

Running TREES using easyR_TREES and plotting results

Description of experiment: This exercise will start with an example we have previously used to introduce us to a fast way to run multiple simulations of TREES with varying parameter values. To do this we will use easyR_TREES which is a collection of functions to run TREES from R and to make common graphs from the results. The examples being used are the previously seen Chinese Cabbage and a new example using Wisconsin Fast Plant from a, currently unpublished, study comparing soil microbial treatments (Brock 2018 personal communications).

Running easyR_TREES: Open up easyR_TREES.R in RStudio and set your working directory to the folder you downloaded from github. TIP: use ctrl/cmd + enter to run lines in Rstudio.

Run_easR_TREES() is the main function to run TREES simulations.

1. Run the two lines that source the functions we will need. If your working directory is correct this should work. If any packages are missing they should automatically be installed here.
2. Look over the first Run_easyR_TREES() function and read the comments for each argument.
3. You should not need to change anything for this example except the N_cores argument. Make N_cores the number of cores you wish to use to run TREES simulations in parallel. (No more than 1-total number of cores)
4. Run this line and you should start seeing the TREES cout in the console.

Plotting results: The next function in the script is Leaf_Area_Plot() this script will plot the leaf areas for each leaf of each plant, under each new value, and for each driver specified. In this case it was only one driver so only one figure will be produced. Immediately after this function is one called Sim_Plot(). This function will plot any output specified from the .sim file generated by TREES.

1. Run Leaf_Area_Plot(). All except the last three arguments should match what was in Run_easyR_TREES(). The figures will be in the easyR_TREES/Outputs/Figures folder. Open and make sure they resemble figure A below.
2. Run Sim_Plot(). This is similar to Leaf_Area_Plot() except you have to specify a column from the .sim file to plot. Check the output matched figure B below and then try plotting a different sim output. See TREES-inputs&outputs-2020.xlsx for some ideas of what you can plot.
3. Change the Which_Sim argument in Sim_Plot() from LAI to PlantNstat and then to waterStress and RLA. You should have 5 graphs total.

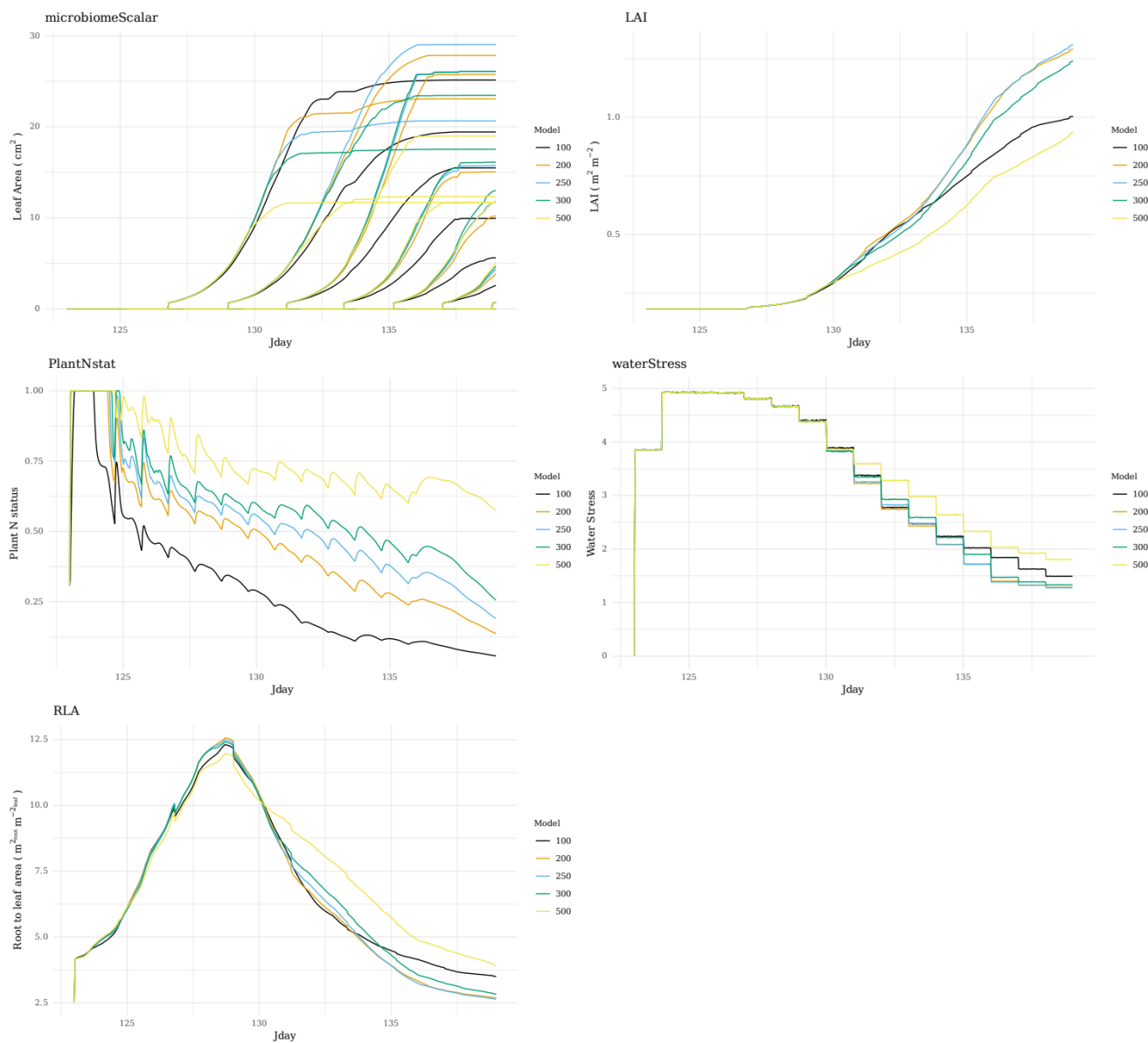


Figure 1: waterStress and PlantNstat are unitless and all other units are indicated in the y-axis.

Short discussion: Think about the following questions and we will discuss the answers together as a group.

1. Looking at the leaf area and LAI plots what kind of response does the model show to an increasing microbiome scalar?
2. Looking at the other diagnostic plots; why might this be?

Running with multiple drivers: Fast plant data was generated from 6 different blocks which resulted in 6 slightly different driver files. easyR_TREES can take in multiple driver files and run the parameter file with any new values over each of them. To do this a vector of the driver names (without extensions or paths) is passed to the Drivers argument.

Plotting results with and without experimental data: The experimental data this model is based on comes from a low microbial biomass so we will again adjust the microbiomeScalar but this time we will compare it to actual experimental data.

1. Run Leaf_Area_Plot like we did before and you will get one plot for each driver.
2. Now lets read in some experimental data and look at how it should be formatted by running the lines starting with Exp_data and ending with head(Exp_data)
3. Now we can plot with the experimental data and see which setting best fits the data.

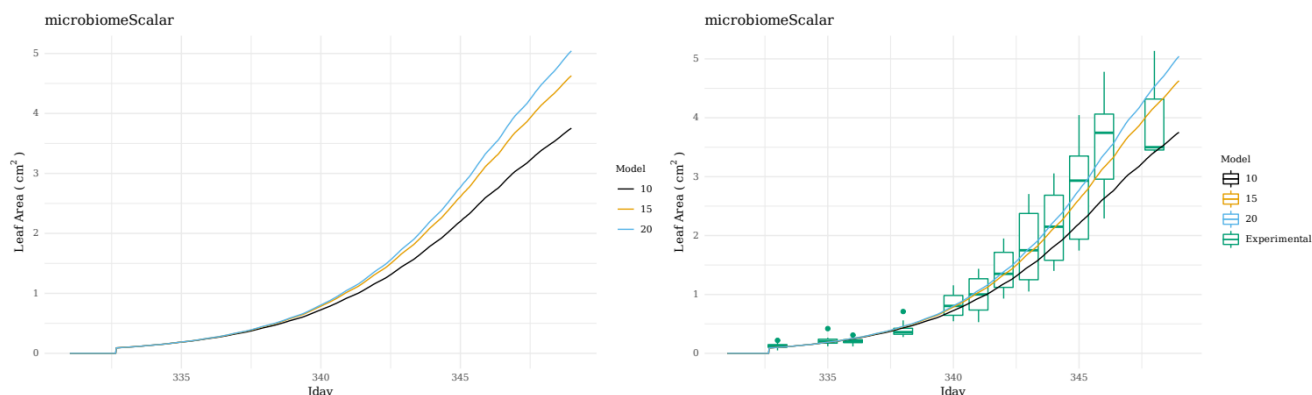


Figure 2: Left shows the model results only and right shows the results with experimental data in the form of boxplots. These are from only one of the drivers and note that running the same function again without changing the title may result in overwriting the figures.

Short discussion: Think about the following question and we will discuss the answers together as a group.

1. Even across a range of microbiome scalars there appears to be more variation than stochasticity in this parameter could account for. What other potentially stochastic parameters could be cause this?

Running with gamma on: The leaf growth parameters are a potential source of stochastic parameters so we can explore those by turning on the gamma leaf growth module. Note the number of iterations will be very unrealistic due to limitations of time. A realistic situation would call for running thousands of simulations to generate a distribution of results. Double check the N_cores is set appropriately for your computer and run the Run_easyR_TREES line. Notice Use_Gamma is TRUE and Iterr is a vector of 1 through 5.

Plotting results: The plotting functions are already designed to handle Gamma being on so running the plotting functions with Use_Gamma as TRUE will generate a plot of your distribution of results.

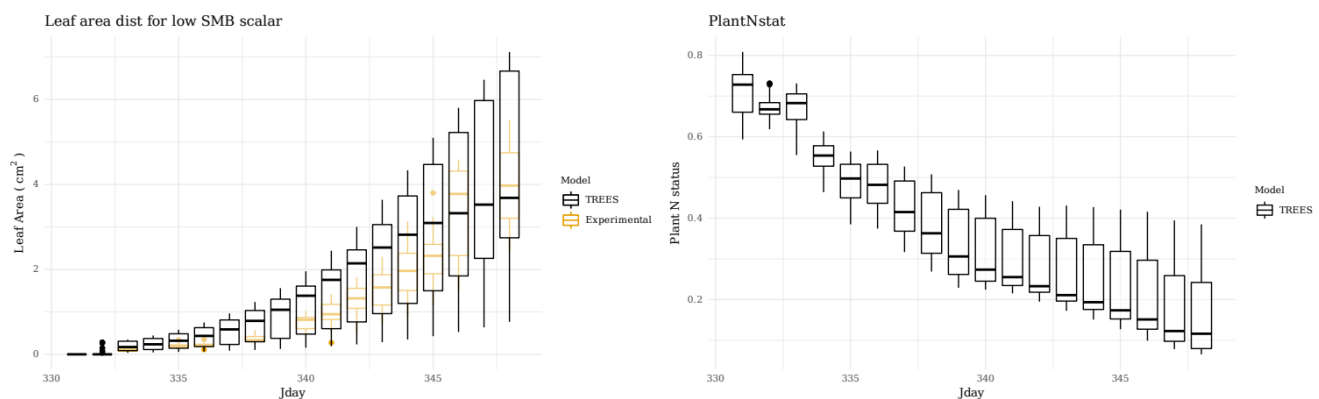


Figure 3: Left shows the distribution of leaf area and right shows the distribution for Plant N status for 5 iterations of TREES with the stochastic leaf module on.

Short discussion: Think about the following question and we will discuss the answers together as a group.

1. What other potentially stochastic parameters do you think could be driving the variation seen experimentally?
2. How would you track or constrain these parameters experimentally?