Phenology Pollination Model Tutorial

Descriptive summary of the phenology pollination model

Paul Glaum [prglaum@ucdavis.edu](mailto:prglaum@ucdavis.edu) & Fernanda S. Valdovinos [fvaldovinos@ucdavis.edu](mailto:fvaldovinos@ucdavis.edu)

Tutorial

1.) Load all Matlab files and network templates into the working directory of Matlab.

Simulations are run and output is produced using the “run\_Phenology\_model” command. All other Matlab scripts will be called automatically if everything is properly loaded into the same working directory folder.

2.) The phenology pollinator model is built upon the preexisting modeling advancements of Valdovinos et al 2013 & Valdovinos et al 2016. Please see those publications for further detail into the underlying model:

* Valdovinos, F.S. et al (2013) Oikos, 122, 907-917
* Valdovinos, F.S. et al (2016) Ecology Letters, 19, 1277-1286

3.) The model is run on plant-pollinator networks input into the main function. The input network should be a text file of 1s and 0s identifying potential pollination interactions. Plants are enumerated along the rows of the networks and pollinators along the columns. Examples are available in our 3 fully connected networks included in this tutorial, but the model runs on other networks, empirical or otherwise. Use the following to load in network text file of your choosing. The example below loads our 30 plant by 50 pollinator fully connected network:

In=sparse(load('fullConnected-30x50.txt'));

4.)

Input parameters:

In: The text file of the network describing potential plant-pollinator interactions.

bloomSpan: The number of time steps in each flowering plant’s bloom period.

breakValue: The number of cycles between subsequent bloom periods.

flightSpan: The number of time steps in each animal pollinator’s flight period.

breakValueF: The number of cycles between subsequent flight periods.

5.) Input the following into the Matlab console after assigning input parameters:

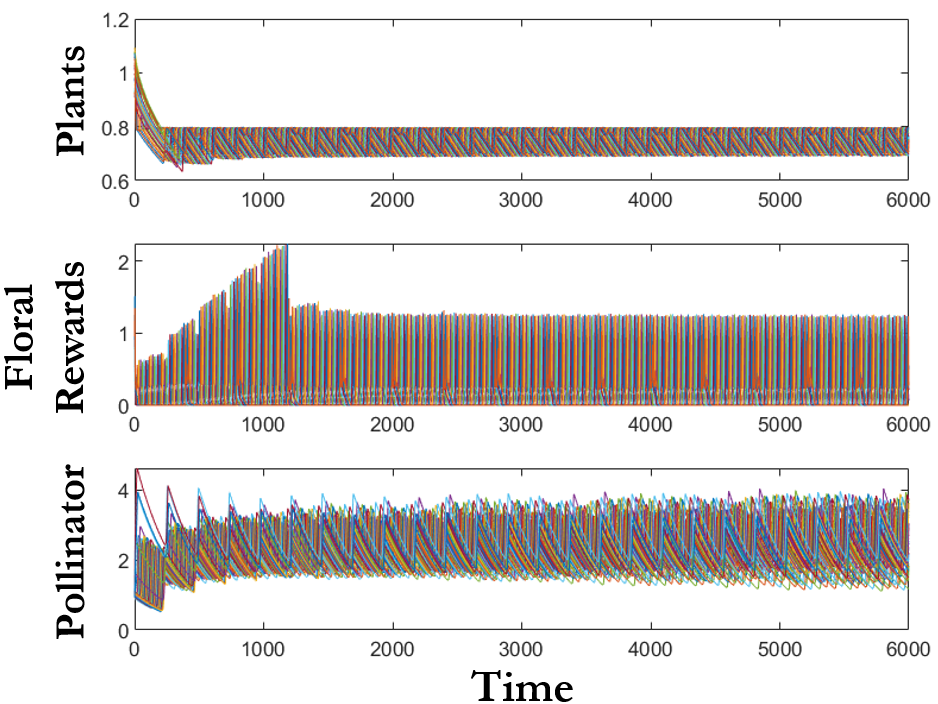
[pI, nectari, ai, alphasi, pf, nectarf, af, alphasf, nectar, avgAlphasf,indiePlantOverlap,overallPlantOverlap]=run\_Phenology\_model(In,bloomSpan,breakValue,flightSpan,breakValueF);

Run time can be somewhat extended, ~5-10 minutes, depending on system specs, phenology parameters, and network size.

6.) Below is a time series from an example run using:

In=sparse(load(‘fullConnected-30x50.txt’));

bloomSpan=50; breakValue=3; flightSpan=30; breakValueF=3;



7.)

Simulation Output:

pI = initial plant density (t=0; note, we don’t use pi to avoid confusion with in Matlab)

nectari = initial floral reward density (t=0)

ai = initial animal pollinator density (t=0)

alphasi = initial alpha values (t=0)

pf = final plant density (t=6000)

nectarf = final floral reward density (t=6000)

af = final animal pollinator density (t=6000)

alphasf = final alpha values (t=6000)

nectar = time series of floral reward

avgAlphasf = alpha values averaged over the last 1000 time steps

indiePlantOverlap = vector of individual plant resource overlap values

overallPlantOverlap = overall resource overlap summed up across all plant species

Additionally, a time series is produced of plant density, floral rewards, and pollinator density as shown above.

8.) To see specialization metrics on pollinator diet breadth use the DFG function with either the alphasf or avgAlphasf variable from a completed execution of run\_Phenology\_model. Use the full output command of DFG as follows:

[v\_dfg, v\_dfg\_st, v\_dfg\_Max, v\_SD, v\_CV] =DFG(alphasf)

The output reads as follows:

v\_dfg = the DFG score for each pollinator

v\_dfg\_st \*\*= the DFG score for each pollinator standardized to the number of plant species, range [0,1]

v\_dfg\_Max = the maximum pairwise difference between a single pollinator’s alpha values on different

plants

v\_SD = the standard deviation of the pairwise differences in alpha values

v\_CV \*\*= the coefficient of variance in the alpha values

\*\*Metrics presented in our analysis

9.) Simulation results data used to recreate Figure 2, Figure 3, and Figures S7-S10 is available at the public google drive at <https://tinyurl.com/y5sexjzs>

Note, the data is large, >3.5 Gb