Requirements Specifications

# Functional Requirements (FR)

High Priority (HP):

1. Calculate dominant Plant Functional Type (PFT: Evergreen Needleleaf, Evergreen Broadleaf, Deciduous Needleleaf, Deciduous Broadleaf, Shrub, Grass, Cereal Crop, and Broadleaf Crop)for each tower site
2. Read in 4 input datasets (L4C reference dataset, L4C meteorological input dataset, Flux Tower Fluxes dataset, IGBP FAO Soil Organic Carbon Inventory dataset) for calibration
3. Compile the L4C meteorological inputs provided from NASA/NTSG and ground-truth data from flux tower sites into table
4. Allow the user to choose one of the nine PFTs (High-1)
5. Guide the user through removing outliers in average annual GPP and RECO calculations
6. Allow user to choose which parameters to use when optimizing GPP and RECO, such as LUE (light use efficiency) for GPP or faut (autotrophic/plant respiration fraction) for RECO
7. Calculate linear ramp functions given current BLPUT (High-6)
8. Allow user to specify number of Numerical Spin-Up iterations
9. Compute comprehensive validation and fit statistics: graph flux tower data against model-estimated data
10. Output updated BLPUT, 4 SOC stock-size maps, and SMRZ minimum and maximum for 2000-2018
11. Calculate flux tower weights as some tower sites are located in same 9km section

Medium Priority (MP):

1. Calculate percentage of area that is occupied for each PFT at each tower site
2. Compile historical data and error metrics for each day for each PFT into table
3. Subset time series variables to user configurable period (High-2)
4. Subset L4C meteorological and L4C reference input to sites that have dominant selected PFT (High-4)
5. Compile ancillary info on each site for selected PFT such as dominant PFT at 9km scale and SOC stock size for the site (High-4)
6. Compile all important data for calibration of selected PFT that includes FPAR, PAR, TSURF, TSOIL, SMSF, VPD, TMIN, SMRZ, minimum SMRZ, maximum SMRZ, and PAW (High-4)
7. Average data for PFT each day of the year (High-5)
8. Display current ramp functions to the user and allow them to save the plots as files (High-7)
9. Allow the user to plot GPP against Emult separately form the ramp functions (Medium-8)
10. Report the differences between new and old value parameters after optimization (High-7)
11. Allow the user to specify Pk and Prh for RECO (High-6)
12. Create 2 plots: Rh/Cbar against TSOIL with the Arrhenius curve on top, Rh/Cbar against SMSF with the SMSF ramp function on top (High-7)
13. Calculate Cbar for each tower site after optimization (High-7, Medium-12)
14. Calculate σ and Bsoc for each tower site to plot σ \* Bsoc against ground truth SOC sizes
15. Run the Analytical and Numerical Model Spin-Ups (High-8)
16. Run preliminary spin up arbitrary period over full operational record 2000-2019
17. Run L4C soil model forward runs
18. Prepare vectors of initial optimized parameters and GPP and RECO optimization objective functions to iteratively change optimization parameters
19. Calculate GPP\*, Kmult\*, NPP\* based off of calculations (Medium-7)
20. Calculate RMSE for each subset of towers for observed both GPP and RECO against L4C Reference Dataset, optimized GPP/RECO, and L4C Forward Model Run (High-9)
21. Calculate NEE (NEE= RECO – GPP) based on optimized parameters then report stats of NEE against L4C Reference Dataset, optimized GPP/RECO, and L4C Forward Model Run

Low Priority (LP):

1. Remove negative values in annual GPP and RECO for each flux tower site (Medium-7)
2. Calculate lower and upper limits for APAR (Medium-8)
3. Allow the user to repeat optimizations after the initial optimization (High-6, Medium-10)
4. Allow the user to plot GPP against Emult and Rh/Cbar against Kmult

# Non-functional Requirements (NF)

High Priority (HP):

1. Have accuracy of calibration that meets or exceeds the current implementation.
2. Be able to process data while maintaining the size being used in the current implementation.
3. Pass a holistic test suite (a script that individually and automatically tests the functionality of the entire project, such as Python’s built-in “unittest”).
4. Be subjectively rated better than the current implementation by the client.
5. Follow a modular, object-oriented design pattern.
6. Have basic documentation for all features.

Medium Priority (MP):

1. Have well-written inline documentation for every class, method, and variable (the code itself).
2. Allow the process, as a whole, to run 20% faster than the current implementation.
3. Use 30% less total, saved memory than the current implementation.
4. Make use of Python virtual environments.
5. Have a test suite for every component.
6. Have sufficient maintainability as determined by the client.

Low Priority (LP):

1. Have a test case written for every method.
2. Have a test suite with 95% code coverage or higher.
3. Be validated on systems other than those specified in the software requirements.
4. Be validated on browsers other than those specified in the software requirements.

# Hardware Requirements (HW)

1. Program to be developed on x86\_64 architecture with SSE4 extensions or later.

# Software Requirements (SW)

1. The Software shall be written in Python 3.
2. The Software shall run on Unix and GNU/Linux based systems.