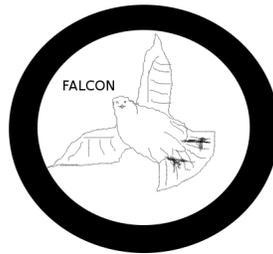


# A quick start guide to FALCON

Stephen J. Beckett\*, Chris A. Boulton, Hywel T. P. Williams

July 11, 2014

College of Life and Environmental Sciences, University of Exeter, Exeter, UK



\*author for correspondence: S.J.Beckett@exeter.ac.uk

## Introduction

FALCON is a free software package for calculating and comparing nestedness in bipartite networks. FALCON is available to download from <https://github.com/sjbeckett/FALCON>. This document is to be used as a quick start guide or as a reference page to using FALCON for the calculation of nestedness. On the following page is a quick set of instructions to get started with FALCON for those already familiar with MATLAB/Octave or R programming languages and nestedness analysis. It may also serve well as a reference guide for those who have read the more detailed instruction manual. Additional documents that users may be interested in reading are:

- **FALCON\_Manuscript.pdf** – a document that describes the concept of nestedness analysis and details the methodology used by FALCON
- **FALCON\_InstructionGuide.pdf** – a more in depth practical guide to using FALCON
- **FALCON\_Development.pdf** – some of the coding and development issues faced during the development of FALCON

## Interactive Mode

1. Navigate to FALCON/MATLAB (if using MATLAB or Octave) or FALCON/R (if using R)
2. Run interactive mode, in MATLAB/Octave just type `InteractiveMode`  
whilst in R type: `source('InteractiveMode.R')` to begin.
3. Follow the on screen instructions!

## Command Line Mode

1. Navigate to FALCON/MATLAB (if using MATLAB or Octave) or FALCON/R (if using R)
2. If using R: `source('PERFORM_NESTED_TEST.R')`
3. Run FALCON using the command :

```
PERFORM_NESTED_TEST(MATRIX, binary, sortVar, MEASURE, nulls, ensNum, plotON)
```

where:

MATRIX – the matrix array you wish to test for nestedness

binary – whether the analysis you wish to perform is binary (1), quantitative (0) or both (2)

sortVar – whether matrices should be preliminary ordered (1) to maximise nestedness found in the input and null matrices or not (0)

MEASURE – is a list of the nestedness measures to be used, which call the files in the MEASURES folder. An example list can be written in MATLAB as: `{'NODF','DISCREPANCY','SPECTRAL_RADIUS'}` (note the braces, {}, are important here!). In R this would be: `c('NODF','DISCREPANCY','SPECTRAL_RADIUS')`

nulls – the vector of null models to be used. For example to use the SS and FF null models this would be `[1 2]` in MATLAB and `c(1,2)` in R. The full set can be called using the empty list; in MATLAB: `[]` and in R: `c()`

ensNum – the type of ensemble to use. If set to the empty list, the adaptive solver will be called. Alternatively this variable can be set the number of null matrices you wish to use in your ensemble

sortVar – whether matrices should be preliminary ordered (1) to maximise nestedness found in the input and null matrices or not (0)

plotON – whether to return plots of the measured distributions for each null model (1) or not (0)