----------Read me file ----------

OptiPOPd: Software to estimate parameters of a population matrix model using a combinatorial algorithm: an investigation and evaluation of the method using simulated trajectories of Northern Spotted Owl

Cornell Wildlife Health Lab,
Animal Health Diagnostic Center,
College of Veterinary Medicine,
Cornell University
Ithaca, New York, USA

Contacts

Software programmer and modeler;
Brenda Hanley, bjh262@cornell.edu (temporary), fomalhaut14@gmail.com (permanent)

Research collaborators:
Brian Dennis, brian@uidaho.edu (permanent)
David Kramer, david.kramer@dec.ny.gov (permanent)
Krysten Schuler, ks833@cornell.edu (permanent)

Overview of the software application

OptiPOPd software contains an optimization algorithm that is designed to fill matrix parameters given 29 years of adult time series data and the theoretical population matrix model of Northern Spotted Owl (Noon & Biles, 1990). As well, this OptiPOPd software contains capabilities to evaluate the performance of the algorithm by comparing algorithm-predicted results against the truth.

The algorithm contains four types of noise (demographic, environmental, both demographic and environmental, and no noise; deterministic), four distance measure types (absolute value, log transformation, sum of squares, and square root transformations), and two types of initial conditions (in and out of stable status). Adult time series trajectories 29 years in length are obtained through simulation to contain every combination of noise, distance type, and initial condition. The algorithm assumes that the time series data is free from sampling error.

Given nothing but the abundances in the time series data for any given trajectory, the algorithm searches the high-dimensional parameter space for combinations of matrix elements that could have produced the target time series trajectory. The software saves the models with trajectories that are among the top 1% or top 0.1% closest to truth. The population matrices are analyzed and the algorithm-predicted eigenvalues, matrix elements, and superparameters are compared against truth.

IMPORTANT: This code contains a life history skeleton and parameter values that are specific to Northern Spotted Owl (Noon & Biles, 1990). Use of this code with other life histories, parameter values, or algorithmic technical details *demands technical reprogramming.*

Interactive Software User Tutorial

Preparing the app for use on your computer:
Step 1: Download the OptiPOPd software code and the pre saved .txt files.
Step 2: Open R software.
Step 3: Install the R package "MASS". To install packages, type install.packages("MASS") into the console and hit "Run."
Step 4: Set your working directory to the location where you saved the .txt files. To set the working directory in R, highlight the console, click "File", then "Change dir…" then navigate to the location where the .txt files are saved. Click OK. T
Step 5. Open the R code of interest.

Using the OptiPOPd software:
The software contains a linear workflow, and is meant to be run in alphabetical sequence:

(A) The optimization algorithm (e.g. Korte & Vygen, 2018). Open the algorithm, specify the total number of initial animals, the percentage of models that you wish to collect, and the number of iterations of the algorithm. This code will automatically save several files. Do not change their names. NOTE: Each algorithm has a runtime of 10-15 hours (depending on your machine). To reproduce the results of our algorithm validation Northern Spotted Owl study (Hanley et al, *unpublished manuscript*), skip part (A), begin running the code at (B), and simply run through code B-L, hitting "run all" each time. This algorithm is available for all combinations of distance (e.g. Dennis et al., 2001), noise type, and initial condition.

(B) Assessing performance of the matrix elements: This code will read pre-saved files (either from the running of A or from the previously run sessions of A) to assess whether the algorithm captured the matrix elements. The code to assess performance of matrix elements is available for all distance types.

(C) Assessing performance of the superparameters: This code will read pre-saved files (either from the running of A or from the previously run sessions of A) to assess whether the algorithm captured the super parameters ("*superparameters"*; Hanley & Dennis, 2019). The code to assess performance of superparameters is available for all distance types.

(D) Paring down the models to the top 0.1%: The original algorithm saved the top 1% best candidate models, but perhaps you wish to see only the top 0.1% of candidate models. This code ranks each candidate model according to its distance and skims off the top best This code is available for all distance types.

(E) Boxplots for the to 1%: This code will read pre-saved files (either from the running of A or from the previously run sessions of A) to assess whether the algorithm captured the true value of

the initial conditions (for all stage abundances), the superparameters, or the true values of each matrix element. The code is available for all distance types.

(F) Boxplots for the to 0.1%: This code will read pre-saved files (either from the running of D or from the previously run sessions of D) to assess whether the algorithm captured the true value of the initial conditions (for all stage abundances), the superparameters, or the true values of each matrix element. The code is available for all distance types.

(G) Ordinary Least Squares Estimates. This code will read pre-saved files (either from the running of A or from the previously run sessions of A, or from D or from the previously run sessions of D) to find the ordinary least squares estimates for the superparameters and eigenvalues. The code is available for all distance types.

(H) Tuljapurkar Correction (Morris & Doak, 2002): This code will read pre-saved files (either from the running of A or from the previously run sessions of A, or from D or from the previously run sessions of D) to correct the algorithm-predicted eigenvalues using the Tuljapurkar correction. The code may be used for all candidate models, the top distances, and the top ten models. The code is available for all distance types.

(I) Boxplots for the Ordinary Least Squares and Conditional Least Squares method: This code will read pre-saved files (either from the running of A or from the previously run sessions of A, or from D or from the previously run sessions of D) to find eigenvalues using the ordinary least squares and conditional least squares methods. The code will plot the eigenvalues for each algorithm combination against the true eigenvalue. The code is available for all distance types.

(K) Trajectory Validation; Getting the juvenile and subadult trajectories: This code will read pre-saved files (either from the running of A or from the previously run sessions of A, or from D or from the previously run sessions of D) to find the time series of juveniles and adults for each candidate model. The code is available for initial conditions in stable status (SSD) for all distance types but may be used to attain trajectories for initial conditions out of stable by replacing everywhere in the code SSD with NSSD.

(L) Checking residuals: This code will read pre-saved files (either from the running of A or from the previously run sessions of A, or from D or from the previously run sessions of D) to check the residuals of the least squares methods. The code may be modified to use with any combination of distance measure, initial conditions, and noise type by modifying the top inputs for the file name of the combination of interest.

Important citations

Caswell, H. (2001). *Matrix population models: Construction, analysis, and interpretation*. 2nd edition. Sunderland: Sinauer Associates.

Dennis, B., Desharnais, R., Cushing, J., Hensen, S., & Costantino R. (2001). Estimating chaos and complex dynamics in an insect population. *Ecological Monographs*, **71**, 277-303.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., & Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, **1**, 1542-1552.

Fox. J. (2016) *Applied regression analysis and generalized linear models.* 2nd edition. Los Angeles: Sage Publications Inc.

Hanley, B., & Dennis, B (2019). Analytical expressions for the eigenvalues, demographic quantities, and extinction criteria arising from a three-stage wildlife population matrix. Natural Resource Modeling, **32**:2.

Klimko, L., & Nelson, P. (1978). On conditional least squares estimation for stochastic processes. *The Annals of Statistics*, **6**, 629-642.

Korte, B., & Vygen J. (2018). *Combinatorial Optimization: Theory and Algorithms.* 6th edition. New York: Springer Publishing.

Noon, B. & Biles, C. (1990). Mathematical demography of spotted owls in the Pacific Northwest. *Journal of Wildlife Management*, **54**, 18-27.

Technical Details of the Software

This software was written under R Version 3.5.3 (2019-03-11) – Copyright © 2019 The R Foundation for Statistical Computing, Inc. This code requires the R package: “MASS”,

Funding

Financial support was provided in part by the Morris Animal Foundation. This software has not been reviewed or endorsed by the Foundation, and the views expressed in this software do not necessarily reflect the views of the Foundation, its officers, directors, affiliates or agents.

License

CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Suggested Citation for this Software:

Hanley, B., Dennis, B., Kramer, D., & Schuler, K. 2019. OptiPOPd: Software to estimate elements of a population matrix model using adult time series data. Validation using simulated trajectories of Northern Spotted Owl [Software]. Cornell University Library eCommons Repository. https://doi.org/10.7298/a416-v747