Appendix A. The number of publications in the period 1981-2008 listed on Web of Science (2009) found using the search term “null model* AND (cooccurrence* OR co-occurrence*)".
Appendix B. Ordered matrices with coefficients of variation in species abundance of 0 (A) and 1 (B).
Appendix C. Type 2 errors incurred by the c-score method of assessing patterns of species co-occurrence for a range of numbers of sites, numbers of species and coefficients of variation in species abundance. Type 2 errors were estimated by progressively randomizing ordered matrices. All errors are given for a critical p-value of 0.1 (two-tailed). 30,000 swaps were used to generate null matrices for testing the significance of the metric.
Appendix D. The majority of c-score analyses using the sequential swap to generate null matrices only use 5,000 swaps. Larger matrices are not always analysed using larger numbers of swaps. Data are presented only for the 141 matrices from 24 papers found during literature surveys (Appendix A) that reported the numbers of swaps used.
Appendix E. Stabilisation of p-values with increasing number of swaps for an unstructured matrix of 50 species and 50 sites. Results from a matrix with a marginally non-significant p-value for 100,000 swaps are presented here. The broken line indicates the critical p-value of 0.05 for this test (upper tail presented here). P-values do not stabilise at marginal non-significance until after 50,000 swaps have been carried out. See Appendix G for the R code that carries out this analysis.
Appendix F. Publications used to generate Figure 1.


Appendix G. R code to assess the stabilisation of p-values of significance with increasing number of swaps. See Appendix E for an example of the graphical output from this code.

```r
require(vegan)
nreps <- 100000 # maximum number of swaps for which to calculate p-values
nestedchecker(x)$statistic > obs # x is the species/sites matrix
output <- matrix(nrow = nreps, ncol = 3)
colnames(output) <- c("c.scores", "utp", "ltb")
commsimulator(x, "swap", thin = 30000) -> base
for (i in 1:nreps) {
  commsimulator(base, "swap", thin = 1) -> base
  nestedchecker(base)$statistic -> output[i, 1]
  length(subset(output[1:i, 1], output[1:i, 1] > obs)) / i -> output[i, 2]
  length(subset(output[1:i, 1], output[1:i, 1] < obs)) / i -> output[i, 3]
}
par(mfrow = c(2, 1))
plot(output[, 2])
abline(h = 0.05, lty = "dashed")
plot(output[, 3])
abline(h = 0.05, lty = "dashed")
```