Manual *v1.0.2*

SapFlower

Automating sap flow data preprocessing, cleaning, modeling, gap-filling, and analysis.

Standalone app available: https://doi.org/10.5281/zenodo.13665919

Source code: https://github.com/JiaxinWang123/SapFlower

SapFlower 1.0.2 Manual

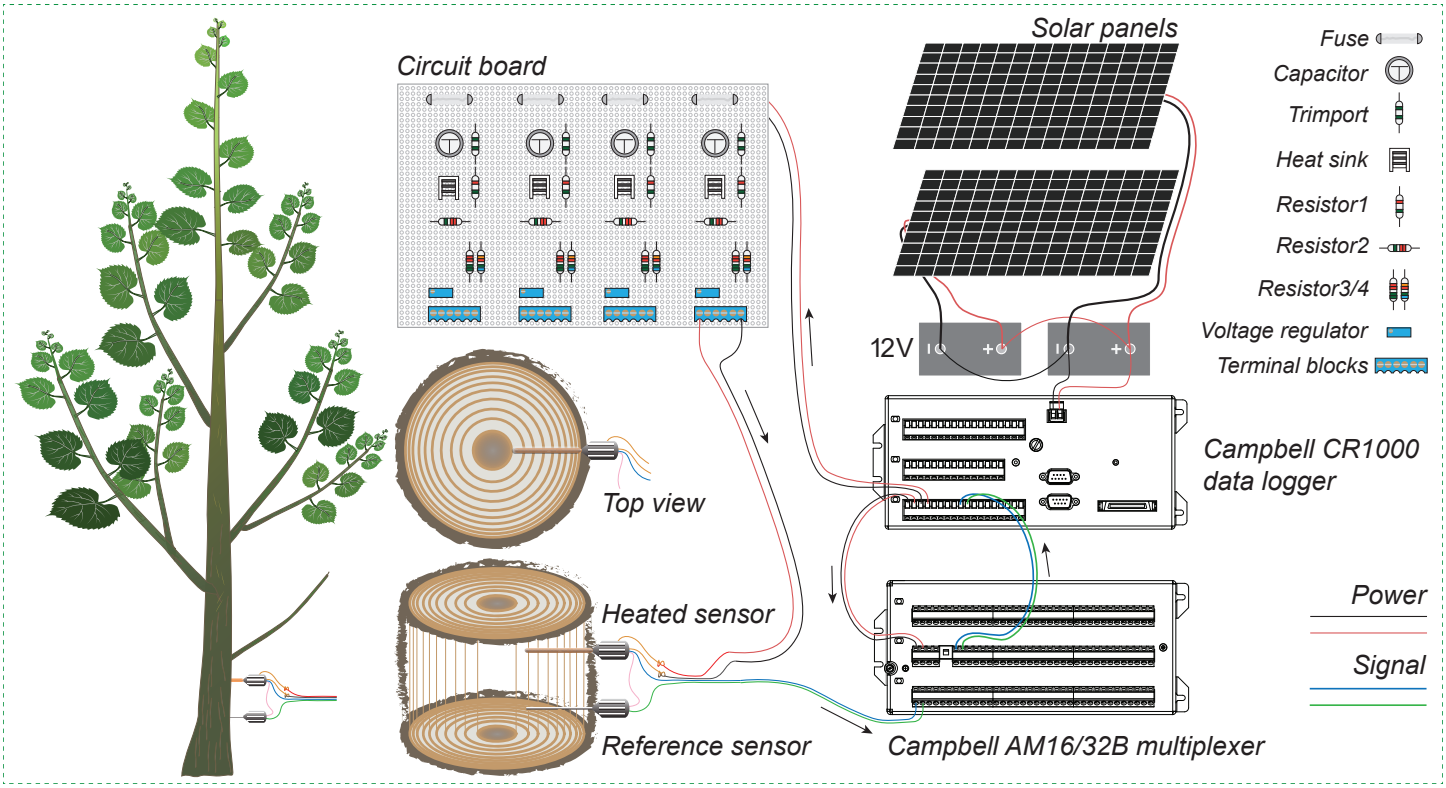
Jiaxin Wang

Department of Forest Resources and Environmental Conservation, Virginia Tech, Blacksburg, VA 24061, USA

For questions and requests, please email:
jiaxinwang362@gmail.com;
jiaxin.wang@vt.edu

Unshackle the hands of scientists from the drudgery of mundane tasks, that they might better weave the tapestry of scientific discovery.

Schematic diagram of configuration and installation of thermal dissipation probes.



1. What is SapFlower?

SapFlower is a tool developed for ecologists and plant scientists who are interested in measuring plant sap flow using Thermal Dissipation Probes (TDPs) based on Granier's equation (Granier 1987). Considering measuring sap flow in the field using solar panels powered sap flow sensors may encounter power outage due to the weather conditions; gap-filling is essential for getting complete data to estimate whole growing season water use. SapFlower was designed to automate sap flow data cleaning, filtering, segmentation, gap-fill model training, gap-filling, and sapwood and water use analysis. Users can feed SapFlower raw data measured from the datalogger, and get complete gap-filled data including dT, K, and F for calculating sap flow. Additionally, users can covert, process, and gap-fill SAPFLUXNET data in SapFlower.

SapFlower integrates some essential functions such as baseline calculation from the Baseliner 4.

2. How does SapFlower work for raw sap flow measurements?

SapFlower has four main parts, namely Project Configuration, Data Preprocessing, Model Training, and Gap-filling, which enable users to handle all sap flow data processing and calculations in one place.

2.1 Project Configuration

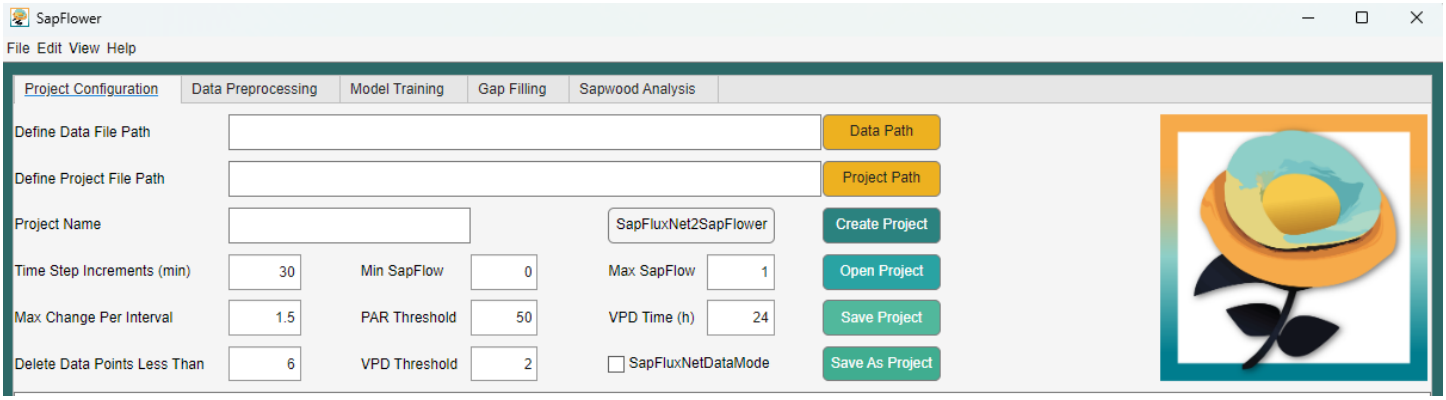
Users can create, open, set up parameters for data filtering, and save projects easily through the Project Configuration Tab.

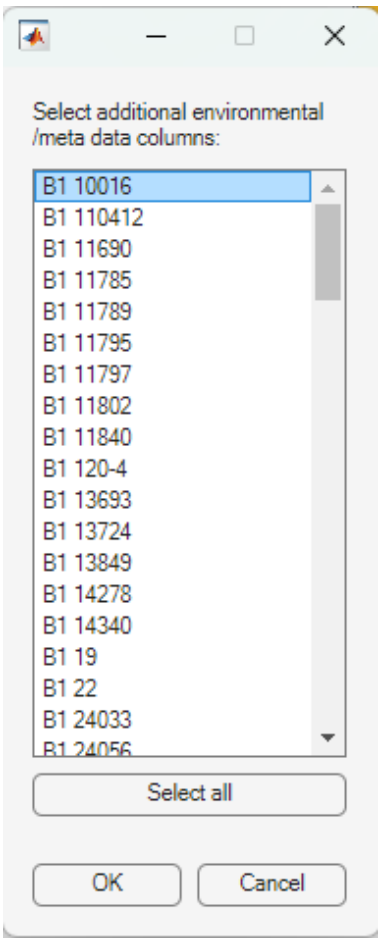
2.1.1 Create a new project

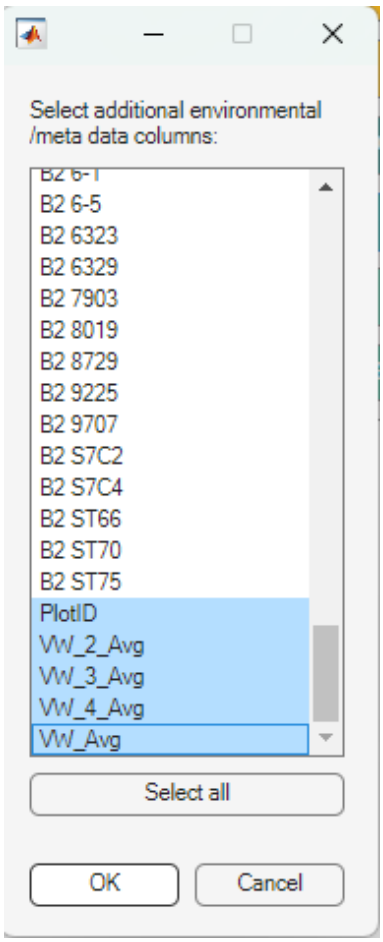
Before creating a project, users can edit the data filtering thresholds for the project. The default values are set to the Toy dataset, users should change them as needed based on their own data.

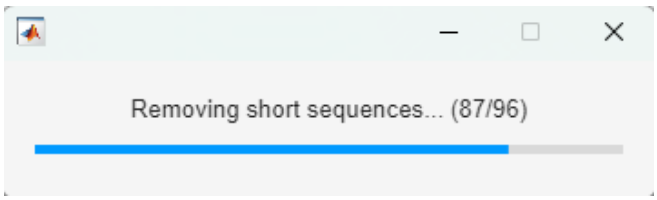
Field name	Description
Define Data File Path	Users can either type or click the "Data Path" button to prompt wot select a data file to edit. The file can be either .csv or .xlsx format, and the file must include some columns for calculation. The mandatory columns are TIMESTAMP, PAR_Den_Avg, AirTC_Avg, RH, and at least one sap flow data column. (Future updates may dynamically handle different users' data structure). The columns must be the same as listed here for indexing. Users can also have other columns, such as LAI (leaf area index), DOY, VPD, and it will calculate them based on TIMESTAMP for DOY, and AirTC_Avg and RH for VPD.
Define Project File Path	Users don't need to define through this edit field, instead, it will prompt to allow user to select a folder to save the project file.
Time Step Increments (min)	Time between two data points. 30 = 30 minutes.
Min SapFlow	Filters data that are below acceptable range. Any dT data less than this value will be deleted. Just set it to zero if you don't know.
Max SapFlow	Filters data that are above acceptable range. Any dT data greater than this value will be deleted. You can first set to a high value to view your data and then come back to set it to a reasonable value for your data.
Max Change Per Interval	Filters data spikes. Any value with an absolute change from the previous time step that is greater than this value will be deleted. You can first set to a high value to view your data and then come back to set it to a reasonable value for your data.
PAR Threshold	Threshold for determining nighttime hours. Note that due to PAR/radiation sensor calibration/drift issues or light pollution, sensors may read some nominal value greater than zero during the night.
VPD Time (h)	Time threshold for low-VPD conditions below which transpiration is expected to cease.

Field name	Description
Delete Data Points Less Than	Filters short runs. Due to the outage of battery or low battery, there will be some short runs of measurement, such as only a few of points measured during the noon. You can set it to zero to preview your data first and then come back to change it to a reasonable value.
VPD Threshold	I would keep this as intact described in Baseliner: Threshold for low-VPD conditions below which transpiration is expected to cease. Note that due to temperature/RH sensor calibration/drift issues, sensors may read some nominal value greater than zero during saturated conditions.









AirTC_Avg	BattV_Avg	DOY
23.3800	13.6600	141
23.0000	14.1300	141
23.3400	13.8800	141
23.7100	13.3500	141
23.4000	13.3300	141
23.5300	13.3400	141
23.6000	13.3400	141
23.5100	13.3700	141
22.9700	13.3700	141
22.5300	13.4200	141
22.6900	13.4600	141
23.1000	13.4700	141
23.2400	13.4700	141

Users can define the data path first, add a name in “Project Name” edit field, and then either use the menu, shortcut (Ctrl+N), or “Create Project” button to prompt to create a project file (.html). After creating a project file, it will save it and open it as the current project file. It will prompt you to select additional environmental or meta data columns. You should select all columns that are not sap flow sensors, so that later you can loop through all your sensors data at once. It may also ask you to select which one is Air temperature and which one is relative humidity if your data doesn’t have a VPD column.

2.1.2 Open an existing project file

Users can easily open an existing project file by clicking “Open Project” either using the menu or button or using shortcut (Ctrl+O). It will prompt you to select additional environmental or meta data columns. You should select all columns that are not sap flow sensors, so that later you can loop through all your sensors data at once. It may also ask you to select which one is Air temperature and which one is relative humidity if your data doesn’t have a VPD column.

2.1.3 Saving a project file

Users can modify their currently loaded project file and save it as new projects or just update some parameters by clicking “Save Project” either using the menu or button or using shortcut (Ctrl+S).

2.2 Data Preprocessing

This panel is designed to help users to clean their data, and the main design was inspired by Baseliner 4. There are two main options, namely manually clean and auto clean. Manually clean requires users to determine valid and effective data through setting filtering thresholds in Project Configuration and manually selecting and deleting or reversing data points in Data Preprocessing. During data cleaning, if users want to and changed update project configuration, they can click “Plot Data” button to view the updated data based on the new project configuration.
Auto clean requires users to set the auto clean parameters in Model Training left top (WindowSize, IQRThresholdMultiplier, HighVariationThreshold, and LowVariationThreshold). Users can use “Auto Clean” to clean and view the data points that will be removed based on currently defined thresholds. Please note that the Auto Clean will always be applied before all model training to ensure that at least some outliers are removed. It’s better if users can use auto clean and manually check and clean some data and fine-tune the auto clean thresholds to get high quality training data.

Users can also use shortcuts for data deletion (Ctrl+D), reversal (Ctrl+R), and undo (Ctrl+Z). To switch different sensors, users can select the sensors from the dropdown menu (Y) or just loop through sensors by clicking “Previous Sensor” or “Next Sensor.”
Users can also edit the current plot by clicking and editing the title, axes, and legend, or export plot by moving the mouse to the right corner of the plot to use the export plot tool. The plot

can be exported in .JPG, .PNG, .TIF, and .PDF format. Users may see that there are some data smoothing functions when right click on the plot, but please note that those haven't been enabled to use. Future updates may enable those functions.

All executing output will be updated in the bottom text area, and users can use it as references when they encounter any unexpected errors or warnings.

Once users finished cleaning, they can go to the Model Training Tab without saving the edited data since it will be updated automatically to the current data table. However, if users want to save the cleaned data, they can then click "Save Data" to save the cleaned data and export the current K values. Alternatively, users can save to replace their original data or save as a new data file by clicking "Finish Editing." The time for saving data depends on the data that users loaded, so please wait until it finishes saving.

Parameter	Description
WindowSize	It separates the whole data sequence into multiple short sequences to calculate the quantile, low and high variation. You can set it based on your time step increments. If your time step increments are 30 min, then you can put 72 to represent 36 hours window. Smaller numbers will make it more sensitive.
IQRThresholdMultiplier	Multiplier for IQR to define outliers (Recommend setting it based on your Max SapFlow)
HighVariationThreshold	Threshold for detecting high variation windows. A smaller number will filter more points. (Recommend setting it based on your Max SapFlow)
LowVariationThreshold	Threshold for detecting low variation windows. A larger number will filter more points. (Set a small number like 0.01~0.05 based on your data)

2.3 Model Training

This Tab is designed to help users train their own models for sap flow data gap-filling. Before model training, users need to set their output data path for cleaned data, trained model, and predicted and gap-filled data. Users just need to set one path; it will set sub folders to store different types of data.

There are several steps that users should confirm before clicking the "Start Training" button, and the "Start Prediction" button will only work after model training.

Step 1. Set output path

Users need to ensure that their data is cleaned/or set up reasonable auto clean parameters for auto cleaning.

Step 2. Set parameters for Auto Clean

Define effective growth period through the Date pickers. The gap-filled output data will use them as references for calculations.

Step 3. Model Selection

Users may select/check at least one and only one model for gap-filling model training. Please note that Time Series models (ARX and ARMAX) require Y values as reference for prediction and can only be used for short period (e.g., 24 hours) predictions. If users have more over 24 hours missing period, I would recommend using Recurrent Neural Networks or Random Forest models.

SapFlower currently supports Linear, non-linear, machine learning, and deep learning models. Specifically, it integrates: Simple Linear, Multiple Linear, ARX, ARMAX, GRU, LSTM, BiLSTM, RandomForest, Support Vector Regression, Gaussian Process Regression, and Kernel Regression. Considering sap flow can be affected by environmental factors, especially VPD, and seasonal variation, I would recommend including DOY, VPD, and PAR in your model for more reasonable predictions. Machine learning models like RandomForest, Support Vector Regression, Gaussian Process Regression, and Kernel Regression are more efficient when dealing with large but consistent dataset. Deep learning neural networks such as GRU, LSTM, and BiLSTM are more powerful, but training parameters such as data spliting rate, training epoch (iteration or how many times should the model be trained), learning rate, and hiddenUnit (hidden layers) should be experimentally set for better performance.

Step 4. Variable Selection

In the current version, users only need to select the Predicting Variables for model training. The variables will be listed under Predicting Variable(s) Tree once the users loaded their project. Users can decide to check one or multiple variables for different models of training. Here VPD is already checked as default considering their substantial relationship with sap flow. Users may leave the Response Variable(s) as default, and they can export K and F values once they finished gap-filling.

Step 5. Sensor(s) to be trained

Sensors will be available for checking once users loaded their project. Users may choose multiple sensors at once to loop through and train models for all checked sensors.

Step 6. Model training hyper parameters

Parameter	Description
SplitForValidation (%)	This is where users can set their data splitting for model training and validation. A number like 20 represents that 20% data will be split to be used for model validation. Users can try different numbers to see how it will affect the model training.
EpochForTraining	Users can determine the numbers of iterations for model training, and this will be only applied to Recurrent Neural Networks. A larger number may get more well trained or over trained model depending on your data and variables selected for model training. You can leave it as default if you don't know how to set.
GradientThreshold	Here it is used to prevent gradient explosion: In deep networks, especially recurrent networks, gradients can sometimes grow uncontrollably large, leading to instability. Clipping helps keep them in check. Improved stability: It improves training stability by ensuring that excessively large gradient updates don't destabilize the optimization process.
InitialLearningRate	The learning rate determines the size of the steps the optimization algorithm takes to minimize the loss function during training. If the learning rate is too high, the model may overshoot the minimum of the loss function, leading to instability. If the learning rate is too low, training may be very slow or get stuck in a local minimum.
ValidationFrequency	This parameter is used for validation. If you set it to one, it will validate immediately after each iteration of training, and if you set it to 5, it will validate every 5 iterations of training.
NumberOfHiddenUnits	The hidden units represent the neurons in the hidden layer that learn internal representations of the input data. The number of hidden units influences the model's ability to capture complex patterns in the data. More hidden units can increase the capacity of the model to learn more intricate relationships, but it also increases the computational complexity and the risk of overfitting if the dataset is small.
SolverForNeuralNetwork	It refers to the optimization algorithm (solver) used to minimize the loss function and update the model weights during training. The solver plays a critical role in determining how the neural network learns and converges. 'adam' (Adaptive Moment Estimation): This is a popular solver that combines the benefits of RMSProp and momentum. It adjusts the learning rate for each parameter and can perform well in many scenarios, especially for deep networks and LSTM networks. It's often a good starting choice for deep learning models because of its adaptive learning rate feature. 'sgdm' (Stochastic Gradient Descent with Momentum): This is the standard stochastic gradient descent method with momentum to accelerate convergence and reduce oscillations. It's effective for many types of neural networks, including CNNs and fully connected networks. 'rmsprop' (Root Mean Square Propagation): This solver is designed to deal with the diminishing learning rates problem in the traditional stochastic gradient descent method by normalizing the gradients based on their recent magnitudes. It is especially useful for recurrent neural networks and deep networks.

Step 7. Start Training.

Once users clicked to start the training process, the data will be auto cleaned and saved to CleanedData folder, and then the model training process will start. Users will be able to monitor the model training process either via the text output or real-time plotting (due to MATLAB license limit, the real-time plotting will be only available for those who are using SapFlower in MATLAB). Trained models will be saved to TrainedModels folder.

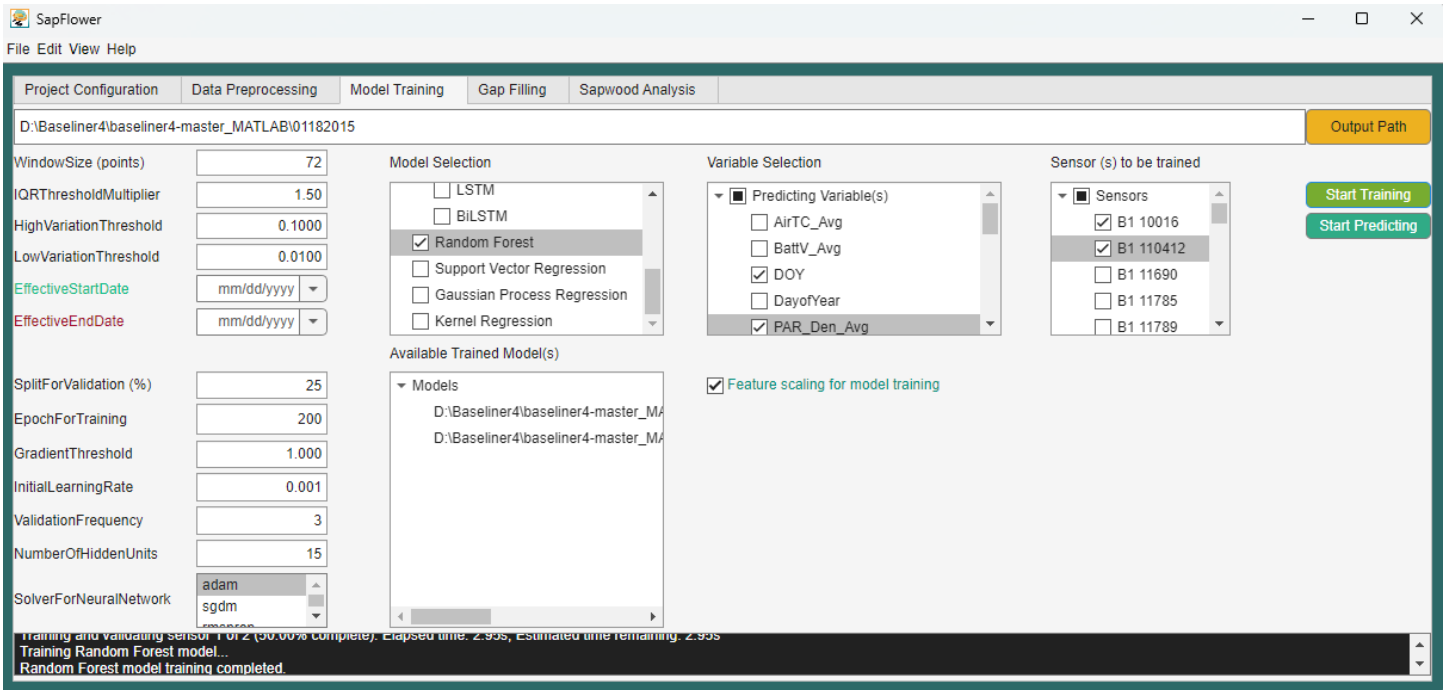
Step 8. Start Predicting.

Once users finish model training, they can start model predicting, and the predicted data for each sensor with model types will be saved to PredictedData folder.

2.4 Gap-filling

This Tab is designed to help users do and visualize gap-filling. Once users finish model prediction, the sensors available for gap-filling will be listed under the right list tree. To do all operations, users must check at least one sensor to be gap-filled. Users can start loading and view cleaned raw data in both table and figure by clicking the "RawData" button. Similarly, users can click the View Predicted Data Tab to view the predicted data plot by clicking "PredictedData" button. After loading both raw data and predicted data, users can then do gap-fill by clicking the "GapFill" button. Gap-filled data will be saved to Gapilled folder. Users can also edit, zoom in/out, and export all plots through using the plot's right corner's plotting tool bar.

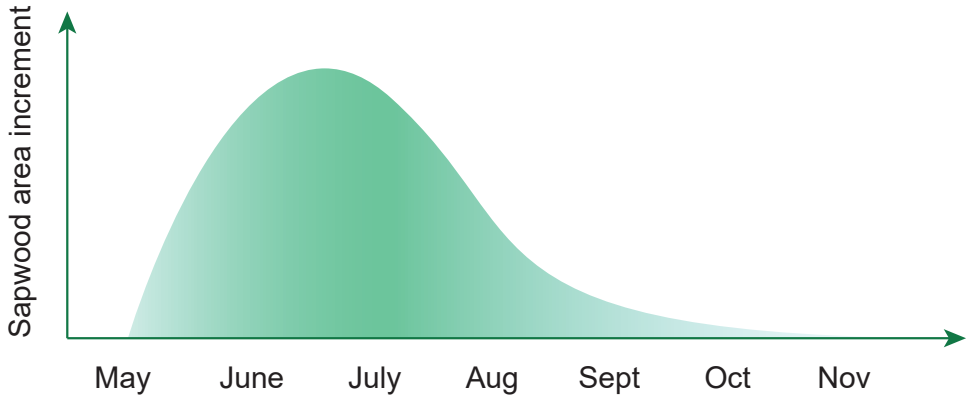
Once users finished gap-filling, they can export K and/or F values through clicking "ExportK-values" and/or "ExportFvalues", and the exported data will be saved to ExportedKvalue and/or ExportedFvalue folder. Users can then do further analysis using that data.



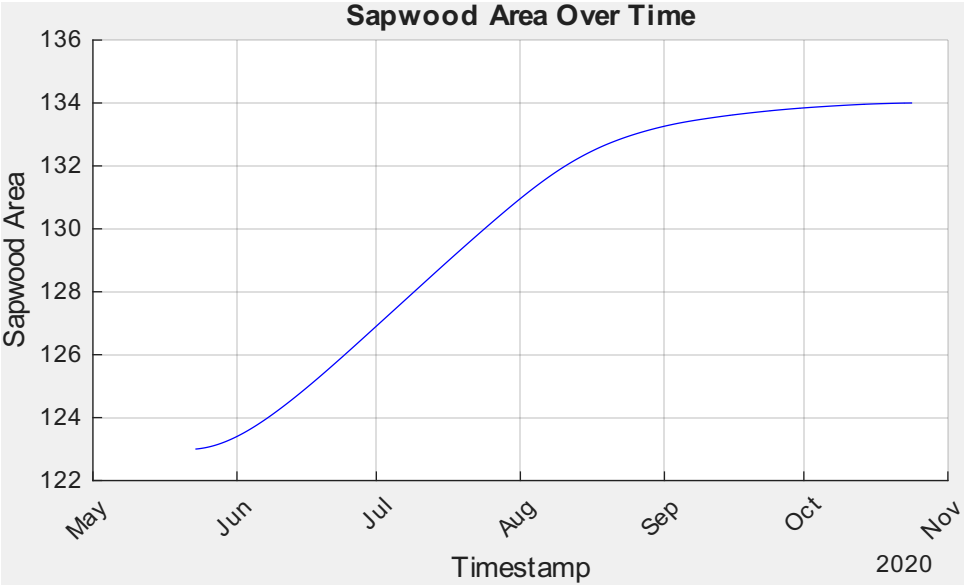
3. How to use SapFlower to scale and model sapwood area?

3.1 For two-point measurements of sapwood area

Considering the variation of increment of sapwood growth across the growing season and the effects of environmental changes on sapwood growth, we proposed weights-based sapwood growth modeling methods. Specifically, we assumed that the horizontal growth of sapwood is non-linear. For example, based on field experiments, we observed that poplar trees grow more sapwood from May to July, and then the growth rate slows down.



Example of sapwood increment of poplars across the growing season.



Example of sapwood area of poplars across the growing season.

Users can either define their sapwood increment rate/weights by pasting their measurements (e.g., measured time-series data of sapwood area increment across the growing season) or draw the weights curve to generate the weights using tools provided in Sapwood Analysis tab. Additionally, users can model their sapwood area across the growing season, if they have at least 6 measurements across the study period.

3.1.1 Paste your measurements of sapwood increment

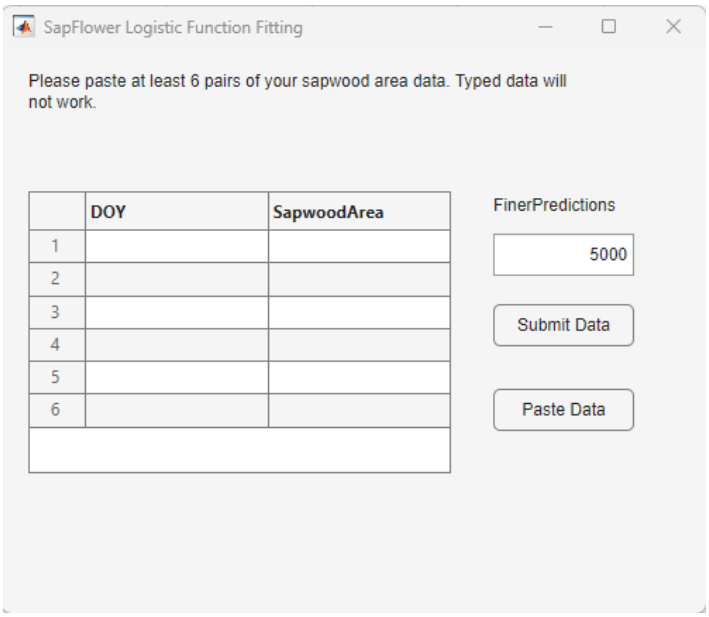
If you have your fine resolution sapwood increment measurements across your study period ready, you can use the paste function. To paste data to the table in Sapwood Analysis tab, users can prepare their measurements in Excel, copy TIMESTAM and Weights/Sapwood increment values (Please do not include the headers) and paste using the Paste button provided in Sapwood Analysis tab. Please note that if you copy a large data set, it will take longer to plot and load. So, please be patient. Then, users should enter the start sapwood area and end sapwood area in StartSapWoodArea and EndSapWoodArea edit field. Once, users define their sapwood area of start and end, they can calculate and plot the scaled sapwood area across the timestamp by clicking Calculate&Plot button.

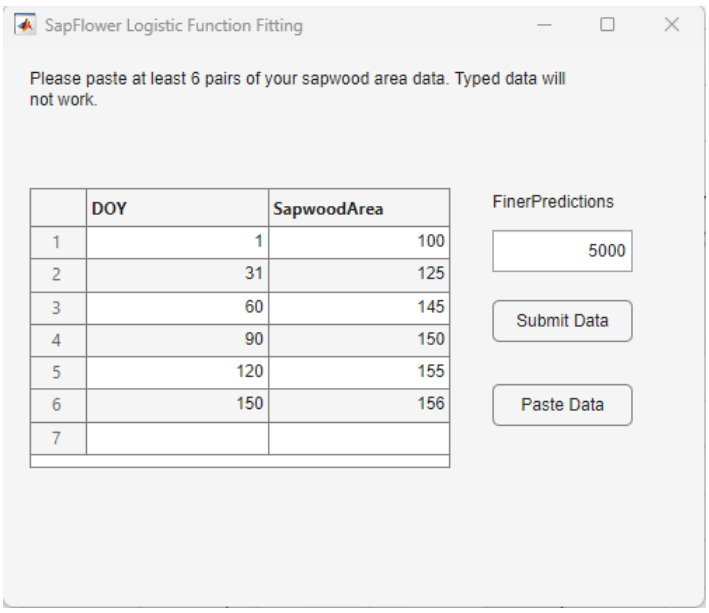
3.1.2 Model and scale your sapwood

If you don't have many measurements of sapwood increment across the growing season, but you have equal or more than 6 measurements of sapwood area/DBH, you can use logistic regression to model and scale your sapwood area data across your study period by using ScaleSapwood function. This part requires you define your BaseDate (the date you started measuring your sapwood area). Users will be required to define Tree/sensor Name (e.g., B1_13849) and BaseDate. If you use DOY, then your BaseDate will be 1/1/2025 for given year 2025. If you Day of the measurement, then your BaseDate will be the date that you started measurement, e.g., June 1st, 2025.

Case 1, Base Date: 1/1/2025		Case 2, BaseDate:6/1/2025	
DOY	SapwoodArea	Day	SapwoodArea
143	100	1	100
173	125	31	125
202	145	60	145
232	150	90	150
262	155	120	155
292	156	150	156

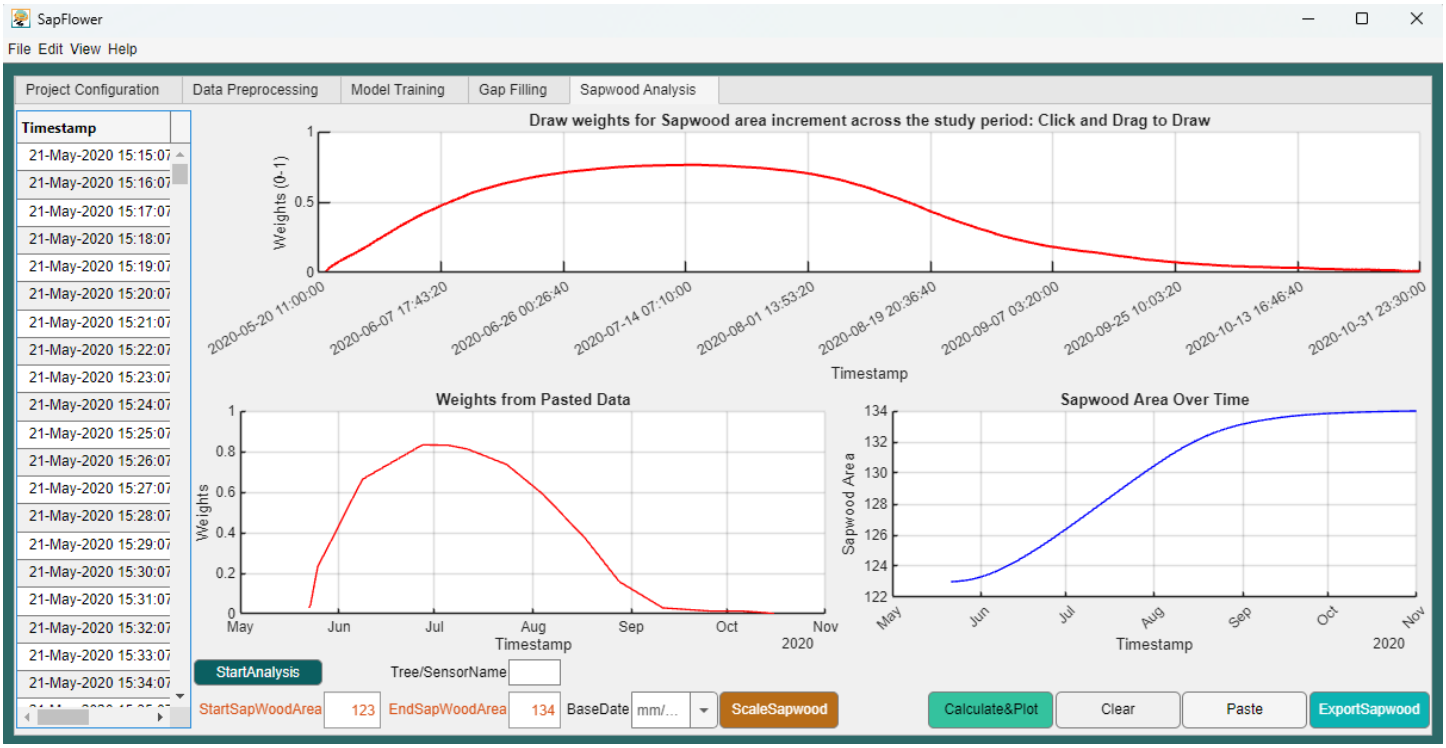
Copy the data (no headers) and paste using Paste Data button in SapFlower Logistic Function Fitting page. Then users may define the FinerPredictions, the larger the finer resolution of the predictions (e.g., if users want to get predictions for every min or second, they should increase the number based on their study period). Once, define the number of predictions, users can click Submit Data button, and the model will be fitted, and they will be asked to provide a path for predictions to be saved.





3.1.3 Draw sapwood increment weights

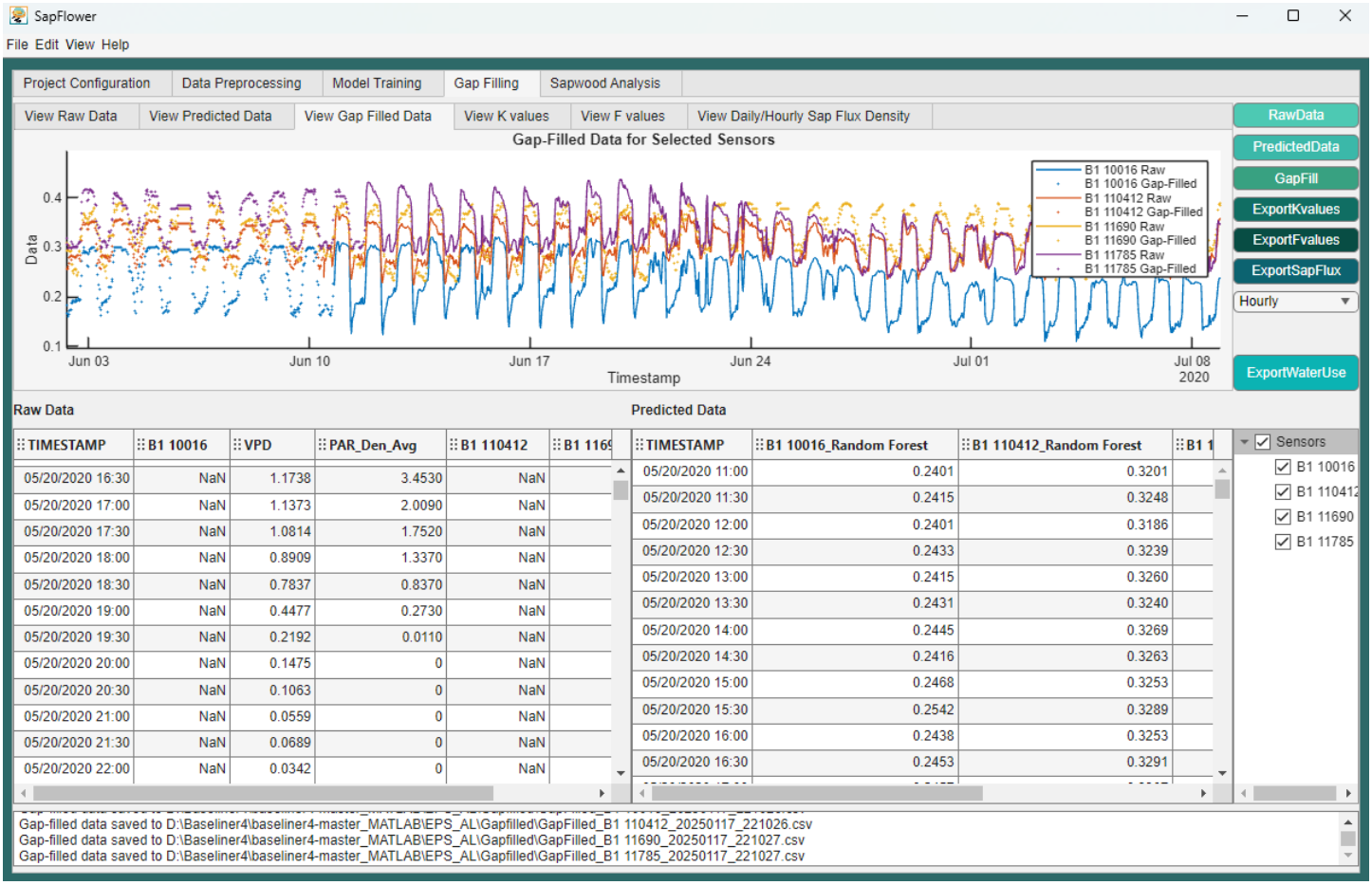
If you only have sapwood measurements from the beginning and the end of your study period, but you know the general growing pattern, or you have environmental data can infer the general pattern of sapwood increment across the study period, then you can use the weights curve tools. First, you need to create and load your sap flow measurement to get your TIMESTAMP from the project tab. Alternatively, if you don't have your data loaded, you will be asked to give a csv file that has TIMESTAMP in it. Then you need to click StartAnalysis button in Sapwood Analysis tab to initialize the module. Then, you can define the SartSapWoodArea and EndSapWoodArea in the edit fields, define the Tree/Sensor Name, draw the pattern curve on the top UI Axis, and click Calculate&Plot button to scale and visualize the sapwood data.



4. Water use analysis

Users can calculate total water use using their analyzed sap flow data and sapwood area data in SapFlower. Specifically, before gap-filling and water use analysis, users should have their sapwood area ready. For example, users need to model or scale their sapwood area data at their sap flow time scale and resolution (or finer resolution). If users' raw sap flow data is measured every 30 min, then users' sapwood area data should be the same (30 min interval) or finer scale (less than 30 min).

Once users gap-filled and exported their sap flux density (F) data, they can load both exported F values data and sapwood area data to do water use analysis. Specifically, SapFlower will use KNNSEARCH to find and match the nearest timestamp to match sap flow and sapwood area data, all csv files in user defined sapwood area file folder will be iterated to find and match all sensors/trees for sap flux data. If multiple sap flux data csv files are in one folder, it will then iterate all and analyze them. In the circumstance that different sensors/trees might have different time scale/range, SapFlower will then save the output for each sensor/tree individually.



5. Work with SapFluxNet data

Considering SapFluxNet has data files in different formats and naming conventions, we need to transform SapFluxNet data (e.g., csv files) into SapFlower accepted data files. Now, SapFlower provides a helper to enable users convert SapFluxNet data to SapFlower data in a high throughput way. When users opened SapFlower, they can click SapFluxNet2SapFlower button in Project Configuration tab. It will ask users to define the path for SapFluxNet csv data, specifically, the path that directly contains csv files. Once users define the path, it will process the data, and users will be able to see the progress.

Once data transform finish, users can load their transformed data to SapFlower as normal, but they must define all parameters, especially Max SapFlow and Max Change Per Interval parameters, in the project configuration tab to ensure that their sap flow data won't be filtered, since SapFluxNet data are calculated sap flow rather than raw measurements (they range from zero to over thousand).

Users may be able to visualize their SapFluxNet data and do modeling and gap-filling as normal, but will not be able to calculate K and F. Please note that it is suggested that users should check SapFluxNetDataMode in Project Configuration tab, and it will disable some functions that are specifically designed for raw sap flow measurements. Otherwise, they may encounter some unexpected errors.

6. Gap-filling using second sensor/tree

In the case of that users want to gap-fill one sensor using another sensor (sensors from same tree, or neighbor tree), they can treat the second sensor/tree as the environmental factor while they are determining environmental factors during project configuration and data loading. They can then use linear regression or other algorithms to fit a model and make predictions.