1. Download the Codes.zip folder and unzip it. This folder contains three sub-folders named TC_Codes, FIA_Codes as well as Inputs_and_Outputs, a .Rproj file named Codes.Rproj and a .pdf file named README.pdf that contains the detail instructions to use these codes. The Inputs_and_Outputs folder contains a R data (.Rdata) and a CSV file (.csv) named parameters_used_to_simulate_basal_area_data and hypothetical_basal_area_data, respectively, and is intended to store the data as well as the outputs from these codes.

   (a) The R data file named parameters_used_to_simulate_basal_area_data is used to provide parameter values to simulate a hypothetical basal area dataset using the proposed approach and contains a list named beta_sigma2 that has three components:

      i. The first component is a vector containing regression coefficients to simulate $y^{(1)}$ data as described in Section 3.1 of the article.

      ii. The second and third components are a vector containing regression coefficients and a scalar containing variance parameter, respectively, to simulate $y^{(2)}$ data from the constrained model as described in Section 3.2 of the article.

      Each of these parameters values has been obtained as a noisy version (in a $\pm 20\%$ vicinity) of the empirical posterior median of the corresponding parameter obtained using the actual basal area data.

   (b) The CSV file named hypothetical_basal_area_data contains the simulated live tree basal area data. (Read the ACC form thoroughly to understand how this hypothetical dataset is simulated and, for which purposes, this dataset should and should not be used in lieu of the original data.) The data is in form of a matrix with five columns where first two entries in each row represents the coordinates of the field plot of live tree basal area data, next two entries represents the time point with year and month number (a number from 1 to 12) of the corresponding year, respectively, and the last entry shows the corresponding the live tree basal area measurement. In case the users want to use their own basal area data, it needs to be a CSV file (.csv) with the same structure as mentioned above and the file will be basal_area_data. Moreover, the users need to replace hypothetical_basal_area_data by basal_area_data in the TC_best_output_merge_basal_area.R file in the FIA_Codes folder.

2. Go to https://doi.org/10.7910/DVN/CZLKCK to download and save the data on Tasseled Cap (TC) variables in Inputs_and_Outputs folder. In case the users want to use their own remote sensing data, it needs to be a R data file (.Rdata) with the following structure: the file should contain a list named pixel_wise_TC_data that has two components:

    (a) The first component should be a location matrix with its rows representing coordinates of the pixel centers from the study area.

    (b) The second component should be the spatiotemporal data in form of a 3-dimensional array where first and second dimensions correspond to pixels and time points (months), respectively. Along the third dimension, it contains values for TC1 (Brightness), TC2 (Greenness) and TC3 (Wetness). The occurrences of Zero (0) within this array indicate missing data at those space-time combinations.

3. Install Rstudio. Then, open Codes.Rproj file using File/Open Project in RStudio or double-click the Codes.Rproj file. In case one wants to use the standalone R, the Codes folder needs to be set as the working directory of R using setwd("dir") where dir should be replaced by the path of the Codes folder.

4. Before sourcing any of these R codes, install the R packages: abind, ape, coda, fields, ggplot2, gridBase, lattice, ltsa, matrixStats, mvtnorm, proj4, RColorBrewer, sp, e1071, gbm, gam.

5. Now, start with TC_Codes folder and run Aggregation.R to aggregate every adjacent 16×16 pixels as described in Section 2 of the article. (Runtime: $\sim$ 21 minutes.)

6. Source Figure_2.R to reproduce Figure 2. (Runtime: Instantaneous.)

7. Then source TC_Model_NUMBER.R where NUMBER should be replaced by any one of the five candidate model numbers from Appendix A.1.1 in the supplementary materials $-$ I, II, IIIA, IIIB and IV. This code runs the particular model on the entire set of available TC observations and fills in the missing values. In addition, it calculates the predictive uncertainty, log likelihood (LL), BPIC and Moran's I statistic. (Runtime: For 60,000 iterations, Model I takes $\sim$ 24 minutes, Model II takes $\sim$ 1.1 hours, Model IIIA takes $\sim$ 3 hours, Model IIIB takes $\sim$ 7 hours, and Model IV takes $\sim$ 8 hours. Additionally, for Moran's I computation, $\sim$ 10 minutes is required for each of these codes.)

8. Run TC_cv_positions_selection.R to randomly select the positions that will be considered as test data in the cross validation. (Runtime: Instantaneous.)

9. Then source TC_Model_NUMBER_cv.R where NUMBER should be replaced by any

one of the five candidate model numbers from Appendix A.1.1 in the supplementary materials − I, II, IIIA, IIIB and IV. This code performs cross validation by holding out a subset of available TC observations as test set. (Runtime: For 36 test sets, per 60,000 iterations, Model I takes ∼ 24 minutes, Model II takes ∼ 1.1 hours, Model IIIA takes ∼ 3 hours, Model IIIB takes ∼ 7 hours, and Model IV takes ∼ 8 hours.)

10. Now, to reproduce Table A.1.1 and Figure A.1.1 of Appendix A.1.3 in the supplementary materials, run Table_A.1.1_Figure_A.1.1.R. (Runtime: Instantaneous.)

11. Source Figure_3_4.R to reproduce Figure 3 and Figure 4. (Runtime: Instantaneous.)

12. Run TC_best_model_output.R to merge the complete sets of TC1, TC2 and TC3 values, after filling in the missing ones using predictions from the best candidate model. (Runtime: Instantaneous.)

Note: The following steps, by default, will run the proposed model on the hypothetical basal area dataset that we have included and discussed in item 1(b) above. In case one wants to simulate a hypothetical basal area data different from what is already included in Inputs_and_Outputs folder, the code file Basal_area_data_simulation.R needs to be run, with a different seed and/or specifying a different set of model parameters. Then, following steps may be performed.

13. Now, go to the FIA_Codes folder and run TC_best_output_merge_basal_area.R to merge the output of TC_best_model_output.R with basal area data. (Runtime: Instantaneous.)

14. Then run Model_for_y1.R, Unconstrained_y2_model.R and Constrained_y2_model.R. (Runtime: For 60,000 iterations, Model for $y^{(1)}$ takes ∼ 1 minute, Unconstrained

Model for $y^{(2)}$ takes $\sim$ 1 minute, and Constrained Model for $y^{(2)}$ takes $\sim$ 5 hours.)

15. Now source Basal_area_cv_positions_selection.R. The outputs of this .R file are two .Rdata files where one contains the random positions of both zero, non-zero live tree basal area data and the other contains only the random positions of non-zero live tree basal area data those will be considered as test data in holdout cross validation method. (Runtime: Instantaneous.)

16. Run Model_for_y1_cv.R, Unconstrained_y2_model_cv.R, Constrained_y2_model_cv.R, Nonparametric_Models_for_y1_cv.R and Nonparametric_Models_for_y2_cv.R. These codes perform cross validation using holdout method. (Runtime: For 36 test sets, per 60,000 iterations, Model for $y^{(1)}$ takes $\sim$ 1 minute, Unconstrained Model for $y^{(2)}$ takes $\sim$ 1 minute, and Constrained Model for $y^{(2)}$ takes $\sim$ 5 hours. The cross validation of Nonparametric Models for $y^{(1)}$ and $y^{(2)}$ takes $<$ 1 minute each.)

17. Source Table_1_2_Figure_5.R to produce equivalents of Table 1, Table 2 and Figure 5 of the article. (Runtime: Instantaneous.)

18. Run Basal_area_model_cv.R to produce the cross validation outputs of combined model for B_{sk} (unconstrained $y^{(1)}$ and constrained $y^{(2)}$ model) equivalent to those described in Section 4.2 of the article. (Runtime: For 36 test sets, per 60,000 iterations, the basal area model takes $\sim$ 5 hours.)

19. Finally produce equivalents of Figure 6, Figure 7 and Figure 8 in the article by sourcing Figure_6_7_8.R. (Runtime: $\sim$ 3 minutes.)

Note: Runtimes, reported for the steps above, are as observed while implementing the codes using a single 3.79GHz processor in a Windows machine with 128GB RAM.