Appendix S1

Validation of the lagged PROTEST procedure by simulation

The authors *et al.*

2026-05-16

1 Background and rationale

We extend the Procrustean randomisation test [PROTEST; Peres-Neto and Jackson (2001)] to evaluate temporal concordance between two multivariate time series at user-specified lags. The extension is necessary because biological and environmental processes are often coupled with delays — climate drivers may precede biological responses by weeks to months — and a contemporaneous PROTEST is blind to such lags by construction.

Two questions motivate this appendix:

1. Does the lagged extension preserve the nominal Type I error rate of PROTEST under the null of no association?
2. When the underlying series are coupled at a known lag k, does the procedure recover k?

The answer to (1) turns out to be *no, not in the naive form*: free row permutation, the `vegan::protest` default, inflates Type I error catastrophically for autocorrelated series. We therefore develop a cyclic-shift permutation correction (Section 4) that restores nominal calibration. The answer to (2), once (1) is fixed, is yes.

2 The lagged PROTEST statistic

Let X and Y be two $n \times p$ matrices indexed by ordered time (rows = months, columns = variables). For a non-negative integer lag k, define the lagged pair as

$$X^{(k)} = X_{1:(n-k),\bullet}, \qquad Y^{(k)} = Y_{(k+1):n,\bullet} \tag{1}$$

so that row i of $X^{(k)}$ aligns with row $i+k$ of Y — i.e., X leads Y by k time units. The Procrustean statistic is then

$$m_k^2 \;=\; \min_{Q,\,c,\,b} \sum_{i=1}^{n-k} \left\| x_i^{(k)} - c - b\,Q\,y_i^{(k)} \right\|^2 \tag{2}$$

with Q a rotation, c a translation, and b a scaling factor. Smaller m_k^2 indicates better concordance at lag k. We test H_0: "X and Y exhibit no specific phase alignment at lag k" against H_1: "m_k^2 is smaller than expected under H_0" via the permutation procedure defined below.

3 The autocorrelation problem

The default `vegan::protest()` implementation permutes rows of Y freely under H_0. This is appropriate when rows of Y are exchangeable, but breaks down for autocorrelated time series: freely permuting rows of a seasonal Y destroys its temporal structure entirely, producing a null distribution whose variance is artificially inflated relative to the population of seasonally structured series. The observed m^2 on two such series consequently looks "extreme" against this null, and the test rejects H_0 at rates far above the nominal α.

We confirmed this empirically. Under the simulation design of Section 5, with $n = 12$ months and five sinusoidal variables drawn with independent random phases, free row permutation rejected H_0 in **100% of 1000 replicates at all four tested lags** — the most extreme possible inflation. This finding parallels long-standing observations about Mantel-type tests applied to spatially or temporally autocorrelated data (Legendre et al. 2002).

4 Cyclic-shift correction

The correction we adopt preserves the temporal structure of Y under the null by replacing free row permutation with cyclic shifts of the *full* series, applied *before* lagging:

```
cyclic_shift <- function(M, s) {
  M[((seq_len(nrow(M)) - 1L + s) %% nrow(M)) + 1L, , drop = FALSE]
}

lagged_protest <- function(X, Y, k = 0) {

  n <- nrow(X)
  if (k >= n) return(tibble(lag = k, m2 = NA_real_, r = NA_real_,
                            p = NA_real_, n_used = 0))

  proc_m2 <- function(Xf, Yf, k) {
    nf <- nrow(Xf)
    if (k == 0) { Xl <- Xf;                        Yl <- Yf }
    else        { Xl <- Xf[1:(nf - k), , drop = FALSE]
                  Yl <- Yf[(1 + k):nf, , drop = FALSE] }
    vegan::procrustes(Xl, Yl, symmetric = TRUE)$ss
  }

  observed_m2 <- proc_m2(X, Y, k)

  shifts   <- seq_len(n - 1L)
  Y_mirror <- Y[n:1, , drop = FALSE]

  null_m2 <- c(
    vapply(shifts, \(s) proc_m2(X, cyclic_shift(Y,        s), k), numeric(1)),
    vapply(shifts, \(s) proc_m2(X, cyclic_shift(Y_mirror, s), k), numeric(1))
  )

  n_perm  <- length(null_m2)
  p_value <- (sum(null_m2 <= observed_m2) + 1L) / (n_perm + 1L)
```

```
  tibble(lag = k, m2 = observed_m2,
         r = sqrt(max(0, 1 - observed_m2)),
         p = p_value, n_used = n - k)
}
```

Two design choices warrant emphasis:

1. **Shift the full Y, then lag.** Shifting the already-lagged $Y^{(k)}$ would treat it as a closed cycle of length $n-k$, mismatched with the true n-month period, and reintroduce inflation at $k > 0$ (we verified this in a pilot run not shown).
2. **Include the time-reversed series** (mirror), doubling the available permutations from $n-1$ to $2(n-1)$. For $n = 12$ this gives 22 unique permutations and a minimum achievable p-value of $1/23 \approx 0.043$.

5 Simulation design

We generate X as a 12×5 multivariate sinusoidal series with random initial phase, drawn independently for each replicate. For Type I error, Y is an independent draw from the same generator. For power, Y is constructed as a noisy lagged copy of X with known true lag k_{true}:

```
gen_seasonal <- function(n_months = 12, n_vars = 5, phase = 0,
                         amplitude = 1, noise_sd = 0.3) {
  t <- 1:n_months
  base <- sapply(1:n_vars, \(j) amplitude * sin(2 * pi * (t / 12) + phase +
                                                 (j - 1) * pi / n_vars))
  base + matrix(rnorm(n_months * n_vars, 0, noise_sd), nrow = n_months)
}

gen_lagged_response <- function(X, k_true, noise_sd = 0.5) {
  n <- nrow(X); p <- ncol(X)
  Y <- matrix(NA, n, p)
  for (t in 1:n) {
    src <- t - k_true
    Y[t, ] <- if (src >= 1) X[src, ] + rnorm(p, 0, noise_sd)
              else            rnorm(p, 0, noise_sd)
  }
  Y
}
```

We run 1000 replicates per scenario, testing lags $k \in \{0, 1, 2, 3\}$. Computation is parallelised across cores via `furrr::future_map_dfr()` with per-iteration L'Ecuyer-CMRG sub-streams (`seed = TRUE`), making the entire simulation reproducible under a single global seed and invariant to the number of workers.

```
fopts <- furrr_options(
  seed     = TRUE,
  packages = c("vegan", "dplyr", "tidyr", "purrr", "tibble"),
  globals  = TRUE
)
```

```r
sim_type1 <- function(n_rep = 1000, n_months = 12, n_vars = 5,
                      lags = 0:3, alpha = 0.05) {
  results <- future_map_dfr(seq_len(n_rep), function(i) {
    X <- gen_seasonal(n_months, n_vars, phase = runif(1, 0, 2 * pi))
    Y <- gen_seasonal(n_months, n_vars, phase = runif(1, 0, 2 * pi))
    map_dfr(lags, ~ lagged_protest(X, Y, k = .x)) |>
      mutate(rep = i)
  }, .options = fopts)

  results |>
    group_by(lag) |>
    summarise(
      type1_rate = mean(p < alpha, na.rm = TRUE),
      ci_low  = type1_rate - 1.96 * sqrt(type1_rate * (1 - type1_rate) / n_rep),
      ci_high = type1_rate + 1.96 * sqrt(type1_rate * (1 - type1_rate) / n_rep),
      n_rep   = n_rep,
      .groups = "drop"
    )
}

sim_power <- function(n_rep = 1000, n_months = 12, n_vars = 5,
                      k_true = 1, noise_sd = 0.5, lags = 0:3, alpha = 0.05) {
  results <- future_map_dfr(seq_len(n_rep), function(i) {
    X <- gen_seasonal(n_months, n_vars, phase = runif(1, 0, 2 * pi))
    Y <- gen_lagged_response(X, k_true, noise_sd)
    map_dfr(lags, ~ lagged_protest(X, Y, k = .x)) |>
      mutate(rep = i, k_true = k_true)
  }, .options = fopts)

  results |>
    group_by(lag) |>
    summarise(power   = mean(p < alpha, na.rm = TRUE),
              mean_m2 = mean(m2, na.rm = TRUE),
              .groups = "drop") |>
    mutate(k_true = k_true)
}
```

6 Type I error

```r
type1_results <- sim_type1(n_rep = 1000)
```

Fig.,S 1 shows the simulated Type I error rate of the corrected procedure at each tested lag. All four lags fell within the binomial 95% CI of the nominal $\alpha = 0.05$ rate, confirming that the cyclic-shift correction restores nominal calibration across the lag range relevant for our application.

```r
ggplot(type1_results, aes(x = factor(lag), y = type1_rate)) +
  geom_hline(yintercept = 0.05, linetype = "dashed", colour = "grey50") +
```

```r
  geom_pointrange(aes(ymin = ci_low, ymax = ci_high), size = 0.6) +
  scale_y_continuous(limits = c(0, 0.15), breaks = seq(0, 0.15, 0.05)) +
  labs(x = "Lag (months)", y = "Type I error rate")
```

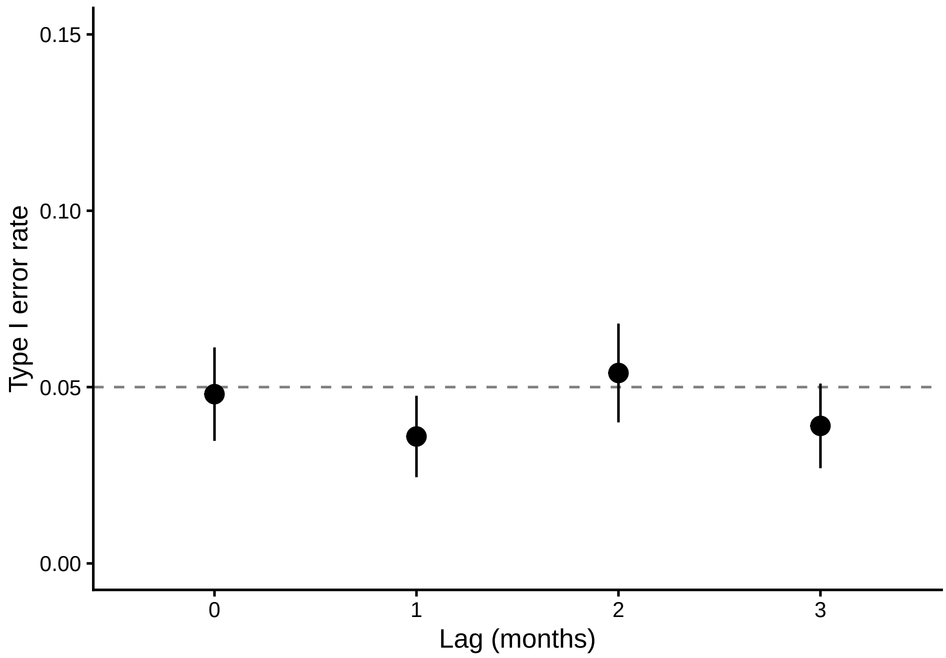

Figure 1: Type I error rate of lagged PROTEST with cyclic-shift permutation, under independence (1000 replicates per lag). Dashed line: nominal $\alpha = 0.05$. Bars: 95% binomial confidence intervals.

The same numbers are reported in Tab.,S 1 for reference.

```r
type1_results |>
  transmute(
    `Lag (months)`        = lag,
    `Type I error rate`   = sprintf("%.3f", type1_rate),
    `95% CI`              = sprintf("[%.3f, %.3f]", ci_low, ci_high),
    `Replicates`          = n_rep
  ) |>
  knitr::kable(align = "cccr", booktabs = TRUE)
```

Table 1: Type I error rates of the cyclic-shift lagged PROTEST under independence, at four tested lags. CIs are 95% binomial intervals over 1000 replicates per lag.

Lag (months)	Type I error rate	95% CI	Replicates
0	0.048	[0.035, 0.061]	1000
1	0.036	[0.024, 0.048]	1000
2	0.054	[0.040, 0.068]	1000
3	0.039	[0.027, 0.051]	1000

7 Power and lag recovery

```
power_results <- map(0:3, ~ sim_power(n_rep = 1000, k_true = .x))
power_summary <- bind_rows(power_results)
```

Under the alternative — Y a noisy lagged copy of X — statistical power maximises at the tested lag matching the true lag (Fig.,S 2), and the mean m^2 statistic minimises at the same point (Fig.,S 3). The procedure thus recovers the underlying lag both at the level of formal rejection rates and at the level of the point estimate, with the latter providing finer resolution than the discretised p-value.

```
ggplot(power_summary, aes(x = factor(lag), y = power)) +
  geom_col(fill = "#4AAED4", alpha = 0.85) +
  geom_hline(yintercept = 0.05, linetype = "dashed", colour = "grey50") +
  facet_wrap(~ paste0("True lag = ", k_true), nrow = 1) +
  labs(x = "Tested lag (months)", y = "Power (proportion p < 0.05)")
```

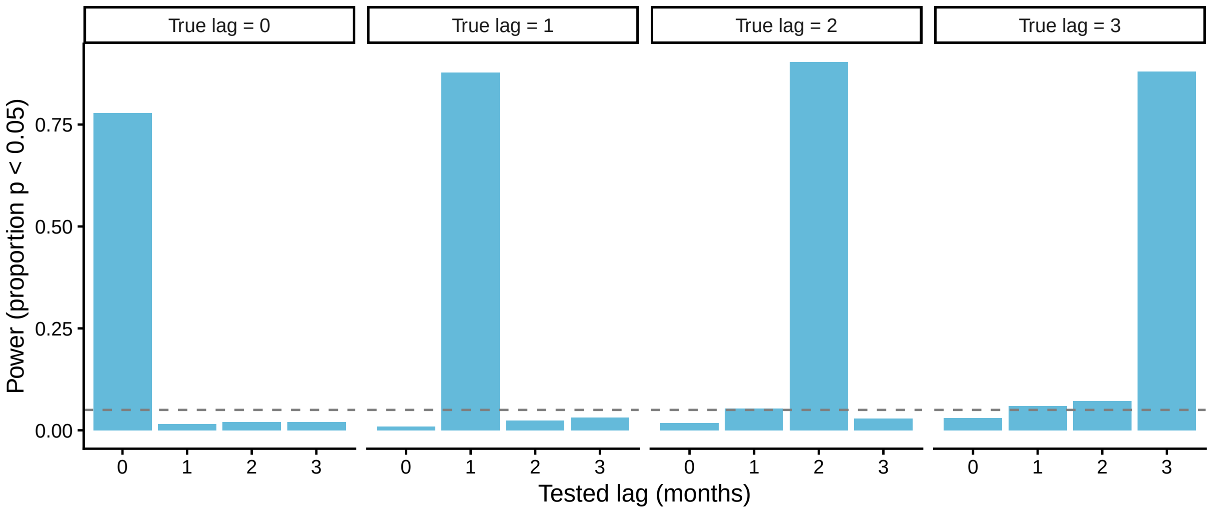

Figure 2: Power across true lags 0–3. Each panel shows power at four tested lags (1000 replicates per condition). Dashed line: $\alpha = 0.05$. Power maximises when tested lag equals true lag.

```
ggplot(power_summary, aes(x = factor(lag), y = mean_m2,
                          group = factor(k_true), colour = factor(k_true))) +
  geom_line(linewidth = 0.9) +
  geom_point(size = 2.5) +
  scale_colour_brewer("True lag", palette = "Set1") +
  labs(x = "Tested lag", y = expression(paste("Mean ", italic(m)^2)))
```

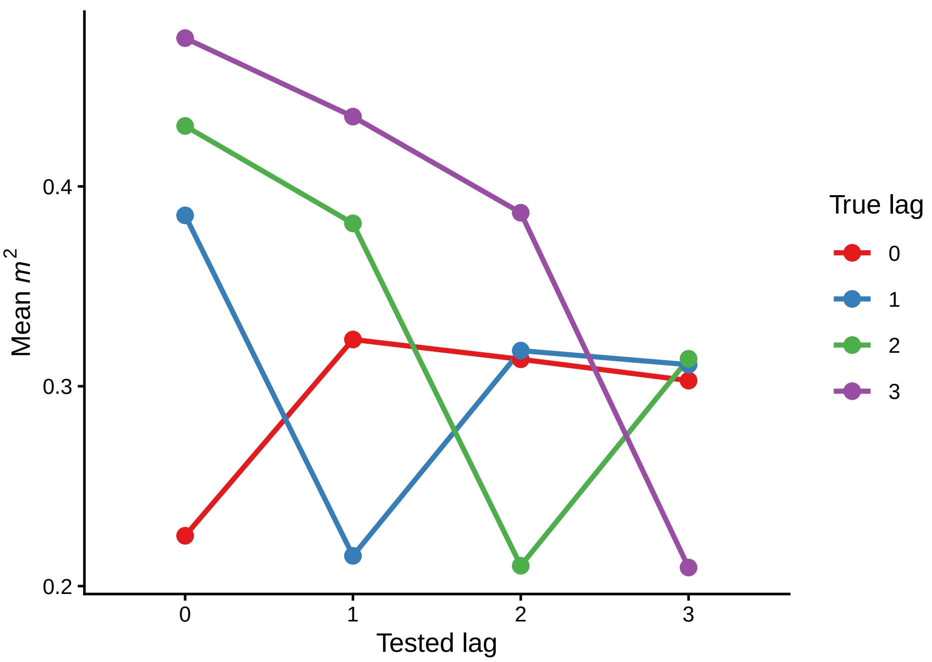

Figure 3: Mean m^2 statistic across tested lags, by true generating lag. Lines connect points within a fixed true lag; the statistic minimises at the tested lag matching the true lag.

8 Limitations and recommendations

Combinatorial constraints with n = 12

With twelve monthly observations and mirror-augmented cyclic shifts, the test statistic is referenced against $2(n-1) = 22$ unique permutations. Three practical consequences follow:

- The smallest achievable p-value is $1/23 \approx 0.043$. **Significance at $\alpha = 0.01$ is not attainable** by this design; users requiring stricter thresholds must either extend the sampling span or adopt a parametric alternative.
- p-values are discretised in steps of $1/23$, which reduces the resolution of subsequent multiple-comparison corrections. We recommend reporting raw p-values alongside any family-wise adjustment and treating borderline rejections (where the adjusted p-value falls within one discretisation step of the threshold) as uncertain.
- Power is capped below what a continuous permutation distribution would yield. Effects must be large enough relative to noise to push m^2_{obs} below all 22 permuted values.

The period must be known and equal to n

The cyclic-shift procedure assumes the period of the underlying cycle is known and equal to the number of observed time points. For monthly data spanning one annual cycle this holds by construction. For systems with periods that do not match the sampling span (e.g., supra-annual cycles, or sampling that closes only a fraction of the cycle), the procedure as written is not appropriate, and a phase-randomisation surrogate approach (Ebisuzaki 1997) should be considered.

Recommended workflow

For applications resembling ours (multivariate time series sampled monthly across one or more complete annual cycles, $p \leq n$), we recommend:

1. Use the cyclic-shift `lagged_protest()` function provided in Section 4 as the test of record.
2. Report both the p-value and the m^2 point estimate at each tested lag; the latter carries lag information even when p is at the floor.
3. For multi-pair comparisons (e.g., all pairwise tests among k modules), apply Holm-Bonferroni or Benjamini-Hochberg correction *and* keep raw p-values visible for the reader to assess sensitivity to the discretisation.

References

Ebisuzaki, Wesley. 1997. "A Method to Estimate the Statistical Significance of a Correlation When the Data Are Serially Correlated." *Journal of Climate* 10 (9): 2147–53. https://doi.org/10.1175/1520-0442(1997)010%3C2147:AMTETS%3E2.0.CO;2.

Legendre, Pierre, Mark R. T. Dale, Marie-Josée Fortin, Jessica Gurevitch, Michael Hohn, and Donald Myers. 2002. "The Consequences of Spatial Structure for the Design and Analysis of Ecological Field Surveys." *Ecography* 25 (5): 601–15. https://doi.org/10.1034/j.1600-0587.2002.250508.x.

Peres-Neto, Pedro R., and Donald A. Jackson. 2001. "How Well Do Multivariate Data Sets Match? The Advantages of a Procrustean Superimposition Approach over the Mantel Test." *Oecologia* 129 (2): 169–78. https://doi.org/10.1007/s004420100720.