Generating and Evaluating Initial 2000 PSC’s

1. Generate initial 2000 probiotic support communities (PSC’s) using one of three sets of already generated native communities (NC’s). Do this using script named: 19oct20_stode_single_killerComms_analysis_forInitialPSCGeneration.R
   1. Delineate the variable ‘file_number’ this will be the name of the directory where results/output are written to – Directory will be “~Documents/SynCom Modeling/[File Number]”
2. Open KillerCommMetrics_file_number_[file_number].csv containing aggregated results for PSC ensemble generated and simulated
   1. Located at “~Documents/SynCom Modeling/[File Number]”
   2. Open .csv file, and then save as in spreadsheet format prior to augmenting this file
      1. Preserve the original .csv file unchanged
3. Filter columns of spreadsheet, and rank rows in descending order based on values in column “V” – labeled as “# of NC whereby target species abundance > 0 in single target speices inoculation AND target species abundance > in PSC inoculation than in single target species inoculation” – rows are the ‘abundance’ score for each PSC
   1. The top row should be aggregate calculated values for the PSC with the highest abundance score.
4. Identify PSC identity numbers of the five PSC’s with the highest abundance score (top five rows of the filtered worksheet).
5. Transfer (copy paste) corresponding information about these five high-abundance performing PSC’s to evolution recording worksheet similar to “ 20oct20_set4eigenstable_redo” located in workbook named “20oct20_model_parameters_sensitivity_functionalEvolution.odt”
6. Run parameter sensitivity script for each of these PSC’s. This script operates on a single PSC from within a particular set. This script will, individually and singly, set each of the 80 parameters unique to the delineated PSC equal to zero, and assess the difference in survival and abundance score upon the loss of the single parameter.
   1. script name is “[Date]_parameterSensitivity_motifFinding_[NC’s used].R
      1. [Date] is the date of most recent script revision
      2. [NCs used] refers to the specific native communities the evolution being conducted originated from. Options are:
         1. eigenstable
         2. time stable
         3. unstable
            1. So far, we only have a working script for the eigenstable native communities. I will have to adapt this script for use with the other two sets of NC’s
   2. You’ll have to delineate the [file_number] (which is the set directory number) and [PSC_number] in the header of the script in order to delineate which PSC the script should analyze. After setting these two variables, run the script.
   3. Output will be stored in the same ‘set’ directory as the original set from which the PSC being analyzed came from. Output is a single .csv file named “ParameterSensitivity_PSC_[PSC number].csv”
7. For each PSC for which the parameter sensitivity script was run:
   1. Open the “ParameterSensitivity_PSC_[PSC number].csv” file located in its respective set directory (located at ~/Documents/SynCom Modeling/[set number]
   2. Import it into spreadsheet and save as separate file in spreadsheet format (.xlsx, .odt), preserving the originally generated .csv file unchanged
   3. Filter columns and sort the entire sheet by column K – sort ascending; the first row is now the row of data with the lowest value in column K.

4. Each row in this spreadsheet records data for the single and individual omission of one interaction parameter (i.e. resulting data after setting this parameter edge weight to zero and re-simulating with all other parameters held constant). Column K records the percentage drop in the abundance score due to the loss of the individual parameter.
5. Identify all parameters whose percentage drop is < 90% of original PSC abundance score when omitted (will be the top entries on this spreadsheet).
6. Transfer (copy paste) information about these parameters, and corresponding metrics about them, to worksheet similar to “ 20oct20_set4eigenstable_redo” located in workbook named “20oct20_model_parameters_sensitivity_functionalEvolution.odt”
7. Rejoice, you are now ready to begin functional network evolution using these five high performing PSC’s.

Simulating initial, and subsequent, rounds of functional evolution on species interactions

For each PSC to evolve:

1. Open script named “19oct20_stode_single_killerComms_analysis_forEvolutionRuns.R”
2. Save this script as a separate file, suggested file name should contain full example file name along with information about which round of evolution, the set number, and the PSC number being evolved. An example filename would be: “19oct20_stode_single_killerComms_analysis_forEvolutionRuns_1evolutionSet255psc1987.R”
3. Within this script, all interaction parameters are by default set to “rnorm(1,mean=param_mean,sd=param_sd)” – which is a command to sample a value from a normal distribution of a particular mean and standard deviation. For the parameters identified in the parameter sensitivity analysis performed above, replace this defalut with the actual parameter value. I find it useful to keep a log of the parameters I have changed within the script itself, commented out at the very top. Also at the very top of the script, I note which round of evolution I am currently on, and the PSC that the parameters were derived from.
4. After appropriate parameter values have been manually changed, run the script. This will generate a new set of 2000 PSC’s.
    1. Delineate a new file_number (set number) in the script unique from other sets run previously. This is done in the first few lines of the script that are actually executed.
5. After the script has run, open the ‘KillerComm Metrics_file_number_[file_number].csv containing aggregated results for PSC ensemble generated and simulated
    1. Located at “~Documents/SynCom Modeling/[File Number]”
    2. Open .csv file, and then save as in spreadsheet format prior to augmenting this file
        1. Do this to preserve the original ‘KillerComm Metrics….” .csv file unchanged
    3.

6. Manually calculate average and standard deviation of the abundance scores for all 2000 PSC’s (use =average() and =stdev() formulas within spreadsheet, abundance scores are located in column V
    1. This is the average abundance score for all 2000 PSC’s in this particular set of PSC’s
7. Record these mean and standard deviation values in a worksheet similar to “ 20oct20_set4eigenstable_redo” located in workbook named “20oct20_model_parameters_sensitivity_functionalEvolution.odt”
    1. Note in this worksheet that the evolution for each PSC has its own ‘section’ where summary metrics as well as cumulative parameters are cataloged and used in future rounds of evolution

8. Next, filter columns of spreadsheet, and rank rows in descending order based on values in column "V" – labeled as "# of NC whereby target species abundance > 0 in single target speices inoculation AND target species abundance > in PSC inoculation than in single target species inoculation" – rows are the 'abundance' score for each PSC
    1. The top row should be aggregate calculated values for the PSC with the highest abundance score.
9. Identify the PSC with the highest abundance score on this worksheet (PSC in the top row after filtering)
10. Run the parameter sensitivity script for this highest-abundance performing PSC. This is done exactly as in step 6 of "Generating and Evaluating Initial 2000 PSC's" above, except only run this script once, for the one PSC identified.
11. Repeat step 7 of "Generating and Evaluating Initial 2000 PSC's"
    1. Compile parameters in a cumulative fashion (e.g. include all parameters identified in all steps of the evolution and carry them forward into subsequent steps of evolution). Keep track of the 'cumulative parameters'
        1. In subsequent rounds of evolution, this cumulative list of parameters will be imported into the 19oct20_stode_single_killerComms_analysis_forEvolutionRuns.R script for the next round of evolution; thus 'growing' the number of parameters that are 'functionally' important to these communities ability to have a high abundance score.
12. Reojice again, you have finished the first round of functional parameter evolution for one PSC
13. Repeat steps 1-12 of "Simulating initial, and subsequent, rounds of functional evolution on species interactions."
    1. Continue repeating these steps iteratively, cumulatively adding to the list of functionally important parameters for each PSC being analyzed
    2. Once no 'new' parameters are identified (i.e. the 'parameter sensitivity' script has not identified any new parameters that havent already been identified in previous steps and are not already on the cumulative list of parameters for this PSC), cease evolution – you are done!

Needed Scripts and Example Files:
- script for generating initial 2000 PSC's:
19oct20_stode_single_killerComms_analysis_forInitialPSCGeneration.R

- script for evolution iterations: 19oct20_stode_single_killerComms_analysis_forEvolutionRuns.R

- script for parameter sensitivity: 21oct20_parameterSensitivity_motifFinding_eigenstable.R

-example worksheet for recording evolution information:
20oct20_model_parameters_sensitivity_functionalEvolution.ods

More notes:
- Worth noting that a completely analogous procedure could be done to 'evolve' network motifs that promote a high 'survival' score, in the same way we have evolved motifs for abundance scores above. The main difference would be the use of column U in place of column V in the ' KillerComm Metrics_file_number_[file_number].csv' spreadsheet (denoting the survival score of each individual PSC), and column I in place of column K in the "ParameterSensitivity_PSC_[PSC number].csv." I

have tried this, though was largely unable to achieve higher survival scores using this evolution protocol/algorithm.

- I generally perform evolution of five PSC’s at a time, in parallel. Some PSC’s will evolve for only a few iterations, while others can take many more (7-8) iterations to reach a state where no new parameters are being identified. I can typically run 5-10 instances of Rstudio, depending on what else I am trying to do on my computer.

- The “19oct20_stode_single_killerComms_analysis_……” scripts take several hours to run, generally 3-8 hours depending on how many instances are running in parallel

- The “21oct20_parameterSensitivity_motifFinding_eigenstable.R” script takes about 10 minutes to complete