Online Prediction of Agricultural Model APSIM: A Step toward Emulation

Charles Labuzzetta

Iowa State University

Online Prediction of Agricultural Model APSIM: A Step toward Emulation

Introduction

Emulation of complex computer models can be performed as a means to avoid immense computational requirements of repeatably running such models for sensitivity analyses and large scale prediction (Gladish, Darnell, Thorburn, & Haldankar, 2019; Stanfill, Mielenz, Clifford, & Thorburn, 2015). Emulation refers to a simplification of the computer model via methods such as Gaussian processes, random forests and generalized additive models, among others, to predict the computer model output via a trained function of the computer model inputs. Such functional methods are more efficient in repetition and require less processing power than the original computer model (Shahhosseini, Martinez-Feria, Hu, & Archontoulis, 2019). Predicted output, however, will differ from the computer model. Therefore it is important to use an emulation method which minimizes the error in prediction. If error can be minimized to a desirable rate, emulation can be used confidently to simulate output data from the computer model in an efficient manner.

One such computer model which researchers have attempted to emulate is the agricultural simulator "Agricultural Production Systems sIMulator," i.e. APSIM (Holzworth et al., 2014). APSIM contains a complex set of modules, each with its own history of research and development, which can be linked together to model complex agricultural systems. "It contains interconnected models to simulate systems comprising soil, crop, tree, pasture and livestock biophysical processes. It is used extensively by researchers to assess on-farm management practices, climate risk/change adaptation strategies, mixed pasture/livestock strategies, agro-forestry resource competition, nutrient leaching under various conditions, gene trait expression and many other applications" (Holzworth et al., 2014).

The National Resources Inventory is a long term project funded by the National Resources Conservation Service which monitors trends in land use, soil, water and other resources on rural lands in the United States. The inventory is a statistical survey which carefully characterizes these factors via representative and efficient survey design

(Nusser, Breidt, & Fuller, 1998). Land use and soil erosion statistics are of particular interest to the National Resource Inventory. The National Resources Inventory makes broad assumptions in its calculation of the the revised universal soil loss equation (RUSLE) at surveyed points for estimation of annual soil loss at the state and county level (Renard, Foster, Weesies, & Porter, 1991). Integrating NRI data, APSIM, and auxiliary data sources may provide for more accurate estimation of soil loss on large spatial scales.

Problem

While the integration of APSIM into current NRI estimation techniques may improve accuracy in soil loss modeling, it would be infeasible to run the APSIM computer model for all points of interest. Therefore, emulation via a more efficient statistical method is desirable. Emulation of a complex computer model is in itself a complex task. It requires careful design and time consuming generation of the data for complex parameter spaces used for model training. This project is a first step toward the complex task of APSIM emulation. Here, a simple APSIM simulation will be generated which will be used to perform online prediction of measures of maize yield, biophysical processes, and soil parameters.

Online prediction is the task of forecasting outcomes within the near future while updating the model at each time step as new observations become available. This prediction task can be performed via a variety of modeling techniques, including multiple linear regression, time-series modeling, and machine learning methods. By using meteorological data from previous years as input for a single APSIM simulation, time-series output of variables of interest will be generated. The meteorological input data, simulation settings, and output variables will all be subject to modeling assumptions that will be explained in the Data Description section of this paper.

The purpose of this project is to explore methods which are adequate for simulating APSIM behaviors on a daily timestep. APSIM is a process based model which may behave in a manner that is incompatible with some statistical model assumptions. In this work, several statistical methods will be evaluated for their usefulness of emulating APSIM. Simple multiple linear regression and residual auto-regressive models will be developed first and serve as a baseline methods to more complex techniques. These complex techniques include alterations to the multiple linear regression and auto-regressive models as well as generalized additive models, and random forests.

Project Objectives

This project serves both as an important introduction for myself into the world of crop and soil modeling via process-based models while also being connected to my work with the National Resources Inventory. Learning objectives that I expect to accomplish with this project in relation to Agronomy 525 are:

- Learn how to run an APSIM simulation using the Maize and Erosion modules
- Learn about APSIM input parameters via generating simulation data
- Create a reproducible project workflow and documentation in GitHub
- Implement several online prediction methods
- Describe the feasbility of online prediction of APSIM

This project has been adapted to better fit within the time constraints of a class project. Several potential research projects which have brainstormed in connection to this project include:

- Study machine learning model emulation accuracy for nitrate and soil loss estimation while including observed meteorological data for previous years
- Compare neural network methods for APSIM emulation to previously tested machine learning methods
- Methods for improving NRI estimation via computer model emulation
- Large scale spatial estimation of nitrate and soil loss via APSIM emulation

Data Description

The APSIM Next Generation model will be used to generate the data studied for this project. The APSIM simulation described in the following Model Description section will generate data for a 35 year simulation of continuous Maize crop on a field in central Iowa. Meteorological and soil data from the central Iowa region will be used to parameterize local growing conditions. The output from APSIM will include the following variables:

- Clock.Today
- Weather.Rain
- Weather.Radn
- Weather.MaxT
- Weather.MeanT
- Weather.MinT
- Weather.VPD
- Maize.Phenology.CurrentStageName
- Maize.AboveGround.Wt
- Maize.Grain.Wt
- Maize.Leaf.Transpiration
- MicroClimate.RadiationInterception
- MicroClimate.PetTotal
- Soil.SoilWater.Runoff
- Soil.SoilWater.Drainage
- sum(Soil.SoilWater.ESW)

- Soil.SoilWater.LeachNO3

The Clock.Today, Maize.Phenology.CurrentStageName and all Weather module variables above will be used as covariates or auxiliary information for the prediction of some or all of the Maize, MicroClimate, and Soil model outputs generated by the APSIM simulation. Online prediction will occur using the true values of the observed Weather variables for the day of prediction, but forecasted values of these covariates could also be used in practice.

Model Description

A 35 year continuous corn APSIM Next Generation simulation was used to generate the data for this project. This simulation was generated by combining the properties of the 'Maize-example-agron-525.apsimx' file provided as a course material and a Classic APSIM simulation 'CornGrowthDevelopment (with solutions).apsim' provided by Matt Nowatske generated at the APSIM Training Course by the APSIM Initiative. The important variable initialization and assumptions are described below:

- Clock: 1/1/1980 - 12/31/2014 (35 years total)
- Weather: Daily (1980-2014) meteorological data downloaded from daymet.ornl.gov for Central Iowa at 42 decimal degrees latitude and -92 decimal degrees longitude. This 'Iowa C.met' file was supplied as course material for AGRON 525.
- Field: 1 ha with slope 0
- Fertiliser: 150 kg/ha of UreaN applied annually at sowing
- Sowing: Maize crop was sown at a fixed date, May 5, of each year at a depth of 50 mm, with 760 mm between rows and 8 plants per square meter. The maize cultivar GH_5019WX was selected due to its similarity in maturation and harvest date given for Central Iowa Maize crops at http://agron-www.agron.iastate.edu/courses/Agron541/classes/541/lesson03a/3a.2.html

- Harvesting: The crop was automatically harvested when APSIM phenology stage reached 'ReadyForHarvesting'

- Soil: A Nicollet No 1 soil paramterization was used originally obtained from http://websoilsurvey.sc.egov.usda.gov/App/HomePage.htm at 42.049 decimal degrees latitude and -93.4345 decimal degrees longitude

- SurfaceOrganicMatter: An initial soybean residual pool of 1250 kg/ha with a 27 g/g C/N ratio was assumed present on the field

- Maize: No default parameters from the Maize model were changed. The GH_5019WX cultivar was applied to the field.

Implementation

The APSIM Next Generation simulation was run using the 'apsimx' R package. The data analysis was conducted completely in R and the implementation can be found in my 'apsimo' repository on Github (https://github.com/labuzzetta/apsimo) which may be installed as an R package in order to reproduce my results. The simulation data generated from APSIM was partitioned into 3 datasets. The first 25 years of the simulation were selected as the training data to be used for the various online prediction methods. The last 10 years of the simulation were partitioned as validation data. The first 7 of these years will be used as testing data to validate the accuracy of each the presented methods. The final 3 years of the data will form a separate validation set which will be analyzed only at the very last stage of this project in order to validate all of the final methods on unseen data. This will be important for determining the final accuracy of these methods.

As the main study for this project, online prediction of the Maize.AboveGround.Wt variable was investigated. Figure 1 presents the Above Ground Biomass trends against the Day of Year for the first 25 years of the simulation. Figure 2 presents these data for the last 10 years of the simulation which will be used for the testing and secret validation sets. The methods developed for this variable are applicable to other Maize module outputs and may also be easily adapted to work with outputs from other APSIM modules.

Among outputs from the Maize module, non-zero outputs occur only between the sowing and harvest dates. To accommodate this behavior, the models used for online prediction must predict non-zero output only during the growing season. As the first step toward full APSIM emulation, I decided to include the Maize.Phenology.CurrentStageName variable within the inputs to the online prediction. Thus, when developing the online prediction algorithm, steps were included which prohibit non-zero online predictions before the crop was sown and after the crop was harvested. Before including these steps, every online prediction model would continue to predict non-zero values far after the crop had been harvested. These errors inflated the root mean square error which made it difficult to compare between methods. Additionally, I realized that the online prediction models should generally be trained only on data from within the growing season as the estimation for the effect of the Weather covariates would be interfered by the non-existance of the crop outside of the growing season.

The ‘online_emulate_maize()’ function of the ‘apsimo’ package is the main contribution of this project. Documentation for the function is provided my ‘apsimo’ package. This function is based on a general algorithm for online prediction which I describe after this paragraph. This general algorithm can be slightly altered to match any of the 5 different methods offered by the prediction type option ‘pred_var’ in the ‘online_emulate_maize()’ function. The ideology behind of each of these methods will also be discussed. Furthermore, the model type can be changed to implement multiple linear regression, auto-regressive, generalized additive models, and random forests using the ‘method’ option of the ‘online_emulate_maize()’ function. In order to organize the description of this function into an understandable format, I will first describe the general algorithm and then describe the changes which occur under each of the ‘pred_type’ options and then formulate the models for each ‘method’ option.

General algorithm:

1. Beginning with the first observation in the test set data, fit a 'method' option-type model to the selected 'pred_var' output variable data in the corresponding format determined by the 'pred_type' option.
2. Iterate through the test set for observation i using the qualifying observations according to the 'pred_type' option to update the model and perform online prediction of the next unseen observation via the following method:
   (a) If the day of the year is greater than or equal to 125, indicate the crop has been sown
   (b) If the previous Maize.Phenology.CurrentStageName observation indicates the crop has ripened, then indicate the crop is harvested on the current day.
   (c) If the crop has not yet been sown or it has been harvested: predict 0
   (d) Otherwise, use the current model to predict the output variable data for the next observation. Return 0 for any negative prediction (this may be altered in some cases).

Changes to general algorithm for each 'pred_type' option:

- online_total_full: This method uses the raw 'pred_var' values from 1 to $i-1$ as the dependent values for the model. The method name includes the term "online" because it updates the model as new information becomes available, "total" because the data format is often the accumulated total at each observation as seen in the Maize.AboveGround.Wt variable, and "full" because all previous observations are included in the model fitting.
- online_change_full: This method uses the difference between the $i-k$ and $i-(k-1)$ as the $i-k$ dependent values for each $k \in (1, n)$ where n is the total number of observations available for training the model in both the training and previously observed testing set data. The initial missing value at $i-(k-1) = -1$

is considered to be 0. The method includes the term "change" because it is based on the change in the dependent variable after each previous observation. While there is still certainly some temporal dependence between observations using this technique this method was developed because the change between observations may be a better approximate the between-observation independence assumption required for some models.

- online_total_local: This method uses the raw 'pred_var' values as the dependent values for the model, but only the previous $i-1$ to $i-l$ observations are included, thus localizing the amount of training data available for training the model. The value of l is set at a default of 30, but can be adjusted via the 'local_dist' option in the 'online_emulate_maize()' function. This method includes the term "local" because it only uses the previous l observations at each update of the model. I found by happenstance that this improved model performance for some 'method' options.

- online_change_local: This method combines the online_change_full method with the localized option which limits the differences in observations to only the previous $i-1$ to $i-l$, where again $l=30$ as a default.

- no_update: The no_update option trains the model only once on the data included only in the training set. The model is not updated at each iteration of online prediction. This method is most appropriate for models which take substantial time to train, especially when using the full extent of the training data.

Model type differences for each 'method' option:

- lm: This model type is a linear model with the following form:

$$\text{pred\_var}_j = \beta_0 + \beta_1 Weather.Rain_j + \beta_2 Weather.Radn_j + \beta_3 Weather.MaxT_j + \beta_4 Weather.MeanT_j + \beta_5 Weather.MinT_j + \beta_6 Weather.VPD_j + \beta_7 DayOfYear_j + \epsilon_j$$

Where $j \in \{i-l, i-1\}$ with l being either the total number of previous observations or the number of observations included in "local" training. It is assumed $\epsilon_j \sim iid\ Normal(0, \sigma^2)$. Model selection to reduce the number of covariates was not performed in order to avoid time being wasted over only small performance improvements by removing correlated covariates when fitting a large number of models.

- ar: This model type is a residual auto-regressive model generated by the following steps:

  1. Fit a generalized additive model for the 'pred_var' training data as follows (note the functional form for the DayOfYear covariate):

     $\text{pred\_var}_j = \beta_0 + \beta_1 Weather.Rain_j + \beta_2 Weather.Radn_j + \beta_3 Weather.MaxT_j + \beta_4 Weather.MeanT_j + \beta_5 Weather.MinT_j + \beta_6 Weather.VPD_j + f(DayOfYear_j) + \epsilon_j$

     Where $j \in \{1, n\}$ with n being the number of training data observations. It is assumed $\epsilon_j \sim iid\ Normal(0, \sigma^2)$. Model selection to reduce the number of covariates was not performed in order to avoid time being wasted over only small performance improvements by removing correlated covariates when fitting a large number of models.

  2. Use the fitted gam to predict 'pred_var' in both the training and test set data using the observed covariate values

  3. Calculate the residuals between all observed and predicted 'pred_var' values

  4. Fit an auto-regressive model to the $\{1, \ldots, i-1\}$ residual values and use the model to predict observation i

- gam: This model type is a generalized additive model with the following form (note the functional form for the DayOfYear covariate):

  $\text{pred\_var}_j = \beta_0 + \beta_1 Weather.Rain_j + \beta_2 Weather.Radn_j + \beta_3 Weather.MaxT_j + \beta_4 Weather.MeanT_j + \beta_5 Weather.MinT_j + \beta_6 Weather.VPD_j + f(DayOfYear_j) + \epsilon_j$

  Where $j \in \{i-l, i-1\}$ with l being either the total number of previous observations or the number of observations included in "local" training. It is assumed $\epsilon_j \sim iid\ Normal(0, \sigma^2)$. Model selection to reduce the number of covariates was not performed in order to avoid time being wasted over only small performance improvements by removing correlated covariates when fitting a large number of models.

- rf: This model type is a random forest. The selected and qualifying 'pred_var' observations are fit with the corresponding Weather.Rain, Weather.Radn, Weather.MaxT, Weather.MeanT, Weather.MinT, Weather.VPD, DayOfYear values as features. The default number of trees and other settings of the randomForest package were used, again to limit the amount of time spent fitting each model and provide consistency for out-of-the-box model assessment.

Discussion

To begin the discussion of this project, I think it is important to review the learning objectives which I identified early in the semester. The objectives were:

- Learn how to run an APSIM simulation using the Maize and Erosion modules
- Learn about APSIM input parameters via generating simulation data
- Create a reproducible project workflow and documentation in GitHub
- Implement several online prediction methods
- Describe the feasibility of online prediction of APSIM

Overall, I think that I was successful at accomplishing these objectives within my special project. The simulation which I decided to run in this project was certainly a simple one. Even though it was simple, I now feel confident importing the necessary data files into APSIM Next Generation and specifying reasonable parameter values for the main components which affect results when running a Maize module simulation. From this simple simulation, I returned the APSIM output variables I was interested in studying for online prediction from the Maize and SoilWater modules and was able to generate the simulated data by running APSIM through the ‘apsimx’ R package. Each of these skills were also reinforced in the lab activities in class.

I have tried hard to ensure that the workflow for this project has been well documented and is easily reproducible through my ‘apsimo’ GitHub repository. Several years ago, I learned how to generate R packages by studying with Dr. Jarad Niemi. I have tried to reproduce as much as I could remember in order to make the simulated APSIM data easily loaded via the package and create an easy-to-use function which can be used to run and compare different online prediction methodologies. There are certainly aspects of the repository and package that can be improved, so I look forward to any feedback that can be provided.

The rest of this discussion will focus on the results from the online prediction methods which I have implemented in this project. I decided to focus mainly on two output variables from the APSIM simulation data: Maize.AboveGround.Wt and Maize.Leaf.Transpiration. Originally, I intended for some of my work on this project to focus on soil variables, but we did not cover the related modules at any depth in the class. Furthermore, as I began to work on this online prediction project, I realized that I would have to develop separate methods for the soil variables as non-zero outputs are not restricted only to within the growing season. I intend to make some exploration into altering my Maize online prediction methods for the soil outputs, but I plan to explore models other than APSIM which are more reliable for soil modeling in regard to emulation options for NRI project erosion estimates.

The online prediction outcomes for several different model types on the Maize.AboveGround.Wt variable are presented in Table 1. The metric used for comparing the online prediction values to the observed outputs is the root mean squared error (RMSE). I include 6 different method types in this analysis, although many others were tested. The baseline test for this variable was the "LM: total" model which used a multiple linear regression model updated at each time-step and trained on all previous observations to predict the next observation. I also trained another linear model using the "online_change_full" option which used the change in

Maize.AboveGround.Wt between each observation as an attempt to reduce the effect of the temporal accumulation of the total Maize.AboveGround.Wt. While the "LM: change" model performed better with the 7 years of "test" set data, on the "secret" dataset of the 3 final years which these RMSE values are reported for in Table 1, the "LM: change" model performed more poorly than the "LM: total" model.

It was not expected that the linear models would perform very well compared to some more complex techniques. I was encouraged by the much better performance exhibited by the "AR: total" model and especially the "GAM: local" model. The "AR: total" model is formulated to accommodate temporal data, and the local generalized additive model better adjusts to the violation in the independence assumption when fitting the functional relation to the temporal information with the locally based method. The Random Forest model performed better than the linear models but I was not overwhelmingly impressed by its result. As I previously mentioned, these methods were evaluated on the "secret" validation data partition which was not analyzed until this final quantification of RMSE values. I did this to ensure that I was not developing methods by inadvertently searching for techniques which were over-fitting to the "testing" validation data partition. To visualize how each of the evaluated online prediction methods fit to the three "secret" years, Figures 3, 4 and 5 show each method compared to the observed values for years 2012, 2013, and 2014 respectively.

I also tested the same methods on the Maize.Leaf.Transpiration APSIM output (Table 2), but did not include the "LM: change" model as it was not applicable to this data type. For this output, the linear model "LM: total" similarly performed the worse, but this was expected as it was the baseline. The "GAM: full" model also performed somewhat poorly, but the "GAM: local" and "RF: no_update" models both performed significantly better. In this case, the "AR: total" auto-regressive model performed the best. I was pleased to see that this auto-regressive method performed well because I have had exposure to using such methods in my time-series coursework, but have not yet had the chance to see it work in practice. Of note, I once again did not see a large advantage to using a machine-learning based technique compared to other methods. It seems that research in the field of model emulation should still consider more traditional based approaches before defaulting to a machine-learning based approach. The RMSE values for evaluated methods on the Maize.Leaf.Transpiration data are found in Table 2 and the predictions are visually summarized in Figures 6, 7 and 8.

The final objective of my special project was to describe the feasibility of online prediction for APSIM outputs. Throughout this project I have also gained some intuition on the feasibility of more fully emulating APSIM. Obviously, online prediction compared to full emulation is a relatively simple task. I think that this project has demonstrated that online prediction of APSIM outputs is certainly feasible, especially when only a single output variable is of interest. When considering multiple outputs at the same time and without taking the time to write specific procedures relevant to each output, the generality of online prediction starts to break down. I do have some concern that full emulation of APSIM when considering many different output variables is a rather difficult task. I am interested in further exploring this topic however and plan to investigate the applicability of conformal inference to this work. One of my goals for this semester was to explore the feasibility of emulating process based models for the purpose of finding new ways to estimate erosion over large spatial scales in relation to my work with the NRI. While I now believe that APSIM would not be a good fit for such estimates, I am happy that I have explored this online prediction project because I have learned enough about process based model emulation that I may be able to select a more appropriate model which will be of use to NRI research.

References

Gladish, D. W., Darnell, R., Thorburn, P. J., & Haldankar, B. (2019, March). Emulated multivariate global sensitivity analysis for complex computer models applied to agricultural simulators. *Journal of Agricultural, Biological and Environmental Statistics*, *24*(1), 130–153. doi:10.1007/s13253-018-00346-y

Holzworth, D. P., Huth, N. I., deVoil, P. G., Zurcher, E. J., Herrmann, N. I., McLean, G., … Keating, B. A. (2014). Apsim – evolution towards a new generation of agricultural systems simulation. *Environmental Modelling Software*, *62*, 327–350. doi:10.1016/j.envsoft.2014.07.009

Nusser, S. M., Breidt, F. J., & Fuller, W. A. (1998). Design and estimation for investigating the dynamics of natural resources. *Ecological Applications*, *8*(2), 234–245. doi:10.1890/1051-0761(1998)008[0234:DAEFIT]2.0.CO;2

Renard, K. G., Foster, G. R., Weesies, G. A., & Porter, J. P. (1991). Rusle: revised universal soil loss equation. *Journal of Soil and Water Conservation*, *46*(1), 30–33. Retrieved from http://www.jswconline.org/content/46/1/30.short

Shahhosseini, M., Martinez-Feria, R., Hu, G., & Archontoulis, S. (2019, August). Maize yield and nitrate loss prediction with machine learning algorithms. *Quantitative Biology*. Retrieved from https://arxiv.org/abs/1908.06746

Stanfill, B., Mielenz, H., Clifford, D., & Thorburn, P. (2015). Simple approach to emulating complex computer models for global sensitivity analysis. *Environmental Modelling Software*, *74*, 140–155. doi:10.1016/j.envsoft.2015.09.011

R Implementation

The R package implementation of this project can be found at https://github.com/labuzzetta/apsimo. Installation details are provided in the repository README file.

name	**method**	**pred_type**	**rmse**
AR: total	ar	online_total_full	13.04
GAM: full	gam	online_total_full	123.27
GAM: local	gam	online_total_local	1.99
LM: change	lm	online_change_full	153.05
LM: total	lm	online_total_full	142.85
RF: no_update	rf	no_update	40.86

Table 1

RMSE values for online prediction methods using the online_emulate_maize() function evaluated on the "secret" validation data partition for the Maize.AboveGround.Wt APSIM output variable.

name	**method**	**pred_type**	**rmse**
AR: total	ar	online_total_full	0.267
GAM: full	gam	online_total_full	0.871
GAM: local	gam	online_total_local	0.324
LM: total	lm	online_total_full	1.353
RF: no_update	rf	no_update	0.347

Table 2

RMSE values for online prediction methods using the online_emulate_maize() function evaluated on the "secret" validation data partition for the Maize.Leaf.Transpiration APSIM output variable.

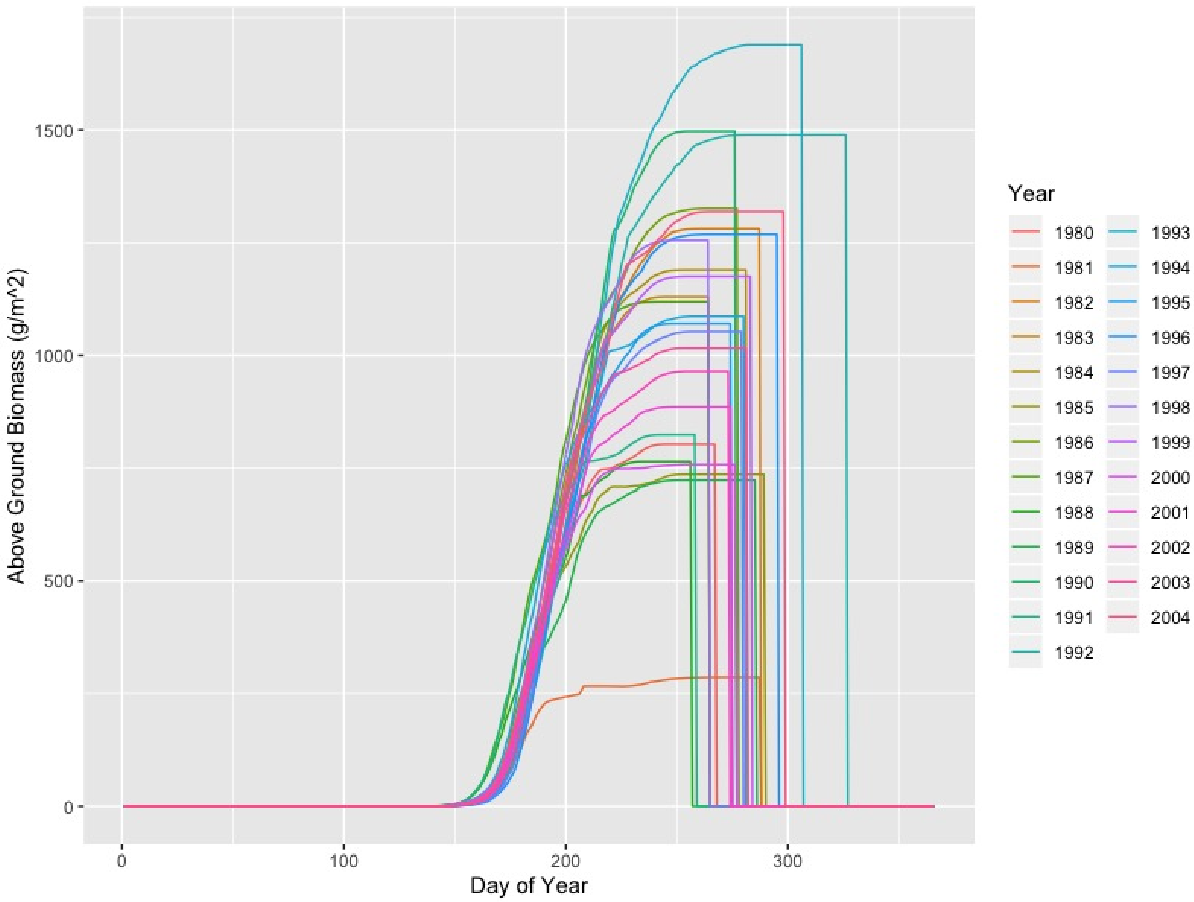

Figure 1. A subset of 25 years of simulated Above Ground Biomass output from APSIM used as a response for training data along with meteorological and day of year covariates.

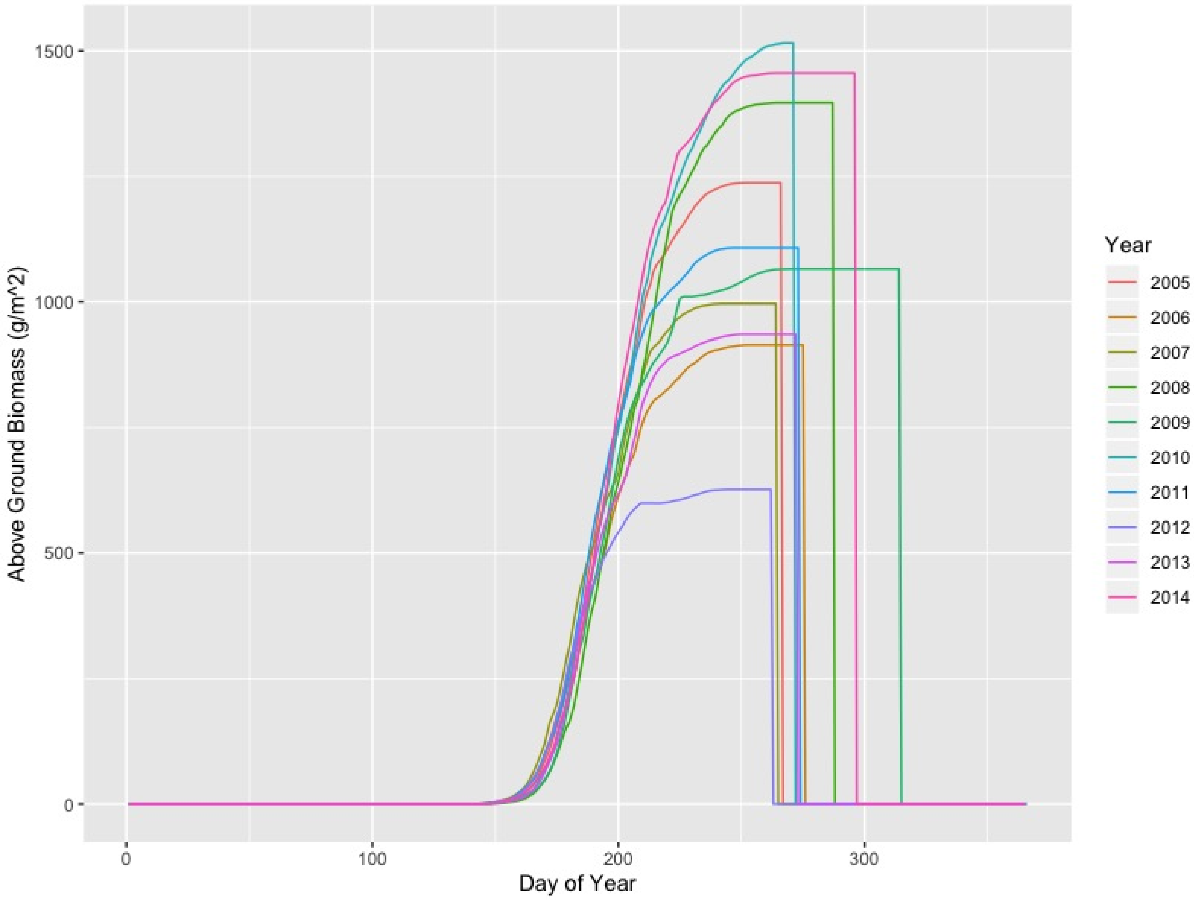

Figure 2. A subset of 10 years of simulated Above Ground Biomass output from APSIM used as validation data of selected online prediction methods.

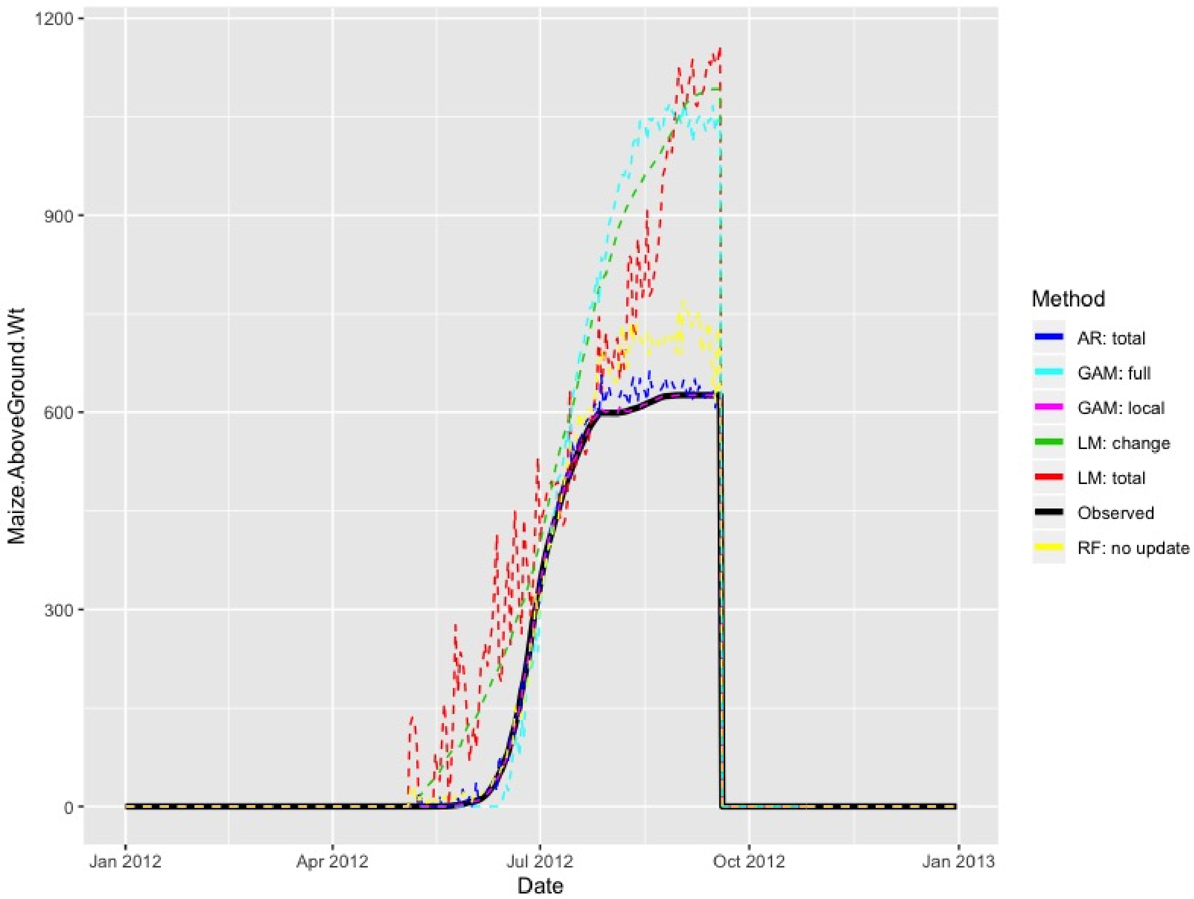

Figure 3. Visualization of online prediction methods evaluated on the first year of data from the "secret" validation partition for Maize.AboveGround.Wt APSIM output in 2012.

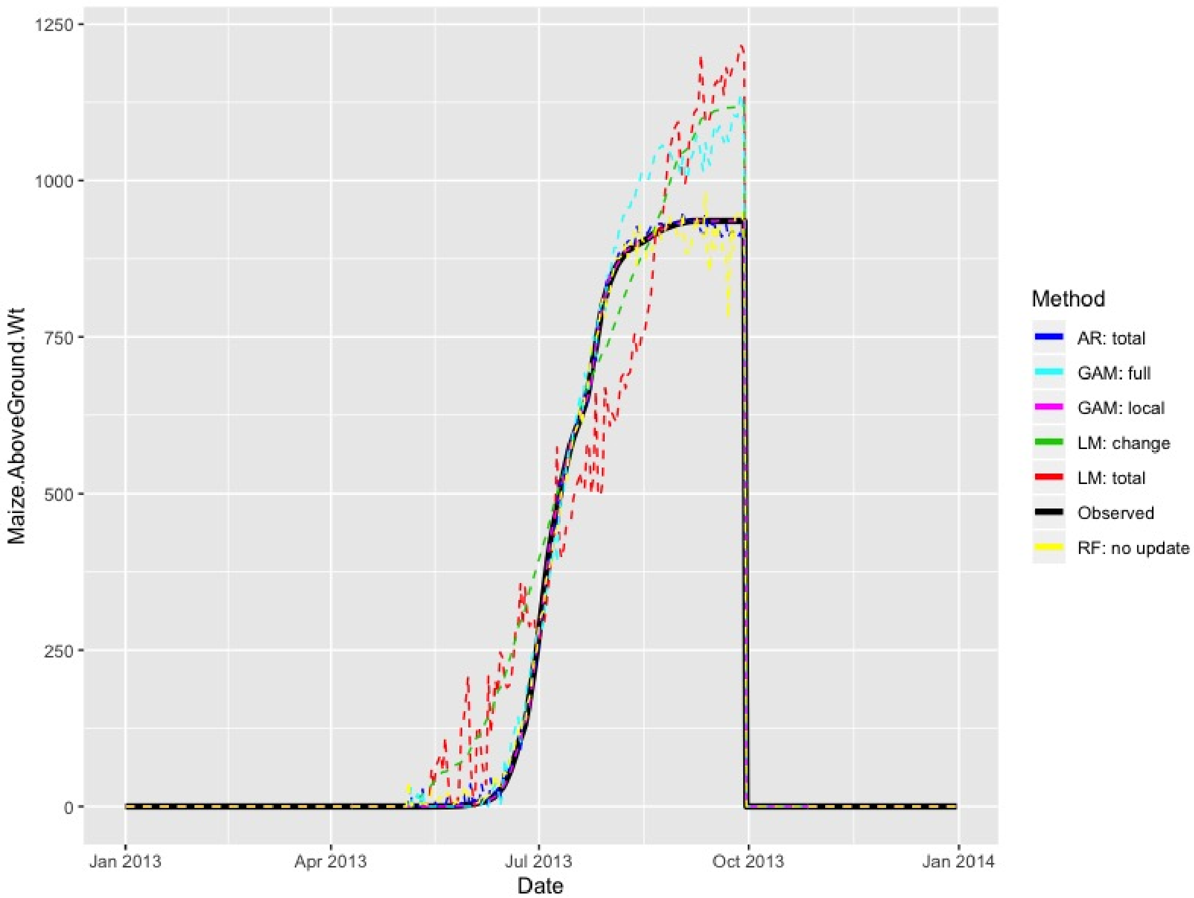

Figure 4. Visualization of online prediction methods evaluated on the second year of data from the "secret" validation partition for Maize.AboveGround.Wt APSIM output in 2013.

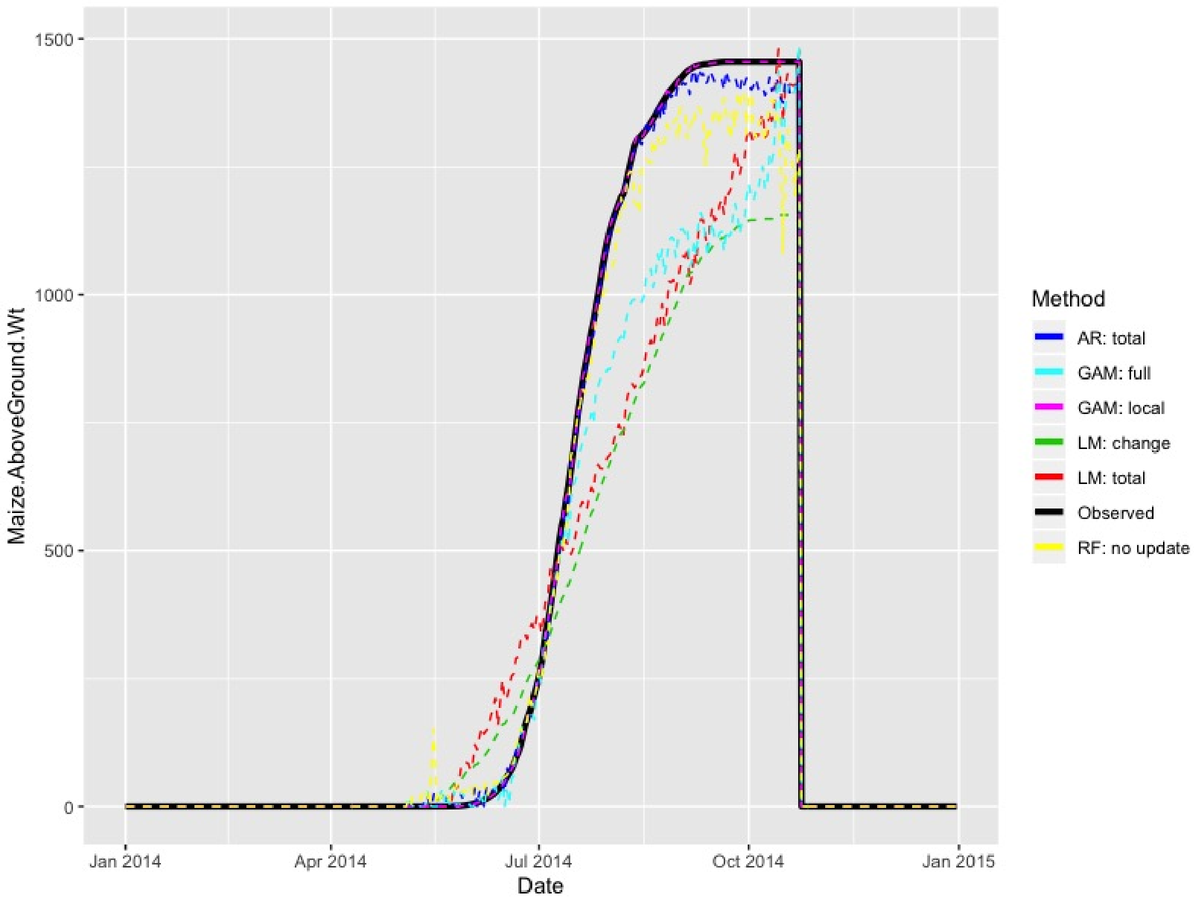

Figure 5. Visualization of online prediction methods evaluated on the third year of data from the "secret" validation partition for Maize.AboveGround.Wt APSIM output in 2014.

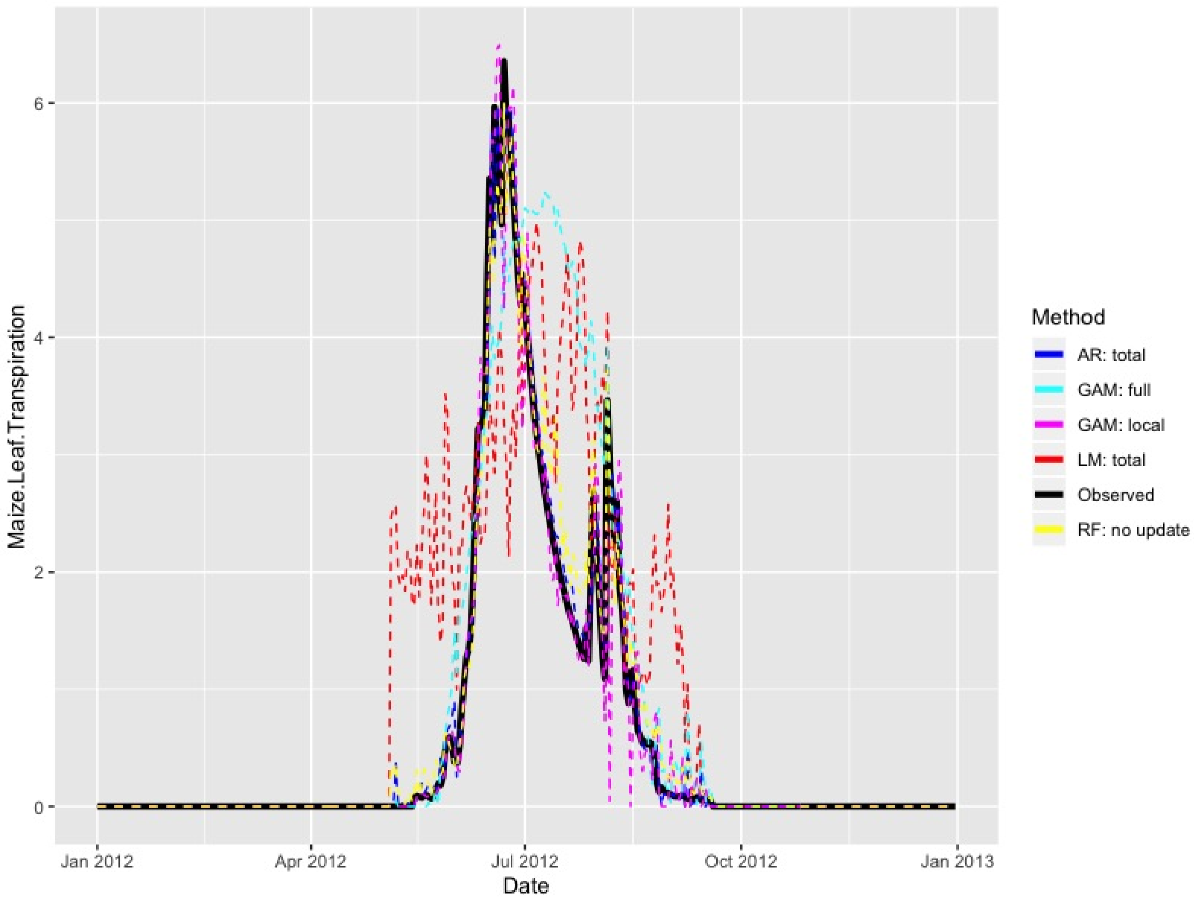

Figure 6. Visualization of online prediction methods evaluated on the first year of data from the "secret" validation partition for Maize.Leaf.Transpiration APSIM output in 2012.

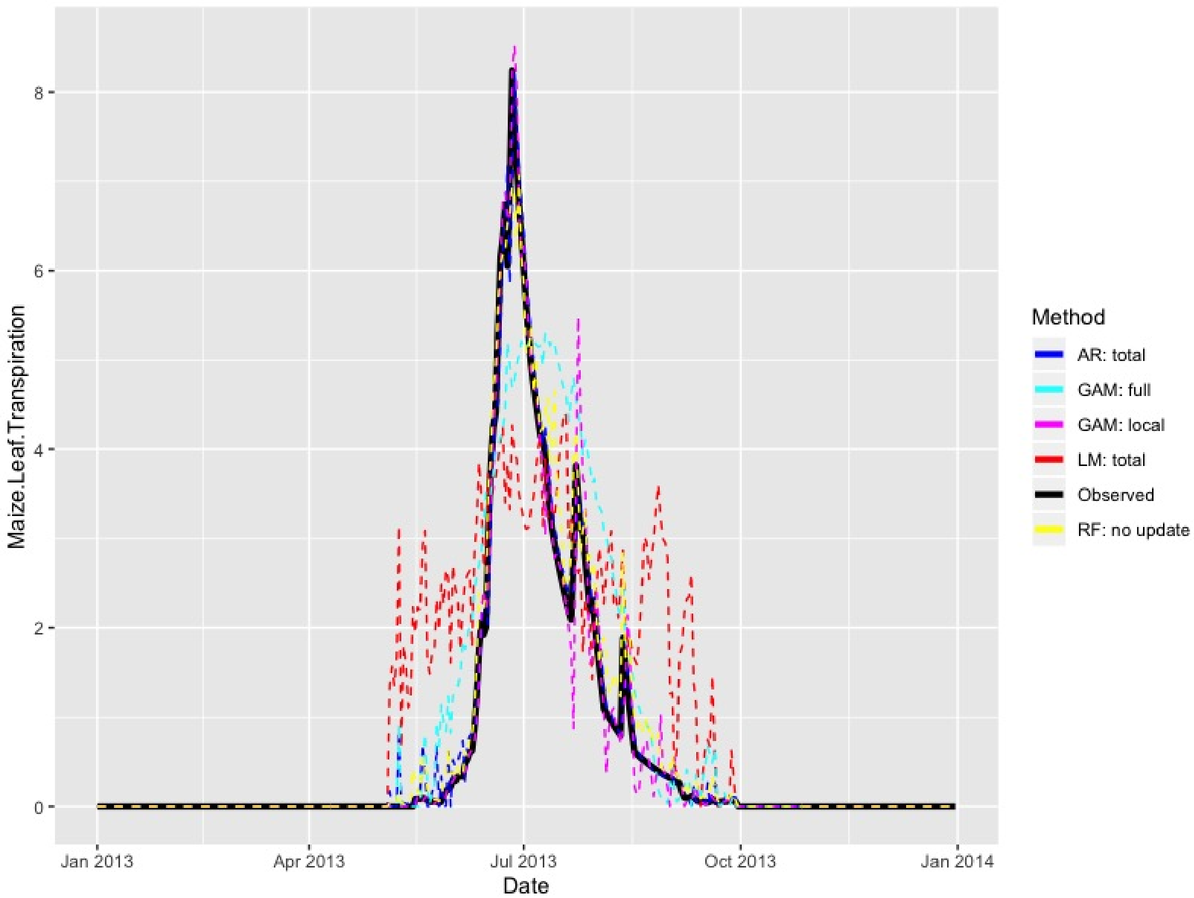

Figure 7. Visualization of online prediction methods evaluated on the second year of data from the "secret" validation partition for Maize.Leaf.Transpiration APSIM output in 2013.

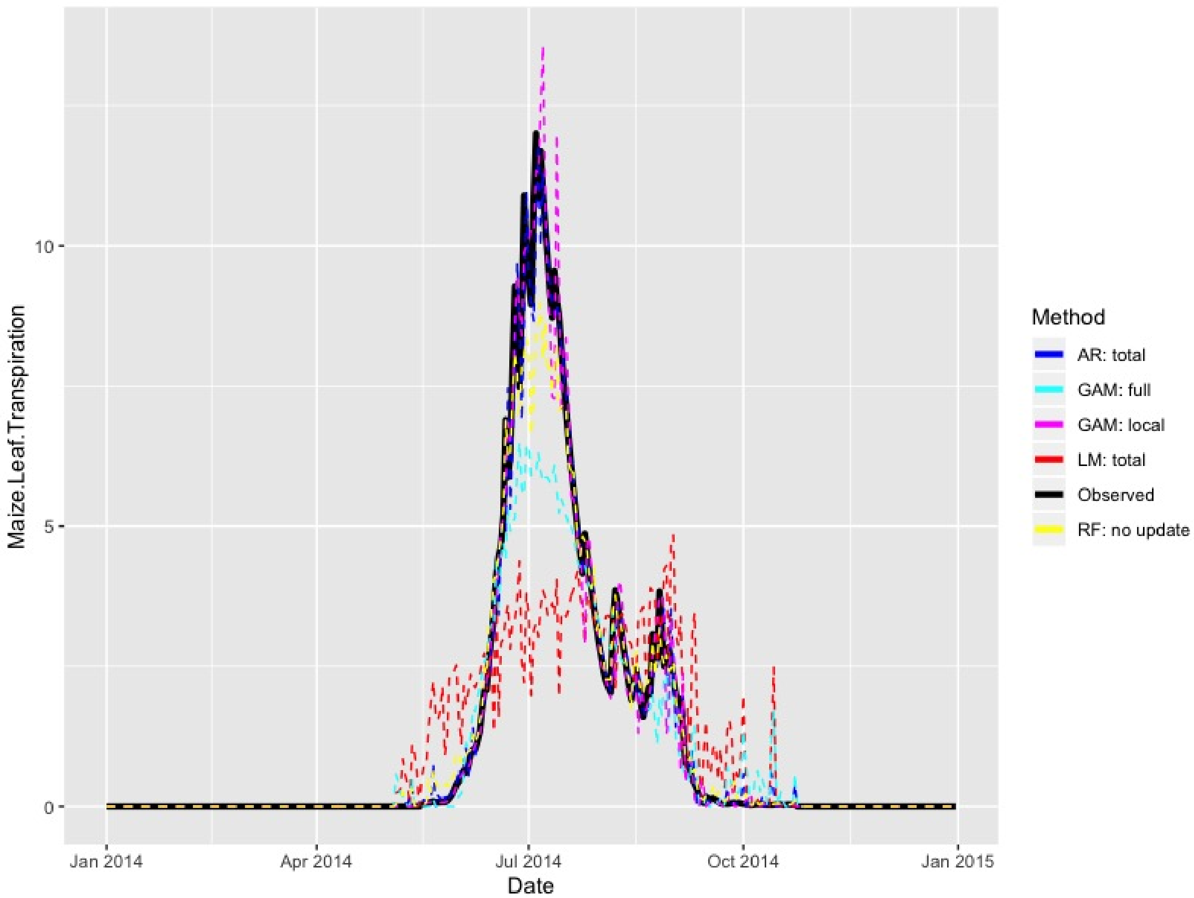

Figure 8. Visualization of online prediction methods evaluated on the third year of data from the "secret" validation partition for Maize.Leaf.Transpiration APSIM output in 2014.