)
> set.seed(0)
```

Note that to produce reproducible results, we need to set two seeds: the `set.seed` function sets the seed in R, ensuring we draw reproducible samples; and the seed passed as an argument to the `fasano.franceschini.test` function sets the seed for the C++ PRNG, ensuring we compute reproducible p-values.

As a first example, we draw two samples from the bivariate standard normal distribution. The Fasano–Franceschini test fails to reject the null hypothesis — that the samples were drawn from the same distribution — at an $\alpha = 0.05$ significance level.

```
> S1 <- mvrnorm(n = 50, mu = c(0, 0), Sigma = diag(2))
> S2 <- mvrnorm(n = 75, mu = c(0, 0), Sigma = diag(2))
> fasano.franceschini.test(S1, S2, seed = 1, verbose = FALSE)

        Fasano-Franceschini Test
```

```
data:  S1 and S2
D = 1425, p-value = 0.4653
```

We next draw two samples from bivariate normal distributions with identical covariance matrices but different locations. The test rejects the null hypothesis at an $\alpha = 0.05$ significance level.

```
> S3 <- mvrnorm(n = 40, mu = c(0, 0), Sigma = diag(2))
> S4 <- mvrnorm(n = 42, mu = c(1, 1), Sigma = diag(2))
> fasano.franceschini.test(S3, S4, seed = 2, verbose = FALSE)

        Fasano-Franceschini Test

data:  S3 and S4
D = 1932, p-value = 0.001832
```

The test can take a while to run when the sample sizes or the dimension of the data are large, in which case it is useful to use multiple threads to speed up computation.

```
> S5 <- mvrnorm(n = 1000, mu = c(1, 3, 5), Sigma = diag(3) + 1)
> S6 <- mvrnorm(n = 600, mu = c(1, 3, 5), Sigma = diag(3))
> fasano.franceschini.test(S5, S6, seed = 3, threads = 4)

        Fasano-Franceschini Test

data:  S5 and S6
D = 263800, p-value = 0.0007002
```

Note that the number of threads used does not affect the results. In particular, as long as the same seed is used, the same p-value is returned for any number of threads.

```
> fasano.franceschini.test(S5, S6, seed = 3, threads = 1)

        Fasano-Franceschini Test

data:  S5 and S6
D = 263800, p-value = 0.0007002
```

5 Comparison with other R packages

In this section, we compare the **fasano.franceschini.test** package with three other CRAN packages that perform multivariate two-sample goodness-of-fit tests.

5.1 Peacock.test

The **Peacock.test** package (Xiao, 2016) provides functions to compute Peacock's test statistic (Peacock, 1983) in two and three dimensions. As no function is provided to compute p-values, we cannot directly compare the performance of this package with the **fasano.franceschini.test** package. However, a treatment of the power of both Peacock and Fasano–Franceschini tests can be found in both the primary literature (Peacock, 1983; Fasano and Franceschini, 1987) and in a subsequent benchmarking paper (Lopes et al., 2007), which found that the two tests have similar power across a variety of alternatives.

5.2 cramer

The **cramer** package (Franz, 2019) implements the two-sample test described in Baringhaus and Franz (2004), which the authors refer to as the Cramér test. The Cramér test statistic is based on the Euclidean inter-point distances between the two samples, and is given by

$$T_{m,n} = \frac{mn}{m+n}\left(\frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\phi\left(\left\|\mathbf{X}_i-\mathbf{Y}_j\right\|_2^2\right)-\frac{1}{m^2}\sum_{i,j=1}^{m}\phi\left(\left\|\mathbf{X}_i-\mathbf{X}_j\right\|_2^2\right)-\frac{1}{n^2}\sum_{i,j=1}^{n}\phi\left(\left\|\mathbf{Y}_i-\mathbf{Y}_j\right\|_2^2\right)\right)$$

for samples $\mathbf{X}_1,\ldots,\mathbf{X}_m$ and $\mathbf{Y}_1,\ldots,\mathbf{Y}_n$. The default kernel function is $\phi(x) = \sqrt{x}/2$, although several other choices are implemented (see the documentation for more details). Several randomization

methods are provided to compute p-values, including bootstrapping (the default) and a permutation test.

5.3 dicproperm

The **dicproperm** package (Allmon et al., 2021) implements the DiProPerm test introduced by Wei et al. (2016). A binary linear classifier is first trained to determine a separating hyperplane between the two samples. The data are then projected onto the normal vector to the hyperplane, and the test statistic is taken to be a univariate statistic of the projected data (by default the absolute difference of means). As in the **fasano.franceschini.test** package, significance is determined using a permutation test.

5.4 Power comparison

To compare the **fasano.franceschini.test** package with the **cramer** and **dicproperm** packages, we performed power analyses using three classes of alternatives: location alternatives, where the means of the marginals are varied; dispersion alternatives, where the variances of the marginals are varied; and copula alternatives, where the marginals remain fixed but the copula joining them is varied.

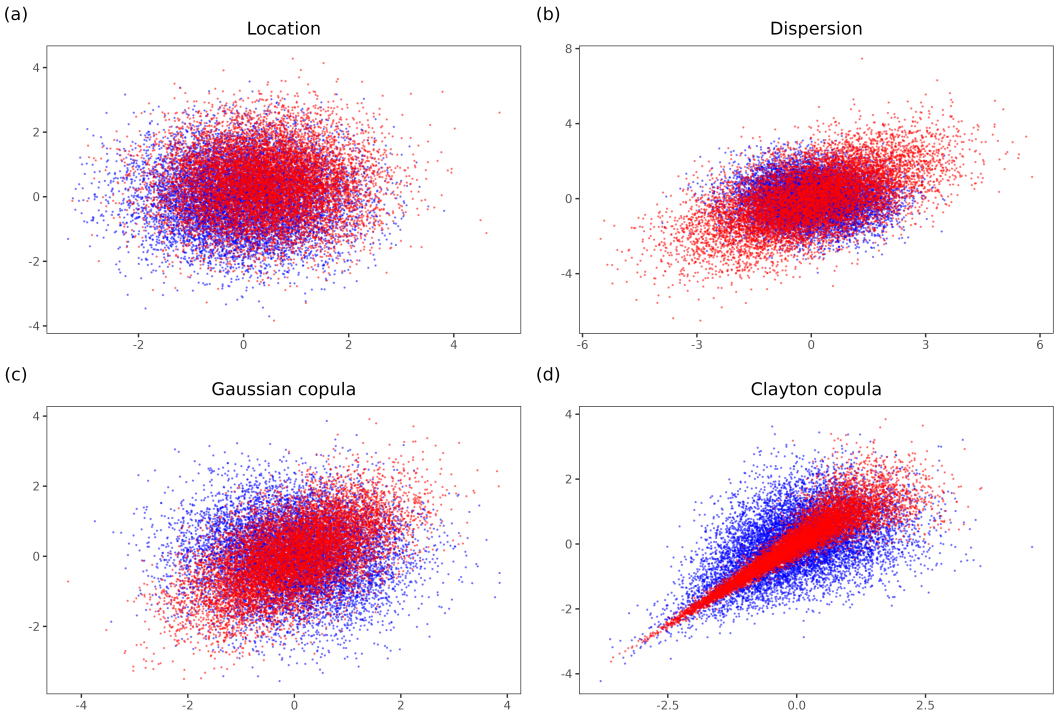

Figure 4: Visualization of the distributions used in power analyses. Each plot shows two samples consisting of 10000 points each. The first sample S_1 is shown in blue, and the second sample S_2 is shown in red. (a) $S_1 \sim N_2(\mathbf{0}, \mathbf{I}_2)$ and $S_2 \sim N_2(\mathbf{0.4}, \mathbf{I}_2)$. (b) $S_1 \sim N_2(\mathbf{0}, \mathbf{I}_2)$ and $S_2 \sim N_2(\mathbf{0}, \mathbf{I}_2 + 1.5)$. (c) $S_1 \sim G_2(0)$ and $S_2 \sim G_2(0.6)$. (d) $S_1 \sim C_2(1)$ and $S_2 \sim C_2(8)$.

For location and dispersion alternatives, we used multivariate normal distributions. We denote the d-dimensional normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ by $N_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and sample from it using the **MASS** package (Ripley, 2021). The $d \times d$ identity matrix, which is sometimes used as a covariance matrix, is denoted by $\mathbf{I}_d$. For copula alternatives, we consider the Gaussian copula with correlation matrix

$$[\mathbf{P}(\rho)]_{ij} = \begin{cases} \rho, & i \neq j \\ 1, & i = j \end{cases}$$

and the Clayton copula with parameter $\theta \in [-1, \infty) \setminus \{0\}$. We denote the d-dimensional distribution with standard normal marginals joined by a Gaussian copula with correlation matrix $\mathbf{P}(\rho)$ by $G_d(\rho)$. We denote the d-dimensional distribution with standard normal marginals joined by a Clayton copula with parameter θ by $C_d(\theta)$. Both distributions are sampled from using the **copula** package (Hofert et al., 2022). Examples of distributions in each of these four families are shown in Figure 4.

In the following analyses, power was approximated using 1000 replications, a significance level of $\alpha = 0.05$ was used, all samples were of size 40, and all R functions implementing tests were called using their default arguments. Although we aimed to cover a wide range of distributions in this analysis, absent any theoretical results concerning these three tests, we cannot guarantee that the results here are generalizable to different sampling distributions or sample sizes.

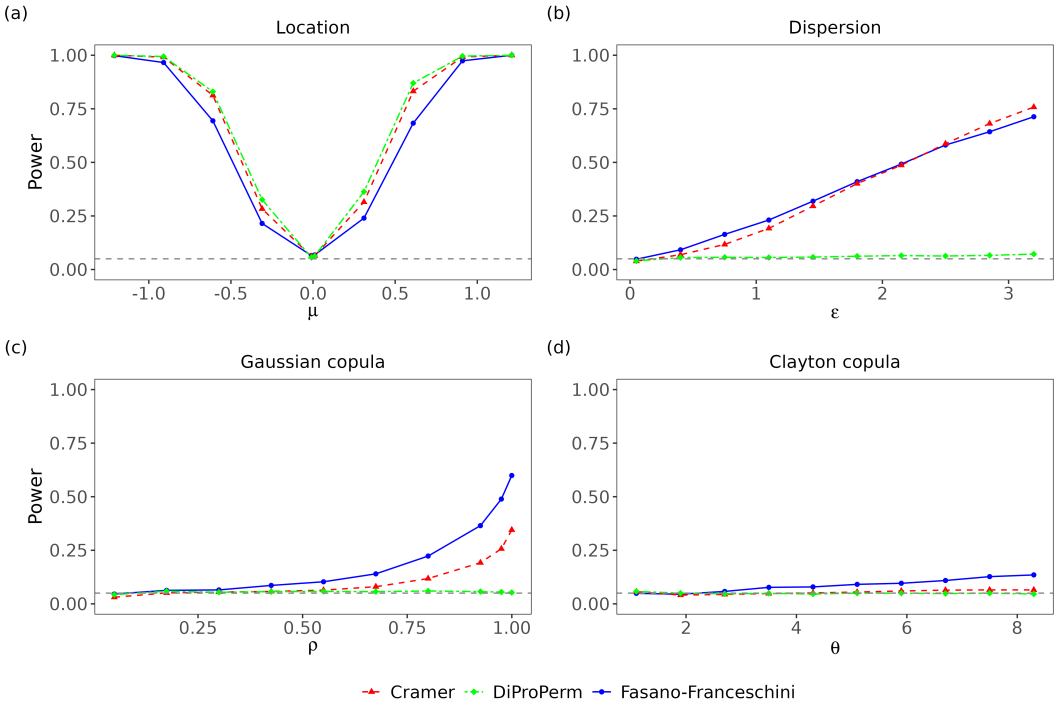

Figure 5: Comparison of power of the Fasano–Franceschini, Cramér, and DiProPerm tests on various bivariate alternatives. (a) Location alternatives, with $S_1 \sim N_2(\mathbf{0}, \mathbf{I}_2)$ and $S_2 \sim N_2(\boldsymbol{\mu}, \mathbf{I}_2)$. (b) Dispersion alternatives, with $S_1 \sim N_2(\mathbf{0}, \mathbf{I}_2)$ and $S_2 \sim N_2(\mathbf{0}, \mathbf{I}_2 + \varepsilon)$. (c) Gaussian copula alternatives, with $S_1 \sim G_2(0)$ and $S_2 \sim G_2(\rho)$. (d) Clayton copula alternatives, with $S_1 \sim C_2(1)$ and $S_2 \sim C_2(\theta)$.

We first examined the power of the tests on various bivariate alternatives (Figure 5). All three tests had similar power across location alternatives, although the Cramér and DiProPerm tests did outperform the Fasano–Franceschini test. Across dispersion alternatives, the Cramér and Fasano–Franceschini tests had very similar powers. On Gaussian copula alternatives, the Fasano–Franceschini test had a consistently higher power than the Cramér test. This was also the case with Clayton copula alternatives, although none of the tests were able to achieve high power. The DiProPerm test was unable to achieve a power above the significance level of $\alpha = 0.05$ on any of the dispersion or copula alternatives. This is likely due to the fact that in these instances, there is significant overlap between the high density regions of the two sampling distributions, making it difficult to find a separating hyperplane between samples drawn from them.

We next examined how the power of the three tests varied when the two sampling distributions were kept fixed but the dimension of the data increased (Figure 6). On the location alternative, the Cramér and DiProPerm tests again outperformed the Fasano–Franceschini test. In particular, for $d > 5$, the Fasano–Franceschini steadily lost power as the dimension increased whereas the other tests gained power. On the dispersion alternative, the Cramér and Fasano–Franceschini tests had nearly identical powers through to $d = 5$, but for higher dimensions the Cramér test consistently outperformed the Fasano–Franceschini test. On the other hand, for both the Gaussian and Clayton copula alternatives the Fasano–Franceschini test had a much higher power than the Cramér test. The DiProPerm test was still unable to attain a power above the significance level on the dispersion alternatives or either of the copula alternatives.

Overall, the Cramér and DiProPerm tests performed better than the Fasano–Franceschini test on location alternatives, especially as dimension increased. On dispersion alternatives, the Fasano–Franceschini and Cramér tests had comparable performance for low dimensions, but the Cramér test maintained a higher power for high dimensions. However, in these cases the marginal distributions differ, and thus a multivariate test is not strictly necessary as univariate tests could be applied to the marginals independently (with a multiple testing correction) to detect differences between the multivariate distributions. On copula alternatives, where a multivariate test is necessary, the Fasano–

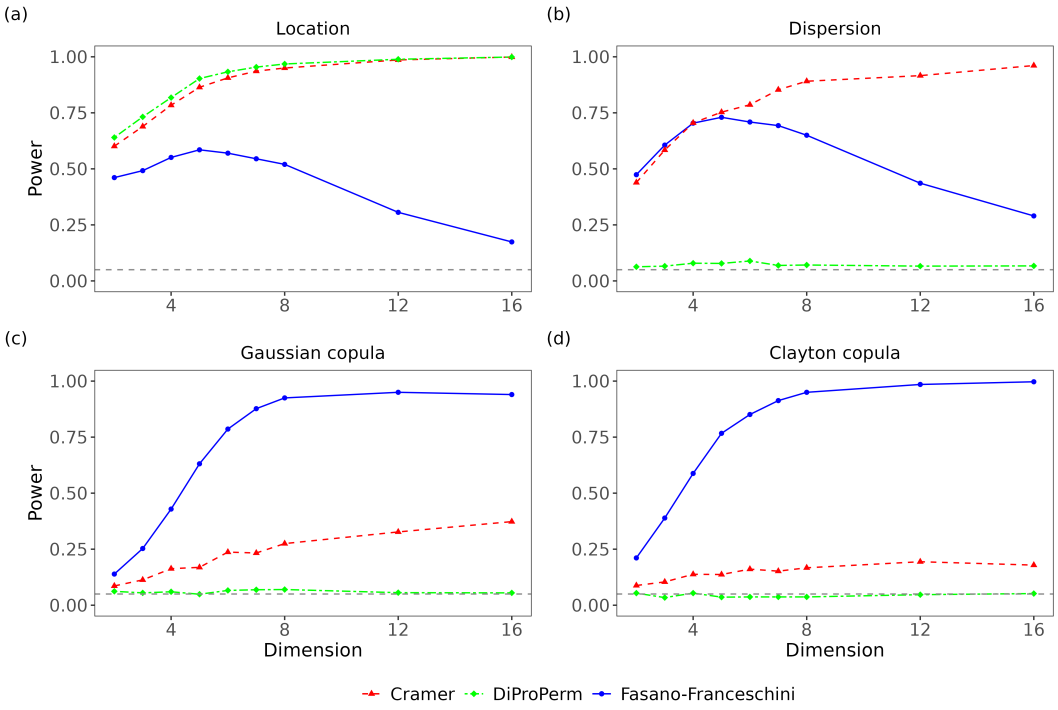

Figure 6: Comparison of power of the Fasano–Franceschini, Cramér, and DiProPerm tests on fixed alternatives as the dimension of the data increases. (a) Location alternative, with $S_1 \sim N_d(\mathbf{0}, \mathbf{I}_d)$ and $S_2 \sim N_d(\mathbf{0.4}, \mathbf{I}_d)$. (b) Dispersion alternative, with $S_1 \sim N_d(\mathbf{0}, \mathbf{I}_d)$ and $S_2 \sim N_d(\mathbf{0}, \mathbf{I}_d + 1.5)$. (c) Gaussian copula alternative, with $S_1 \sim G_d(0)$ and $S_2 \sim G_d(0.6)$. (d) Clayton copula alternative, with $S_1 \sim C_d(1)$ and $S_2 \sim C_d(8)$.

Franceschini test consistently outperformed both the Cramér and DiProPerm tests. Thus while the Fasano–Franceschini did not achieve the highest power in every case, we believe it to be the best choice as a general purpose multivariate two-sample goodness-of-fit test.

6 Summary

This paper introduces the **fasano.franceschini.test** package, an R implementation of the multivariate two-sample goodness-of-fit test described by Fasano and Franceschini (1987). We provide users with a computationally efficient test that is applicable to data of any dimension and of any type (continuous, discrete, or mixed), and that demonstrates competitive performance with similar R packages. Complete package documentation and source code are available via the Comprehensive R Archive Network (CRAN) at `https://cran.r-project.org/web/packages/fasano.franceschini.test` and the package website at `https://braunlab-nu.github.io/fasano.franceschini.test`.

7 Computational details

The results in this paper were obtained using R 4.2.0 with the packages **fasano.franceschini.test** 2.2.1, **diproperm** 0.2.0, **cramer** 0.9-3, **MASS** 7.3-60, **copula** 1.1-2, and **microbenchmark** 1.4.10 (Mersmann, 2023). Plots were generated using **ggplot2** 3.4.2 (Wickham et al., 2023) and **patchwork** 1.1.1 (Pedersen, 2020). All computations were done using the Quest high performance computing facility at Northwestern University.

8 Acknowledgments

This research was supported in part through the computational resources and staff contributions provided for the Quest high performance computing facility at Northwestern University which is jointly supported by the Office of the Provost, the Office for Research, and Northwestern University

Information Technology. Funding for this work was provided by NIH/NIA R01AG068579, Simons Foundation 597491-RWC01, and NSF 1764421-01.

References

J. Allaire, R. Francois, K. Ushey, G. Vandenbrouck, M. Geelnard, and Intel. *RcppParallel: Parallel Programming Tools for 'Rcpp'*, 2022. URL https://CRAN.R-project.org/package=RcppParallel. R package version 5.1.5. [p164]

A. G. Allmon, J. Marron, and M. G. Hudgens. *diproperm: Conduct Direction-Projection-Permutation Tests and Display Plots*, 2021. URL https://CRAN.R-project.org/package=diproperm. R package version 0.2.0. [p166]

S. Atasoy, L. Roseman, M. Kaelen, M. L. Kringelbach, G. Deco, and R. L. Carhart-Harris. Connectome-harmonic decomposition of human brain activity reveals dynamical repertoire re-organization under LSD. *Scientific Reports*, 7(1):1–18, 2017. URL https://doi.org/10.1038/s41598-017-17546-0. [p159]

L. Baringhaus and C. Franz. On a new multivariate two-sample test. *Journal of Multivariate Analysis*, 88(1):190–206, 2004. ISSN 0047-259X. URL https://doi.org/10.1016/S0047-259X(03)00079-4. [p165]

J. L. Bentley. Decomposable searching problems. *Information Processing Letters*, 8(5):244–251, 1979. ISSN 0020-0190. URL https://doi.org/10.1016/0020-0190(79)90117-0. [p161]

F. Chiang, O. Mazdiyasni, and A. AghaKouchak. Amplified warming of droughts in southern united states in observations and model simulations. *Science Advances*, 4(8):eaat2380, 2018. URL https://doi.org/10.1126/sciadv.aat2380. [p159]

M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer Berlin Heidelberg, 3rd edition, 2008. ISBN 978-3-540-77974-2. URL https://doi.org/10.1007/978-3-540-77974-2. [p161, 164]

D. Eddelbuettel, R. Francois, J. Allaire, K. Ushey, Q. Kou, N. Russell, I. Ucar, D. Bates, and J. Chambers. *Rcpp: Seamless R and C++ Integration*, 2022. URL https://CRAN.R-project.org/package=Rcpp. R package version 1.0.9. [p164]

G. Fasano and A. Franceschini. A multidimensional version of the Kolmogorov-Smirnov test. *Monthly Notices of the Royal Astronomical Society*, 225(1):155–170, 03 1987. ISSN 0035-8711. URL https://doi.org/10.1093/mnras/225.1.155. [p159, 160, 162, 165, 168]

C. Franz. *cramer: Multivariate Nonparametric Cramer-Test for the Two-Sample-Problem*, 2019. URL https://CRAN.R-project.org/package=cramer. R package version 0.9-3. [p165]

J. M. Hahne, M. A. Schweisfurth, M. Koppe, and D. Farina. Simultaneous control of multiple functions of bionic hand prostheses: Performance and robustness in end users. *Science Robotics*, 3(19):eaat3630, 2018. URL https://doi.org/10.1126/scirobotics.aat3630. [p159]

J. Hemerik and J. Goeman. Exact testing with random permutations. *TEST*, 27(4):811–825, Dec 2018. ISSN 1863-8260. URL https://doi.org/10.1007/s11749-017-0571-1. [p163]

C. Heuchenne and G. Mordant. Using space filling curves to compare two multivariate distributions with distribution-free tests. *Journal of Computational and Applied Mathematics*, 416:114494, Dec. 2022. ISSN 0377-0427. URL https://doi.org/10.1016/j.cam.2022.114494. [p159]

W. Hoeffding. The Large-Sample Power of Tests Based on Permutations of Observations. *The Annals of Mathematical Statistics*, 23(2):162–192, 1952. ISSN 0003-4851. URL https://doi.org/10.1214/aoms/1177729436. [p163]

M. Hofert, I. Kojadinovic, M. Maechler, and J. Yan. *copula: Multivariate Dependence with Copulas*, 2022. URL https://CRAN.R-project.org/package=copula. R package version 1.1-0. [p166]

A. Justel, D. Peña, and R. Zamar. A multivariate Kolmogorov-Smirnov test of goodness of fit. *Statistics & Probability Letters*, 35(3):251–259, 1997. ISSN 0167-7152. URL https://doi.org/10.1016/S0167-7152(97)00020-5. [p159]

S. Kaczanowska, D. W. Beury, V. Gopalan, A. K. Tycko, H. Qin, M. E. Clements, J. Drake, C. Nwanze, M. Murgai, Z. Rae, W. Ju, K. A. Alexander, J. Kline, C. F. Contreras, K. M. Wessel, S. Patel, S. Hannenhalli, M. C. Kelly, and R. N. Kaplan. Genetically engineered myeloid cells rebalance the core immune suppression program in metastasis. *Cell*, 184(8):2033–2052.e21, 2021. ISSN 0092-8674. URL https://doi.org/10.1016/j.cell.2021.02.048. [p159]

A. N. Kolmogorov. Sulla Determinazione Empirica di Una Legge di Distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4:83–91, 1933a. [p159]

A. N. Kolmogorov. Über die Grenzwertsätze der Wahrscheinlichkeitsrechnung. *Bull. Acad. Sci. URSS*, 3:363–372, 1933b. URL https://www.mathnet.ru/eng/im5009. [p159]

R. H. C. Lopes, I. Reid, and P. R. Hobson. The two-dimensional Kolmogorov-Smirnov test. In *XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research*, 2007. URL https://bura.brunel.ac.uk/handle/2438/1166. [p159, 165]

O. Mersmann. *microbenchmark: Accurate Timing Functions*, 2023. URL https://CRAN.R-project.org/package=microbenchmark. R package version 1.4.10. [p168]

M. Naaman. On the tight constant in the multivariate Dvoretzky–Kiefer–Wolfowitz inequality. *Statistics & Probability Letters*, 173:109088, 2021. ISSN 0167-7152. URL https://doi.org/10.1016/j.spl.2021.109088. [p159]

E. Ness-Cohn, M. Iwanaszko, W. L. Kath, R. Allada, and R. Braun. TimeTrial: An Interactive Application for Optimizing the Design and Analysis of Transcriptomic Time-Series Data in Circadian Biology Research. *Journal of Biological Rhythms*, 35(5):439–451, 2020. URL https://doi.org/10.1177/0748730420934672. PMID: 32613882. [p162]

J. A. Peacock. Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notices of the Royal Astronomical Society*, 202(3):615–627, 03 1983. ISSN 0035-8711. URL https://doi.org/10.1093/mnras/202.3.615. [p159, 165]

T. L. Pedersen. *patchwork: The Composer of Plots*, 2020. URL https://CRAN.R-project.org/package=patchwork. R package version 1.1.1. [p168]

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, USA, 3rd edition, 2007. ISBN 0521880688. URL https://doi.org/10.1145/1874391.187410. [p162]

A. Ramdas, R. F. Barber, E. J. Candes, and R. J. Tibshirani. Permutation tests using arbitrary permutation distributions. 2022. URL https://doi.org/10.48550/arXiv.2204.13581. [p163]

B. Ripley. *MASS: Support Functions and Datasets for Venables and Ripley's MASS*, 2021. URL https://CRAN.R-project.org/package=MASS. R package version 7.3-54. [p166]

N. V. Smirnov. Sur la distribution de ω^2 (criterium de M.R. v. Mises). *Com. Rend. Acad. Sci. (Paris)*, 202: 449–452, 1936. [p159]

N. V. Smirnov. On the distribution of the mises ω^2 criterion [in Russian]. *Rec. Math. N.S. [Mat. Sbornik]*, 2:973–993, 1937. URL https://www.mathnet.ru/eng/sm5636. [p159]

N. V. Smirnov. On the deviations of the empirical distribution curve [in Russian]. *Rec. Math. N.S. [Mat. Sbornik]*, 6(48):3–26, 1939. URL https://www.mathnet.ru/eng/sm5810. [p159]

N. V. Smirnov. Approximate laws of distribution of random variables from empirical data. *Uspehi Matem. Nauk*, 10:179–206, 1944. [p159]

N. V. Smirnov. Table for estimating the goodness of fit of empirical distributions. *The Annals of Mathematical Statistics*, 1948. ISSN 0003-4851. URL https://doi.org/10.1214/aoms/1177730256. [p159]

S. Wei, C. Lee, L. Wichers, and J. S. Marron. Direction-Projection-Permutation for High-Dimensional Hypothesis Tests. *Journal of Computational and Graphical Statistics*, 25(2):549–569, 2016. URL https://doi.org/10.1080/10618600.2015.1027773. [p166]

L. Weihs. C++ Range Tree Data Structure, 2020. URL https://github.com/Lucaweihs/range-tree. [p164]

H. Wickham, W. Chang, L. Henry, T. L. Pedersen, K. Takahashi, C. Wilke, K. Woo, H. Yutani, and D. Dunnington. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*, 2023. URL https://CRAN.R-project.org/package=ggplot2. R package version 3.4.2. [p168]

F. Wong and J. J. Collins. Evidence that coronavirus superspreading is fat-tailed. *Proceedings of the National Academy of Sciences*, 117(47):29416–29418, 2020. URL https://doi.org/10.1073/pnas.2018490117. [p159]

Y. Xiao. *Peacock.test: Two and Three Dimensional Kolmogorov-Smirnov Two-Sample Tests*, 2016. URL https://CRAN.R-project.org/package=Peacock.test. R package version 1.0. [p165]

Connor Puritz
Department of Engineering Sciences and Applied Mathematics, Northwestern University
Evanston, IL 60208
ORCiD: 0000-0001-7602-0444
connorpuritz2025@u.northwestern.edu

Elan Ness-Cohn
Department of Molecular Biosciences, Northwestern University
Evanston, IL 60208
ORCiD: 0000-0002-3935-6667
elan.ness-cohn@northwestern.edu

Rosemary Braun
Department of Molecular Biosciences, Northwestern University
Evanston, IL 60208
ORCiD: 0000-0001-9668-9866
rbraun@northwestern.edu