Supplementary Information for

Estimating divergent forest carbon stocks and sinks via a knife set approach

In the format provided by the authors and unedited

Supplementary Method 1: The practical guideline of the knife set (KS) approach in R programming language via cluster or cloud computing

1. Introduction

Target variables in the forest and natural environment are easily diverging and irregular. Furthermore, many studies have their own research aims, so the existing studies in the world often have different independent variables and the data sheets have the characteristics that contain quite a lot of missing values. For conquering problems mentioned above, we design a knife set (KS) approach that contains multitaskable functions and can be operated easily. The flowchart of the KS is described in the Figure 4 of the main text of "Estimating divergent forest carbon stocks and sinks via a knife set approach". Below we elaborate a very simple and low-cost method to practice the process of the algorithm that combines simple linear regression, machine learning (Gibbs sampling combined random forests) and climate classification tables to ensure that policymakers of any age in any country can fully understand and operate this algorithm by themselves easily.

2. Pre-treatment and Environment setting

2.1. Raw data collection

Raw data collected from google scholar, Nature publish group, Science (American Association for the Advancement of Science, AAAS), ScienceDirect, and so on for English peer-reviewed journal articles. However, not all critical and reliable data are published in English. Therefore, according to different species that need to be estimated, look for peer-reviewed articles in different languages and countries, and it is better to have local native speakers in the research team to correctly understand and collect data. For instance, *Phyllostachys edulis* (Carrière) J.Houz. (Moso bamboo) forests are mainly distributed in China, Japan, South Korea and Taiwan, so our research team has native speakers of simplified Chinese, Japanese, Korean, and traditional Chinese. Further, we collected local peer-reviewed research articles from CNKI[1], J-stage[2], KISS[3], and airiti Library[4].

After collecting peer-reviewed articles, carefully read articles, input data in an excel file, save it as .csv for further analysis, and double check for prevent mistyping. The raw data used in this study is as Supplementary Dataset 1, and the framework of template shown as Supplementary Figure 4. At the meanwhile of reading journal researches and making raw data sheet, record or define all variables in a table. (For instance, in the case of this research, common terms of variables defined as Supplementary Table 1.)

[1] Source: https://h5.cnki.net/#/home
[2] Source: https://www.jstage.jst.go.jp/browse/-char/ja
[3] Source: http://kiss.kstudy.com/search/sch-result.asp
[4] Source: https://www.airitilibrary.com/Search/alJnlbrowse

2.2. Pre-treatment of *k*-fold cross validation

k-fold cross validation is a common method used to evaluate machine learning models on a limited raw datasheet. The method is to randomly split the data into k sets, and then regard one set as validation dataset, and the rest of *k*-1 sets as training dataset, and repeat this process until each set is regarded as validation dataset once. And the prediction results are compared with the original raw data for performance comparison. Therefore, before the whole process starts, the value of *k* must be determined even though there is no hard rule for this value. Further, in previous studies, *k* was usually pre-set to 3 or 10 (divided into 3 or 10 validation dataset).

Under normal circumstances, studies rarely test the stability of a data matrix. However, the simple test of the stability of the data matrix is carried out to determine the value of *k* because there are only 81 observations[5] in this study.

2.2.1. Stability test in practice

For determining the *k*, we designed a dig testing which is randomly digging original data sheet in different percentage (such as 2%, 5%, 10%, 20%, and so on), then calculate the mean absolute different ratio of each dug value (MADREDV) as Eq (S1), and draw the correlation between dig rate and mean absolute different ratio of each dug value (MADREDV) (%).

$$\mathrm{MADREDV} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{(dnrvPi - Oi)}{Oi}\right| \qquad \mathrm{Eq(S1)}$$

where $dnrvPi$ is the dug and re-estimated value; Oi is the original dug values collected from peer-reviewed articles; and n is the number of dug observations.

The results of stability test in this study show that if the dig rate over 10%, the MDREDV will sharply increase over 50% (Supplementary Figure 5). Hence, we decided *k*=10 in this study and let the MDREDV will below 50%. (Of course, it is possible directly skip the above test, let *k*=10 or 3, depending on the number of observations. If observation size is large, *k*=3 will be better because of saving computing power.)

2.2.2. Separate data sheet into *k*-folds

Through the above test, it is possible to decide a suitable *k* (For instance, in this study, we decided *k*=10.). Then randomly re-arrange orders of observations in raw datasheet, and evenly divided observations of it into 10 .csv files (10% observations per file) as validation dataset. Grab the rest of each validation dataset as training dataset (90% observations per file) prepared for further analysis.

2.3. Environment setting

Because most of the computer operating systems used are Micro Soft (MS) Windows, the environment settings in this process will all take MS Windows as an example and operate via

[5] 81 observations for training and k-fold cross validation, and 24 observations for holdout testing.

R statistics and R studio.

2.3.1. Install R in cluster computing (Windows)

Download R from https://cran.r-project.org/mirrors.html and R studio from https://rstudio.com/products/rstudio/download/ , then install them in every single computer in the cluster used to run R codes and operate the following algorithm.

2.3.2. Install R in cloud computing (Amazon Web Services, AWS) (optional)

If research fund is enough, it is also possible using cloud computing to speed up all calculations. Below we introduce how to operate Amazon Web Services (AWS) for cloud computing.

2.3.2.1. Visit AWS web site https://aws.amazon.com/ , click Amazon EC2, apply a new account, and then log in.

2.3.2.2. Press “Launch instance” (Then you will see whole of the instance types)

2.3.2.3. Go to “Community AMIs” (AMIs means Amazon Machine Images).

2.3.2.4. Press “Ubuntu” and search “rstudio”

2.3.2.5. Choose “RStudio-1.3.1073_R-4.0.2_CUDA-10.1_cuDNN-7.6.5_ubuntu-18.04-LTS-64bit - ami-00b6908a3a1faec83 “

2.3.2.6. Choose an instance type you want. [If you need very strong computing power and have enough funding, it is also possible to choose 128 vCPUs and 3904 (GiB) Memory.]. Then, press “Next: Configure Instance Details”.

2.3.2.7. Check the instance details, then press “Next: Add storage”.

2.3.2.8. Check the detail of storage. (If your data matrix is really large, it is better to choose suitable storage size. e.g. 100 GiB), press “Next: Add Tags”, and then press “Next: Configure Security Group”.

2.3.2.9. Choose SSH for Type. Then, press “Add Rule”, choose “Custom TCP” for Type, set 80 for Port Range. Further, press “Add Rule”, choose “Custom TCP” for Type, set 8787 for Port Range. (Make sure the source of 80 and 8787 both are “0.0.0.0/0, ::/0”). In this page’s end, press “Review and Launch”.

2.3.2.10. After check the details of instance, press “Launch”. Then, the user interface will ask you to choose an existing key pair or create a new key pair. (The step is quite important because the key can use to connect via SSH). If it is the first time to launch instance, create a new key pair. Then, press “Launch Instance”.

2.3.2.11. Wait for 2 to 8 minutes until the Instance State turn yellow to green light.

2.3.2.12. Copy the Public DNS (IPv4) and paste to URL bar, then press enter for entering the user interface of your new launched instance.

2.3.2.13. Input Username “rstudio” and Password which is preset your instance ID provided by AWS (such as this kind of format i-09d7c7c7e455d5612).

2.3.2.14. After enter the R studio user interface, the following steps are the same as usual computers for operating R. (Due to the user interface of AWS is frequently changing, the latest process will continually update on GitHub from time to time if the first author is still alive, at https://github.com/gn03138868/WTF_sub-algorithm)

2.4. Input data

After collecting raw data, saving file as .csv, and building environment for running codes, open R studio software and run the code as below:

```
                                inputrawdata.R10.k.fold.X.training
                                <- read.csv("D:/Folder/rawdata k-fold X training.csv",
# Define the pathway of your input .csv files #
                                header=T, row.names = 1,
# It means the first row is title #
                                sep=",")
# Separate each value via comma #
                                head(inputrawdata.R10.k.fold.X.training)
# Show the input raw data #
                                summary(inputrawdata.R10.k.fold.X.training)
# Basic information of input raw data #
                                str(inputrawdata.R10.k.fold.X.training)
# Show the structure of input raw data #
```

The name defined after data entry does not have a mandatory form. It is possible named logically and easy to recognise as well. Furthermore, sometimes, different countries' pathway in R perhaps has different styles. If you are not sure about it, please google it and refer other pathways' styles.

2.5. Install packages in R

Install packages in R before running the KS.

```
                                install.packages("mice")
                                require(mice)
# Borrow the framework from MICE for our wandering through #
# the random forests (WTF) sub-algorithm #
```

At the beginning, we wanted to write the entire WTF architecture by ourselves, but found that there would be thousands or even tens of thousands lines of codes, which was very inconvenient for policymakers to understand and apply. Fortunately, in the R community, there are packages that have been released and similar frameworks can be applied for our research

purpose. MICE[6] is an excellent one of them.

```
                                    install.packages("Hmisc")
                                    require(Hmisc)
# Package for Pearson's correlation test #

                                    install.packages("corrplot")
                                    require(corrplot)
# Package for correlation matrix #
```

If the total number of variables is below 50, it will be more convenient to use corrplot. If not, Hmisc must to be installed. We will elaborate more details about Pearson's correlation in section 3.

```
                                    install.packages("ggplot2")
                                    require(ggplot2)
                                    install.packages("GGally")
                                    require(GGally)
                                    install.packages("ggfortify")
                                    library(ggfortify)
# Install packages for visualisation #
```

It is possible to draw different kinds of figures after installing above packages in R.

2.6. Data observation

After finish input data and install packages, observe the data for checking missing values.

```
                                    md.pattern(inputrawdata.R10.k.fold.X.training)
# Observe missing values #
```

3. Feature selection for linear regression (LR) models – 1st layer

3.1. Single features selecting (correlation tests)

We wanted to use stepwise for feature selection for making multilinear regression model initially. However, our data has a special character - a lot of missing values exist in data sheet, which means not every research investigate usual features in their own research. And after several

[6] The R package mice imputes incomplete multivariate data by chained equations constructed by van Buuren and Groothuis-Oudshoorn (2011). The original purpose in designing MICE was to use chained equations to impute missing values in the data sheet. Following this, Shah et al. (2014) used the framework of MICE and combined random forests to impute missing values. In our purpose is to estimate the unknow and chaotic target variables related to forest carbon stocks and sinks.

times modification and rolling correction improved by conferences and seminars, we decided to construct simple linear regression models for estimating target variables which could be predicted by independent variables of the 1st layer of the KS and must be in high accuracy and precision and to leave the irregular and divergent target variables to the next layer - Wandering through the random forests (WTF) sub-algorithm for estimating the target variables contained a lot of unobserved variables. Hence, we use Pearson (Pearson, 1901), Spearman (Spearman, 1904), and Kendall (Kendall, 1938) correlation methods to select features of each independent variable. The R codes describe as below:

```
                    cor.test(~Variable A + Variable B,
                    Data's name,
                    method = "spearman", "pearson" or "kendall")
# Correlation tests #
```

where the Variable A is target variable, the Variable B is independent variable, Data's name is the name of data you input in R, method = "_____" is for determining the method who want to use for correlation test.

If the Pearson correlation coefficient $r > 0.85$ and $p<0.01$, construct linear regression (LR) models for the target (dependent) variables. If not, transfer the estimation work to the next layer of the KS.

3.2. Constructing linear regression models

The independent variables with high precision and accuracy for target variables can be screened out after analysing all the correlation between target and independent variables. Then construct a simple linear regression model for each target variable as the Eq (S2) and (S3) below.

$$Y = aZ + b \qquad \text{Eq(S2)}$$

$$Y = c \times e^{aZ} \qquad \text{Eq(S3)}$$

where Y is a target variable, Z is an independent variable, a is a coefficient of the independent variable, *b* is constant, *e* is Euler's number, and *c* is the coefficient of *e*.

The above equations can be easily made in Microsoft Excel even if in R as the same, which software policymakers use depending on which softwares they are familiar with.

```
LRmodelsName = lm(Variavle B~ Variable A,
                  data = Data's name,
                  na.rm = TRUE)
# Construct simple linear regression model as Eq(S2) #
```

Where the RLmodelsName is the Regression models you named, lm(Variavle B~ Variable A) is a function code in R statistics software.

```
LRmodelsName = lm(Variavle B~ exp(Variable A),
                  data = Data's name,
                  na.rm = TRUE)
# Construct simple (exponential) linear regression model as Eq(S3) #
```

Where the exp(Variable A) is an exponential function code in R statistics software.

```
summary(LRmodelsName)
# Statistics summary #
# If the R^2 < 0.85 or >-0.85, it is good for constructing linear regression models. #
```

Select independent variables which can be used to estimate target variable in high accuracy and precision via above simple correlation tests.

3.3. Diagnosis of linear regression models

After constructing leaner regression models, it is better to do some diagnosis than never. Before operating the diagnosis, draw the figure of distribution first.

```
qplot(x = Variable B,
      y = Variable A,
      data = Data's name, na.rm=TRUE)
# Observing distributions of target and independent variables. #
```

where the Variable A is target variable, the Variable B is independent variable, Data's name is the name of data you input in R, na.rm=TRUE is for negating missing values.

Theoretically, after the regression model is fitted, its residual value must conform to the three assumptions of normality, independence, and homogeneity of variance, so the next three tests are used. Check whether these three conditions are satisfied or not.

```
shapiro.test(LRmodelsName$residual)
# Normality test #
```

where the LRmodelsName is the name of linear regression models constructed above.

If the *p*-value is high (perhaps over 0.05 depended on the confidence level), it cannot reject the null hypothesis which means the assumption of normal distribution of residual value is reasonable. If the *p*-value is low, the normal distribution of the residual value is unreasonable.

```
durbinWatsonTest(LRmodelsName)
# Independence test #
```

If the *p*-value is high (perhaps over 0.05 depended on the confidence level), it cannot reject the null hypothesis which means the independence of the residual value is reasonable.

If the *p*-value is low, the independence of the residual value is unreasonable.

```
ncvTest(LRmodelsName)
# Homogeneity of variance test #
```

If the *p*-value is high (perhaps over 0.05 depended on the confidence level), it cannot reject the null hypothesis which means the homogeneity of variance of the residual value is reasonable. If the *p*-value is low, the homogeneity of variance of the residual value is unreasonable.

Through the above R codes and processes, simple linear regression models can be prepared for estimating target variables in the 1st layer of the KS approach if above conditions are satisfied. Input estimated values, which obtained from those simple LR models, into the washed data matrix as an input matrix for the 2ed layer's training.

4. Wandering through random forests (WTF) sub-algorithm – the 2ed layer

4.1. Background of the WTF sub-algorithm

A case of simple linear regression (LR) models has good performance in estimating aboveground carbon (AGC) stocks but awful in belowground (Supplementary Figure 6). The major reason is that target variables are always affected by a lot of environmental and anthropogenic factors (Table S2) caused the correlation between the dependent and independent variables are easily irregular even divergent and chaotic.

For conquering the problem, we used a machine learning method called wandering through the random forests (WTF) sub-algorithm, in the 2ed layer to solve the problems that the 1st layer cannot handle. The WTF sub-algorithm is composed of random forests and a Markov chain Monte Carlo (MCMC) type algorithm - Gibbs sampling. The pseudocode of the WTF sub-algorithm describes in Supplementary Table 2.

Through the WTF sub-algorithm, target variables with complex independent variables can be estimated higher accurately and precision than simple linear regression models. Below we describe the process of implementing R statistical operations.

4.2. The 2ed layer in practice

For finding the lowest loss function of combination in duplication, CARtrees, and iteration numbers, we input training dataset for running R codes via cluster computing or cloud computing. If we write the framework of the WTF algorithm completely by ourselves, it will cost quite a lot of time and need to debug. Fortunately, "mice" package provides the corresponding Gibbs sampling architecture, and can be bridged to the "randomForest" package. So here borrowed and combined these two packages as the basis for running the WTF sub-algorithms. We appreciate to these research teams, who wrote these two packages, and give them respect (van Buuren and Groothuis-Oudshoorn, 2011; Shah et al., 2014). Now it is possible run this WTF sub-algorithm with just a few lines of codes.

4.2.1. After making some simple LR models, estimating the NA values, and inputting in original datasheet as a new one for the 2ed layer, load the new datasheet into Rstudio.

```
                                    inputrawdata.R10.k.fold.X.training
                                    <- read.csv("D:/Folder/rawdata k-fold X training for L2.csv",
# Define the pathway of your input .csv files #
                                    header=T, row.names = 1,
# It means the first row is title #
                                    sep=",")
# Separate each value via comma #
                                    head(inputrawdata.R10.k.fold.X.training)
# Show the input raw data #
                                    summary(inputrawdata.R10.k.fold.X.training)
# Basic information of input raw data #
                                    str(inputrawdata.R10.k.fold.X.training)
# Show the structure of input raw data #
```

4.2.2. Then, start to training out the minimised loss function combination of duplication, CARtrees, and iteration numbers. The implemented R code is as follows.

```
## m=X; maxit=Y; trees=Z ##
## It is better to try different combinations of X, Y, and Z based on different data ##
characteristics ##
mice.data.R10.k.fold.k.training.1.mXmYtZ
                <- mice(inputrawdata.R10.10.fold.2.training.1,
            m = X,              # generate 200 data sheets #
                                # It is possible to generate more than 100 sheets# # if you
                                like #
            maxit = Y,           # max iteration times#
            method = "rf",      # CART means Classification And Regression Tree #
                                # rf means random forest #
        # (It is also possible choose other methods such as linear regressions #
        # logistic regression, cart, random forest, bootstrap if you'd already
        # installed related packages) #
            ntree=Z,
        # ntree means CARtrees number in random forest #
            seed = 188)
        # set.seed()Let the sample survey data be the same every time #
summary(mice.data.R10.k.fold.k.training.1.m200m150t100)
save.image("Z:/… …/result_file_name.RData")
```

4.2.3. After running the above codes, call results out for calculating.

```
                    miceX.R10.k.fold.k.training.1.mXmYtZ <-
                    complete(mice.data.R10.k.fold.2.training.1.m200m10t10, X)
# first data sheet called as mice1#
# repeat this code to call all duplicates X = 1, 2, 3, …, X #
                    head(miceX.R10.k.fold.k.training.1.mXmYtZ)

                    miceX.R10.k.fold.k.training.1.mXmYtZ
                    [is.na(miceX.R10.k.fold.k.training.1.mXmYtZ)]<-0
# replace NA to 0 for further calculations #
```

4.2.4. After call out results, calculate MSE of different X.

```
                    mice.mean1_10.R10.k.fold.k.training.1.mXmYtZ<-
                    (mice1.R10.k.fold.k.training.1.mXmYtZ
                    +mice2.R10.k.fold.k.training.1.mXmYtZ
                    +mice3.R10.k.fold.k.training.1.mXmYtZ
                    + …
                    +miceX.R10.k.fold.k.training.1.mXmYtZ
                      )/X
# Take the means of sum of the matrix, #
# which is the largest number of duplication X, #
# as the mean of maximum duplicates dataset values (Mi). #
                    head(mice.mean1_10.R10.k.fold.k.training.1.mXmYtZ)
```

4.2.5. After calculating the MSE of different combination of X, Y, and Z, then draw the relationships between $\lambda(M)$ and the different combination of them (Supplementary Figure 7).

4.2.6. Choose the combination with the lowest $\lambda(M)$ for further testing set after all combination of $\lambda(M)$ have converged. For instance, in the case of Supplementary Figure 7, the combination with the lowest $\lambda(M)$ is X = 150, Y=150, and Z=100.

After above process, we get a trained combination for one of *k* folds in training set.

5. *k*-fold cross validation[7]

5.1. Combine original training and validation dataset as one .csv file after get the best

[7] Do not to grab "testing data set" into the WTF sub-algorithm (the 2er layer) for training or to find parameters.

combination of duplications, iterations, and numbers of CARTrees.

5.2. Then, input variables via simple LR models made from step 3.2. [In this step, it is possible to use Excel, MATLAB, or R which the operator prefers to use.]

5.3. Run the same process from step 4.2.1. to 4.2.4. and get the mean of output matrix from the 2ed layer.

6. Climate classification table – the 3rd layer

If the mean of output matrix has missing values inside, which means the output target variable have not pass the OOB (out-of-bag) test mentioned in Table S5, need to follow IPCC Guidelines for National Greenhouse Gas Inventories and make Table S4 prepared from Table S1. Then, input the values into the output of the 2ed layer as the output of the 3rd layer which is also the final output of the KS. (At least, using the average of the global data can ensure that the accuracy of the estimated results is high.)

7. Final output compared with benchmark

7.1. Obtain the benchmark

Use the IPCC Guidelines for National Greenhouse Gas Inventories[8] and re-estimate the target variables to replace the original data matrix as the benchmark data matrix.

7.2. Performance calculating

Calculate the mean absolute percentage error (MAPE), root mean square error (RMSE), mean absolute error (MAE), mean error (ME), which mentioned in main text of Equation (2) to (4), of the benchmark and the estimated KS output provided by *k*-fold cross validation. Of cause, it will also get each observation's absolute percentage error (APE), absolute error (AE), and error. For further statistical process. (Meaning of indices for the performance of algorithms describe in Supplementary Figure 8.)

7.3. Statistical process of different algorithms

After obtain the benchmark and performance calculating, save the results data as a .csv file. And choose one of the most familiar statistic software of yourself and input the data into the software for further process. The flowchart of statistics process of comparing the performance between different algorithms as shown in **Supplementary Figure 9**. In this study, we use R statistic software to operate it.

7.3.1. Shapiro-Wilk normality test

[8] Original sources of the IPCC's guideline: https://www.ipcc.ch/report/2006-ipcc-guidelines-for-national-greenhouse-gas-inventories/
Recently, it was refined here in 2019: https://www.ipcc.ch/report/2019-refinement-to-the-2006-ipcc-guidelines-for-national-greenhouse-gas-inventories/

This step is for check that the group is normal distribution or not (Royston, 1982). If the *p*-value > 0.05, the data is normal distribution; if the *p*-value < 0.05, the data is not normal distribution.

7.3.2. Homogeneity of variances' test in multiple groups

It is better to check the homogeneity of variances before recognising those each group is significant different or not via Bartlett's or Levene's test. If the result of homogeneity of variances' test is homogeneity, use one-way ANOVA; if not, use Welch's Heteroscedastic F Test.

7.3.2.1. Bartlett's test

Bartlett's test is suit for normal distributions groups. If the *p*-value > 0.05, variances are homogeneity; if the *p*-value < 0.05, variances are heterogeneity.

7.3.2.2. Levene's test

Levene's test is suitable for other types' distribution. If the *p*-value > 0.05, variances are homogeneity; if the *p*-value < 0.05, variances are heterogeneity.

7.3.3. one-way ANOVA or Welch's Heteroscedastic F Test

If the *p*-value < 0.05, difference is statistically significant; if not, difference is statistically insignificant (alpha = 0.05).

7.3.4. Post-hoc test

If variances in one-way ANOVA are significant different, just operate HSD, LSD (for same samples' number) or Scheffe test (for different samples' number). Or if the variables are heterogeneity and the results of Welch's Heteroscedastic F Test is significant different, the post-hoc test is better to use Games-Howell method.

8. Holdout testing (optional)

Holdout testing is used to test whether the model has overfitting[9]. (That is, it performs well when training the model, but it performs poorly in empirical applications.) The testing is good to use if operators have a large dataset. The dataset will split 20% to 30% of observations for holdout testing generally and isolated from training and validation dataset[10].

---

[9] Holdout testing for validating the performance is very important if we want to observe the generalisation ability of machine learning models (such as the convolutional neural network [CNN]). The network of CNN becomes a very flexible function mapper by adding more neurones in the hidden layer. This rises the risk of overfitting. However, it is elegantly to prevent this kind overfitting by out of bag (OBB) test in the step of random forests (Breiman, 2001) of the WTF sub-algorithm of KS approach. Although some studies suggest that holdout testing is no need to used when the number of observations is not enough, we still strongly recommend that if the number of samples is sufficient, it is nice to operate it. After all, each dataset has different characteristics.

[10] Some studies will call training dataset, validating dataset, and testing dataset as the same as training data set, validation dataset and holdout dataset in this study. It is because we want prevent the misunderstanding in the study which the training dataset splits into training set and validating set in OOB test in the WTF sub-algorithm based on the Gibbs sampling and random forests.

8.1. Attach observations of holdout dataset to the training dataset.

8.2. Remove nearly all target variables and independent variables, and remain those traits which policymakers have.

8.3. Operate the KS approach

8.4. Get the output of target variables

8.5. Calculate the MAPE, RMSE, MAE, and ME as the same as step 7.2.

8.6. If the performance of holdout testing is better than *k*-fold cross validation or there is no significant difference, the model is not overfitting. Congratulations!

9. Estimating target variables by the KS approach

9.1. Attach observations[11] of unknow places which policymakers perhaps just have limited traits of them.

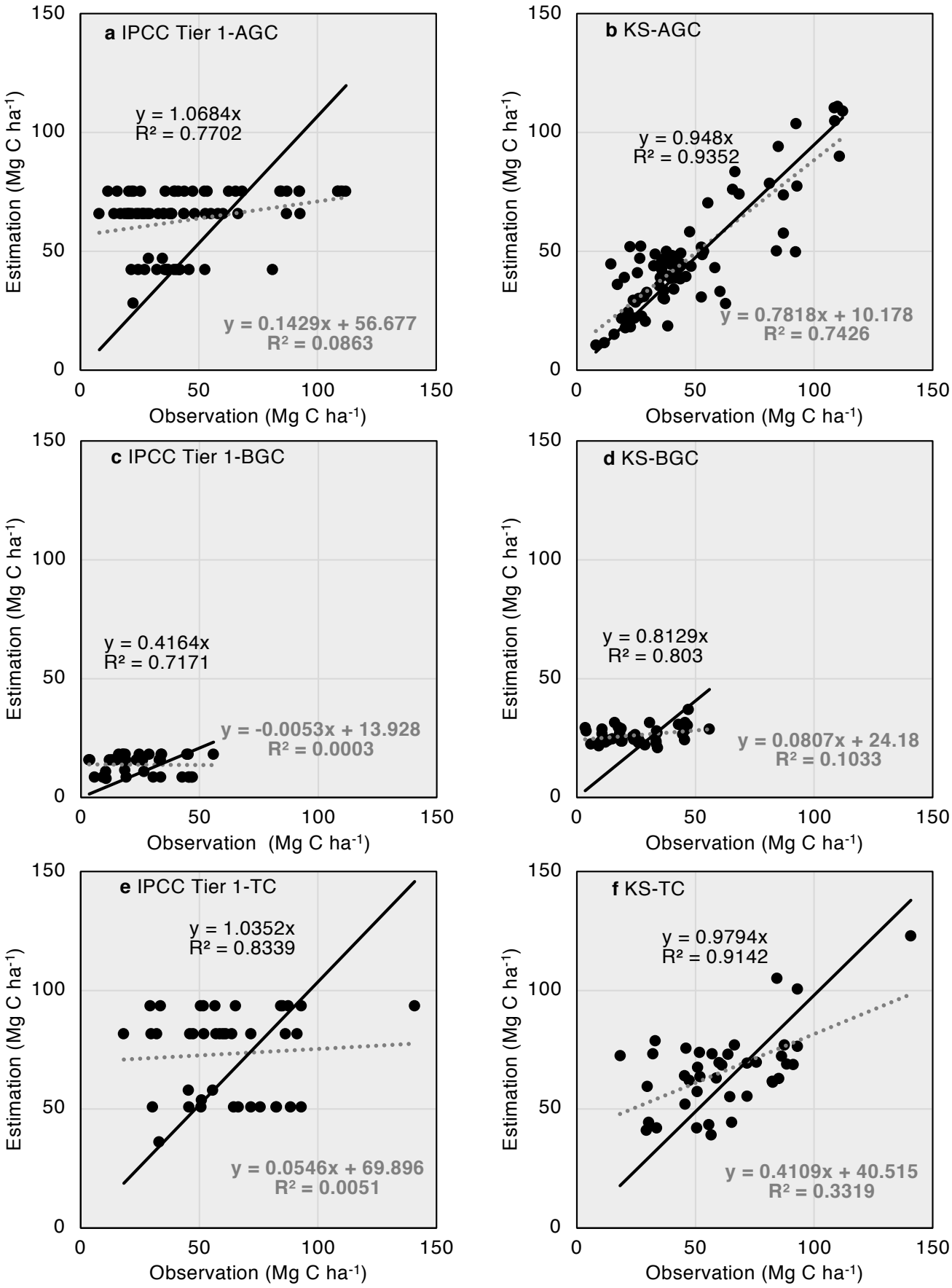
9.2. Operate the KS approach

9.3. Get the output of target variables

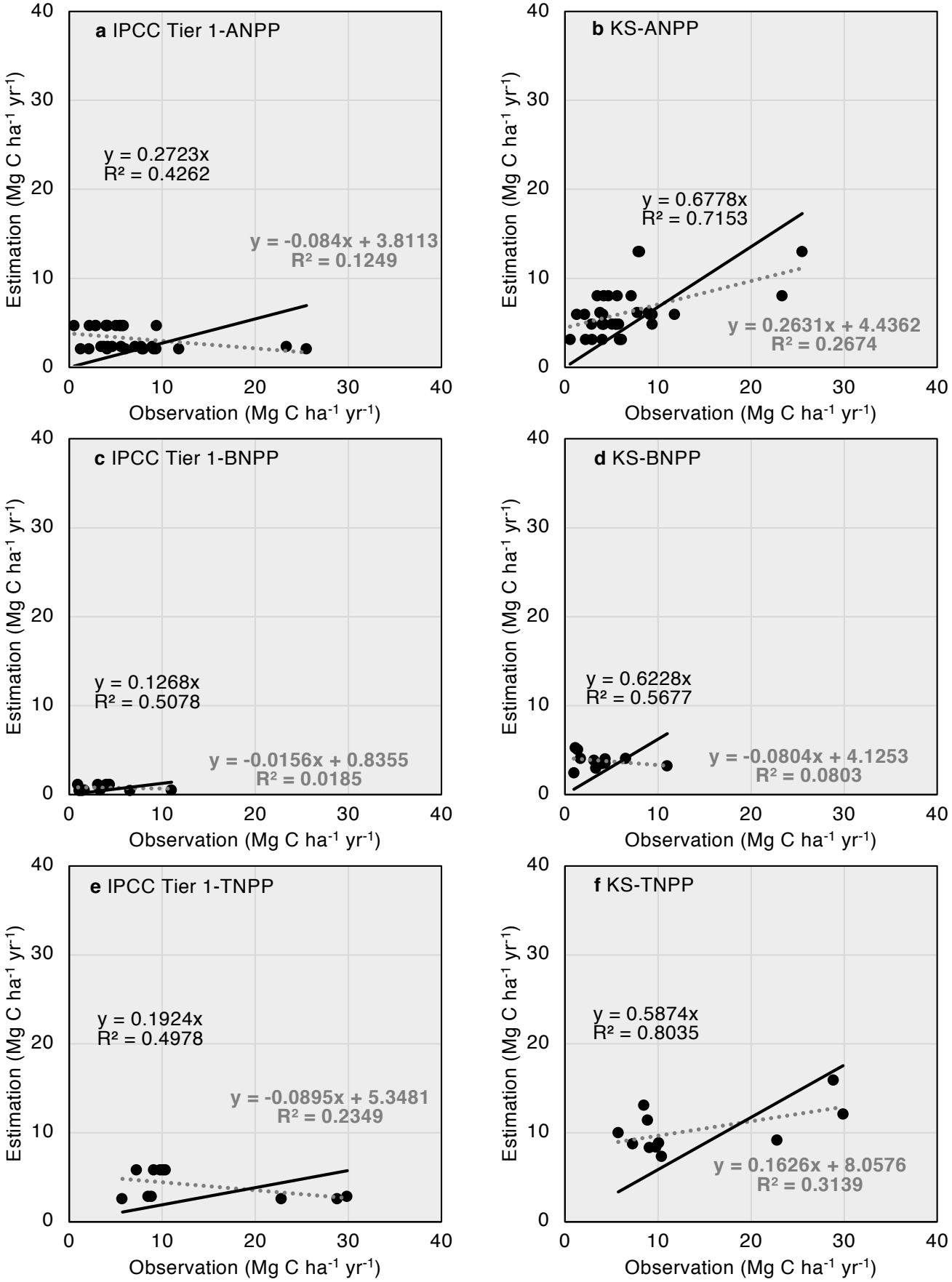
10. Final Visualisation

After running all the processes above, it is possible to visualise the result in table or box plot depended on the preference of the operator. In this study, we reveal the result in the main text of Table 2. The latest version of the process will continually updated on GitHub (https://github.com/gn03138868/WTF_sub-algorithm). If there are some bugs in the code script, please report to the first author or corresponding author. We sincerely thank you.

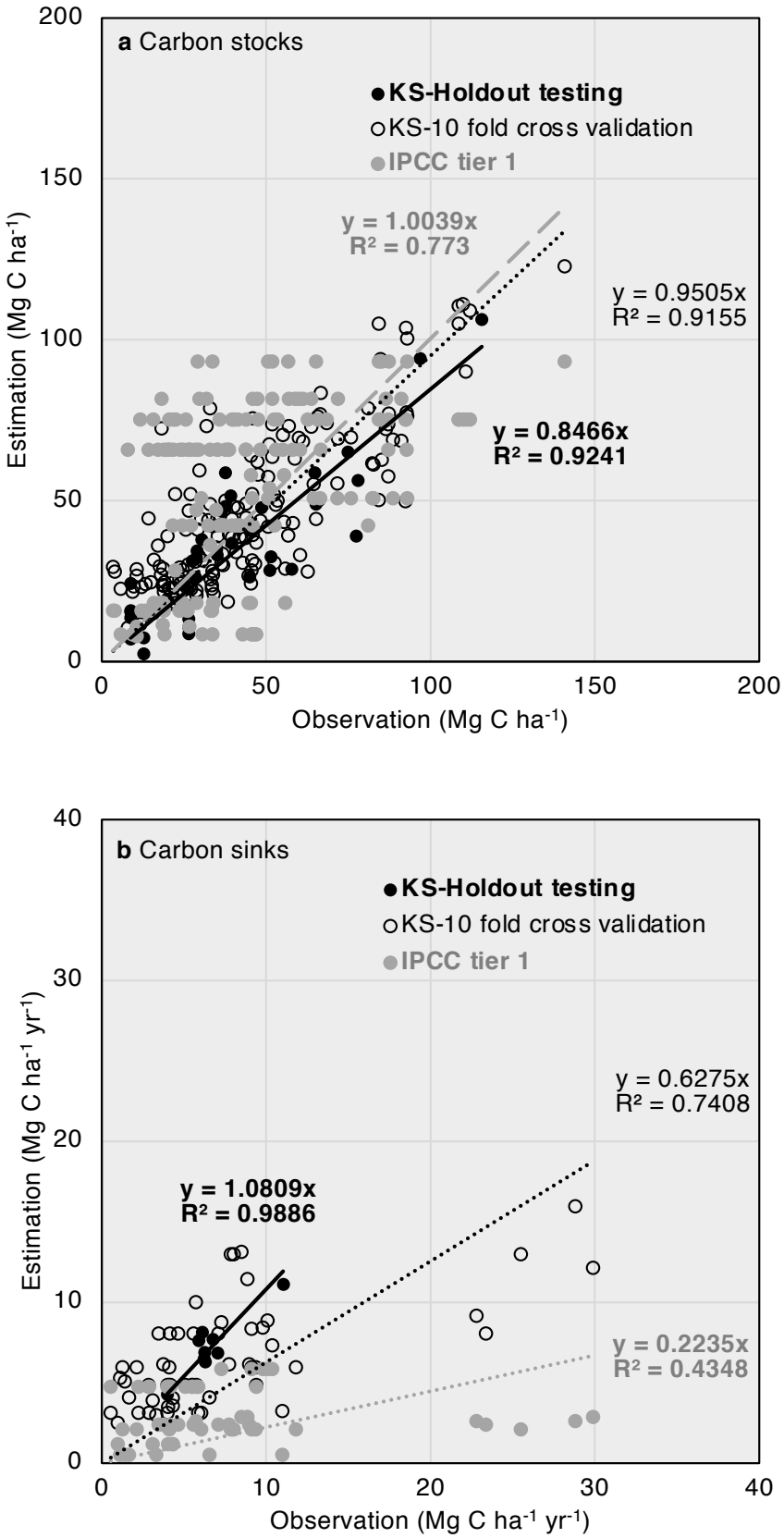
[11] Input lower than 1/k % observations when operate the KS approach once by once.



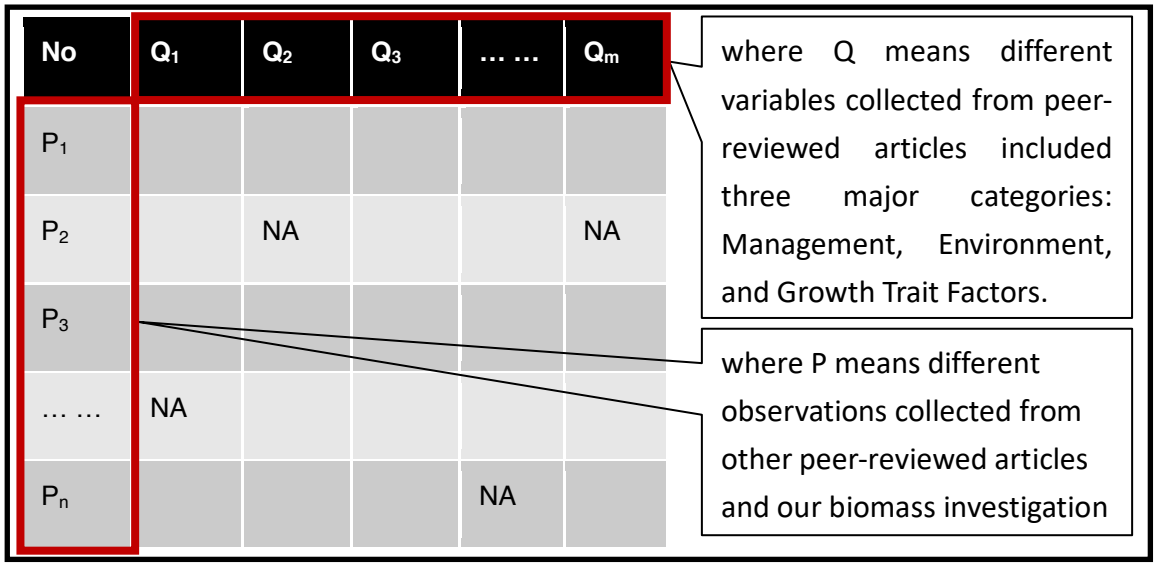
Supplementary Figure 1. Correlation between observation and estimation in forest carbon stocks of *P. edulis*. **a**, Correlation between observation and estimation of aboveground carbon (AGC) (n=77) via IPCC's Tier 1 method. **b**, Correlation between observation and estimation of AGC via knife set (KS) approach. **c**, Correlation between observation and estimation of belowground carbon (BGC) (n=39) via IPCC's Tier 1 method. **d**, Correlation between observation and estimation of BGC via KS approach. **e**, Correlation between observation and estimation of total carbon (TC) (n=40) via IPCC's Tier 1 method. **f**, Correlation between observation and estimation of TC via KS approach. The R value in black font represents the estimated accuracy (intercept is zero). The R value in grey font represents the estimated precision.



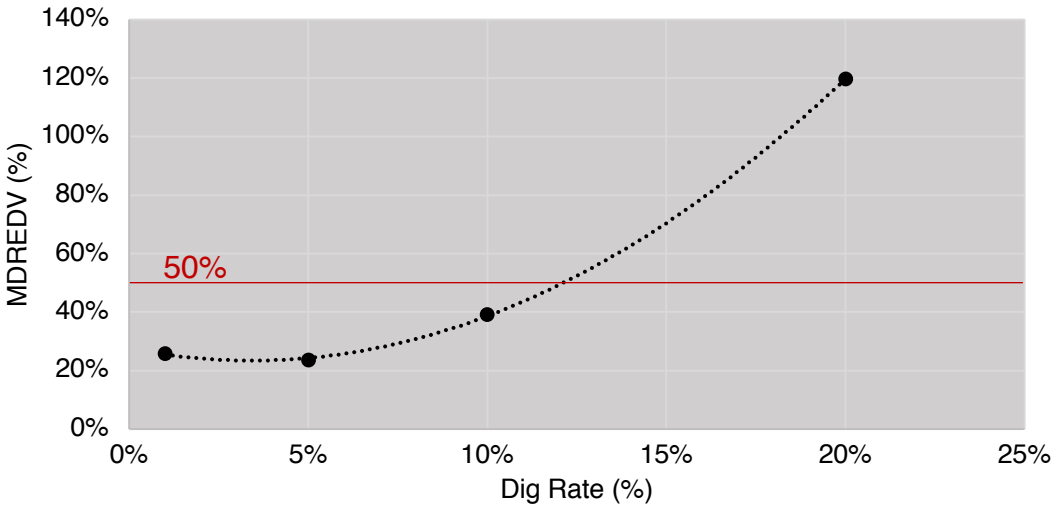
Supplementary Figure 2. Correlation between observation and estimation in forest carbon sinks of *P. edulis*. **a**, Correlation between observation and estimation of aboveground net primary production (ANPP) (n=33) via IPCC's Tier 1 method. **b**, Correlation between observation and estimation of ANPP via knife set (KS) approach. **c**, Correlation between observation and estimation of belowground net primary production (BNPP) (n=11) via IPCC's Tier 1 method. **d**, Correlation between observation and estimation of BNPP via KS approach. **e**, Correlation between observation and estimation of total net primary production (TNPP) (n=11) via IPCC's Tier 1 method. **f**, Correlation between observation and estimation of TNPP via KS approach. The R value in black font represents the estimated accuracy (intercept is zero). The R value in grey font represents the estimated precision.



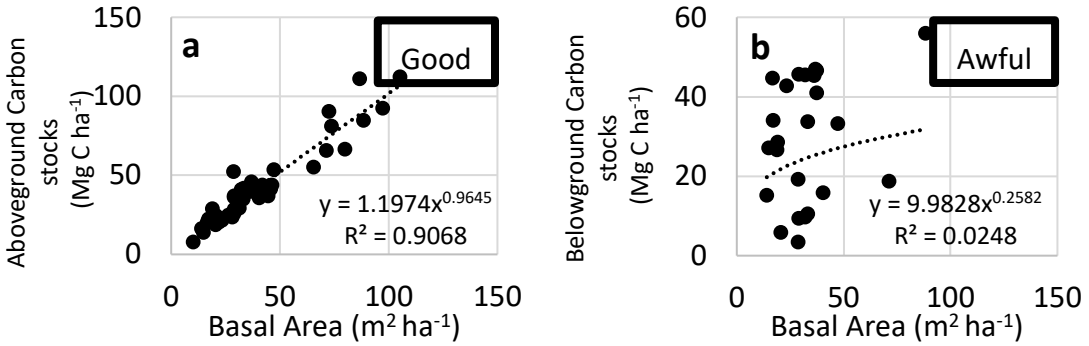
Supplementary Figure 3. Correlation between observation and estimation in forest carbon stocks and sinks of *P. edulis* in holdout testing, k-fold validation, and benchmark (IPCC tier 1 method). **a**, Correlation between observation and estimation of carbon stocks in holdout testing (n=38), 10-fold validation (n=156), and benchmark (IPCC tier 1 method) (n=156). **b**, Correlation between observation and estimation of carbon stocks in holdout testing (n=9), 10-fold validation (n=55), and benchmark (IPCC tier 1 method) (n=55). The R value represents the estimated accuracy (intercept is zero). The bold black font represents the holdout testing of the knife set (KS) approach. The black font represents the 10 fold cross validation of the KS approach. The grey font represents the benchmark (IPCC tier 1 method).



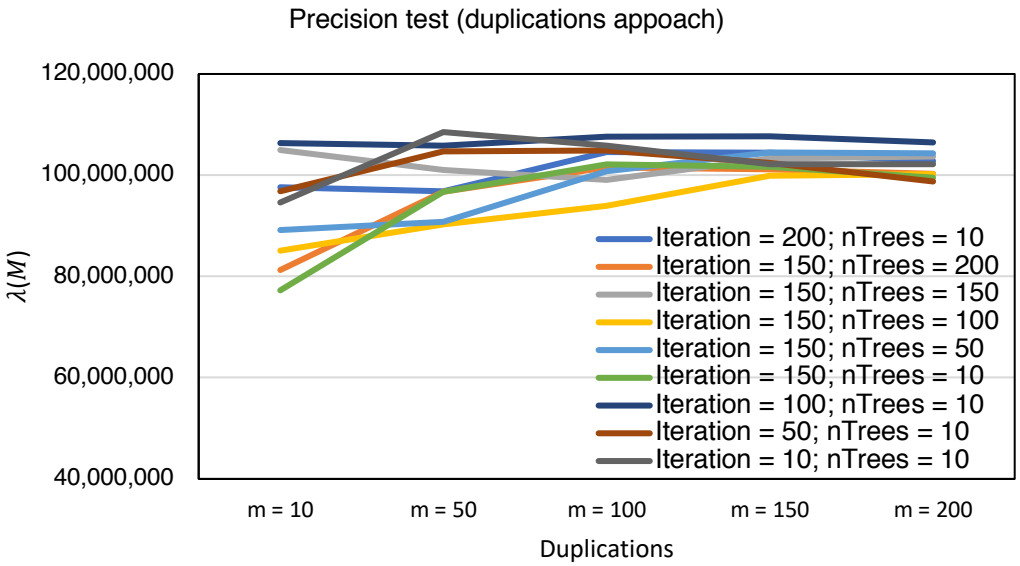
Supplementary Figure 4. Frameworks of raw data for further analysis of the knife set approach.



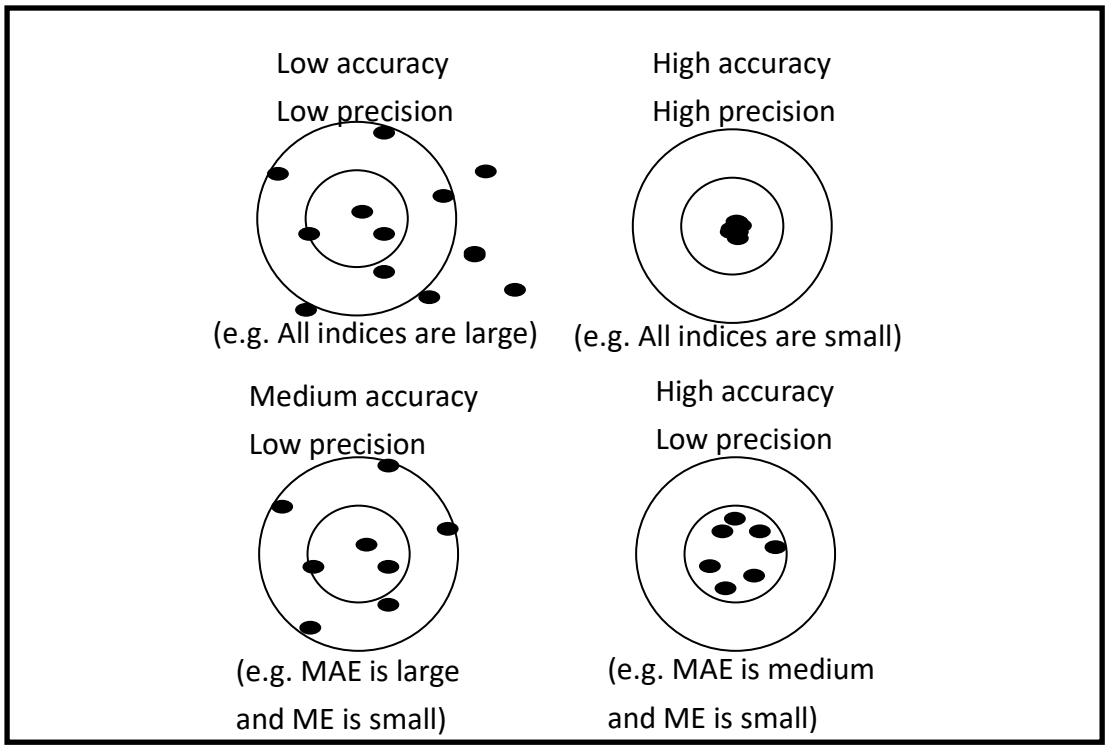
Supplementary Figure 5. Correlation between dig rate and mean absolute different ratio of each dug value (%) in the original datasheet estimated via random forest with Gibbs sampling. Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm for obtaining a sequence of observations which are approximated from a multivariate probability distribution when direct sampling is difficult.



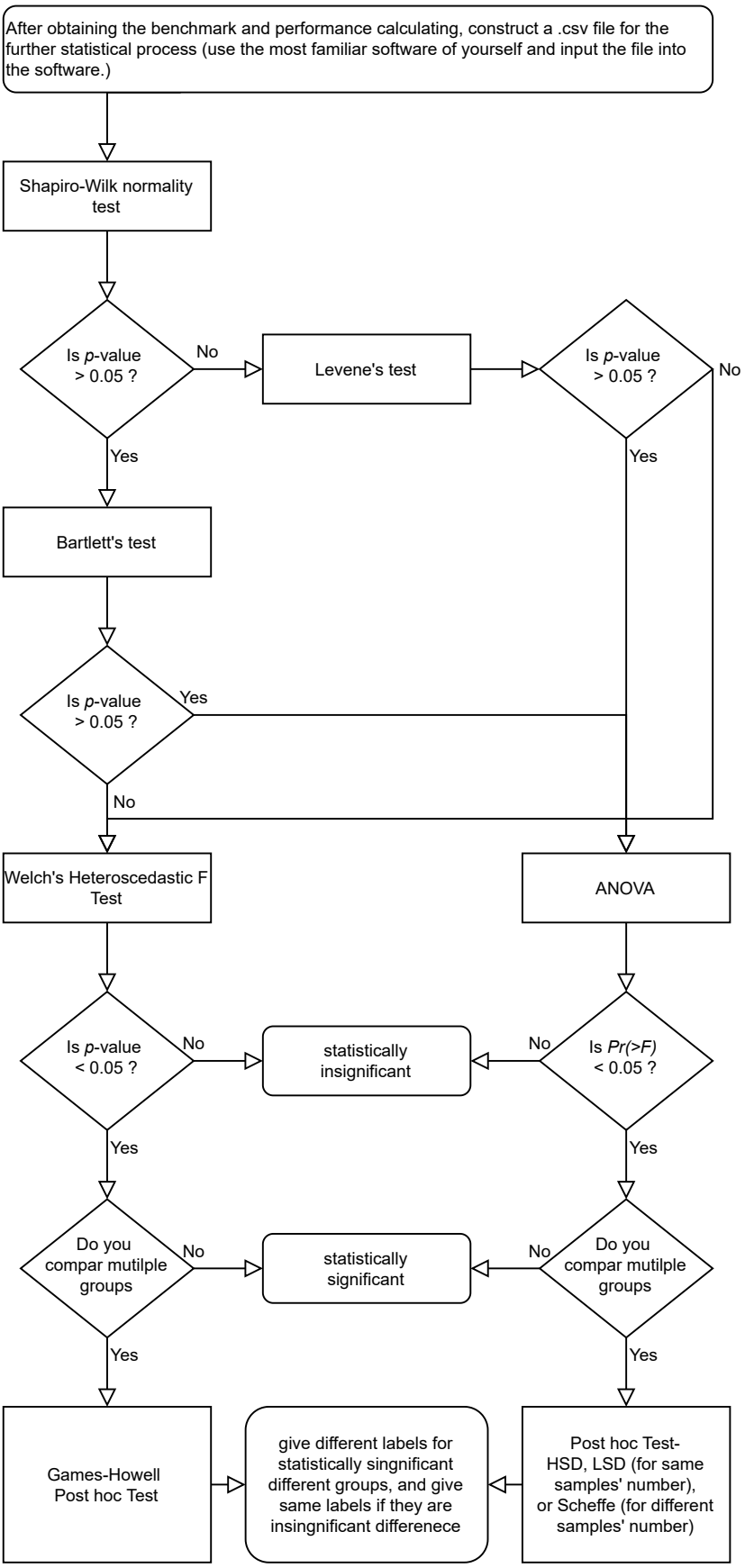
Supplementary Figure 6. Linear regression (LR) models for estimating aboveground carbon (AGC) and belowground carbon (BGC) stocks. a, A simple LR model for estimating AGC with the basal area as an independent variable. **b**, A LR model for estimating BGC with the basal area as an independent variable.



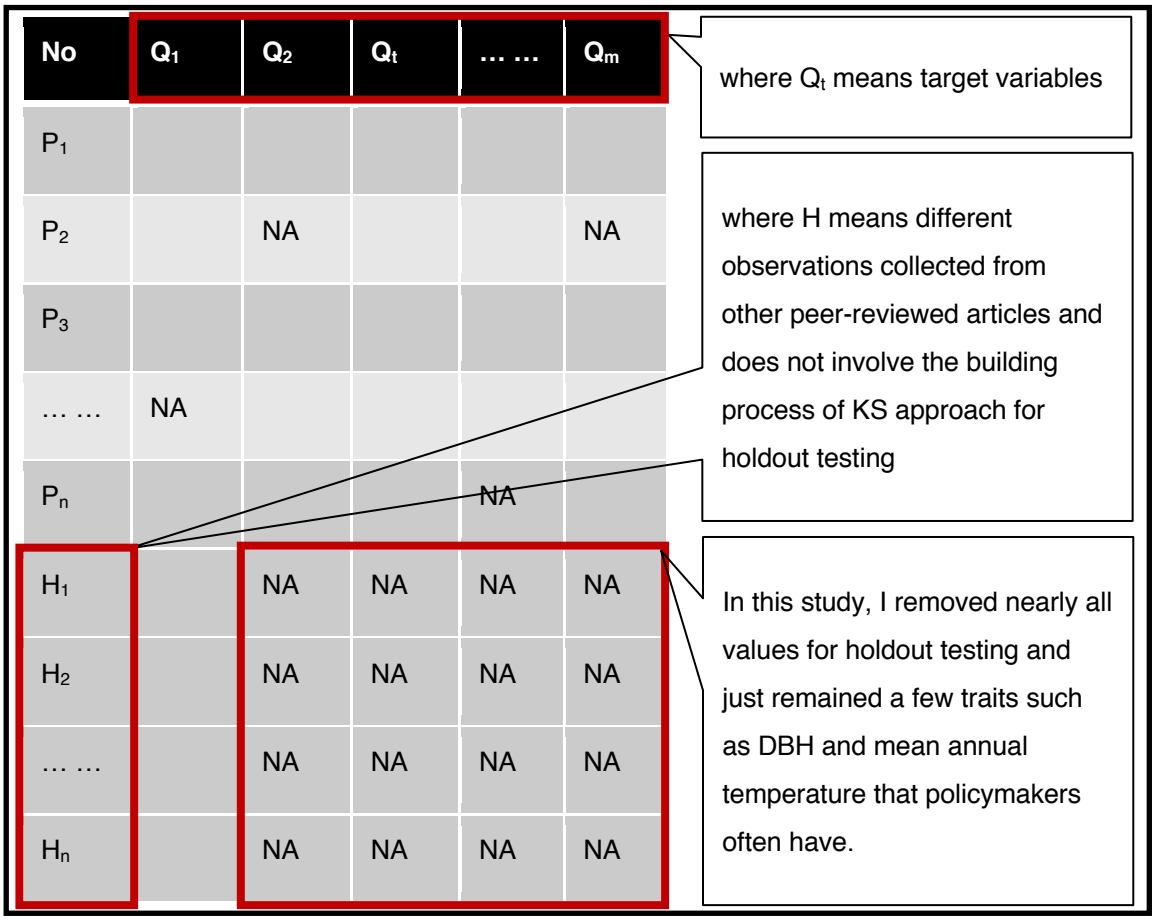
Supplementary Figure 7. Relationships between $\lambda(M)$ and the different combination of duplications, max iteration, and nTrees numbers in one of *k* folds.



Supplementary Figure 8. Meaning of indices for the performance of algorithms.



Supplementary Figure 9. Flowchart of statistical process of different methods.



Supplementary Figure 10. Frameworks of holdout dataset for testing the generalisation ability of the knife set approach.

Supplementary Table 1. Common terms and definitions of variables associated with photosynthesis, respiration, and carbon cycles in this study

category of factors	Variables	Units	Definition
Time	Beginning year	yr	The recorded starting year of each study in *P. edulis* forests.
Time	Finished year	yr	The recorded ending year of each study in *P. edulis* forests.
Management	Managed	[0,1]	A dummy independent variable which indecates *Phyllostachys edulis* (*P. edulis*) forests have been managed or not.
Management	Fertilised	[0,1]	A dummy independent variable which indecates *P. edulis* forests have been fertilised or not.
Management	Weeded and selective cut	[0,1]	A dummy independent variable which indecates common management methods operated in *P. edulis* forests.
Management	Shoot dug	[0,1]	A dummy independent variable which indecates *P. edulis* forests have been dug bamboo shoots or not.
Management	Clear cutting	[0,1]	A dummy independent variable which indecates *P. edulis* forests have been clear cut or not.
Management	Mixed with other forests	[0,1]	A dummy independent variable which indecates *P. edulis* forests have mixed with other forests or not.
Environment	Mean annual temperature	°C	Average of mean daily temperatures in a year. (Some of research use the Average of mean monthly temperatures in a year depending on their definition of weather stations.)
Environment	Warmth index		Warmth Index (WI) proposed by Kira (1945) is known as an index which is closely related to the distribution of vegetation. Warmth Index is calculated by the following equation. WI=Sum of (Tm-5) when Tm is above 5 °C (Tm: Monthly Mean Temperature)
Environment	Annual rainfall	mm yr^{-1}	The sum of daily rainfall in a year.
Environment	Snow	mm yr^{-1}	The sum of daily snowfall in a year.
Environment	Elevation	m	The height above sea level.

Environment	Relative humidity	%	A ratio of the amount of atmospheric moisture present relative to the amount that would be present if the air were saturated.
Environment	Sunshine duration	hr yr^{-1}	The sum of the time for which the direct solar irradiance exceeds 120 W m^{-2}.
Environment	Wind speed	m s^{-1}	The rate at which air is moving in a particular area.
Environment	Water content (soil)	kg kg^{-1}	A measure of the amount of water (mass) contained in a unit mass of soil.
Environment	pH (soil-KCl)		A measure of the acidity or basicity (alkalinity) of a soil by a diluted solution of potassium chloride (KCl) used in the analysis, instead of water (H_2O), which gives the pH (soil-KCl) value.
Environment	pH (soil-H_2O)		A measure of the acidity or basicity (alkalinity) of a soil by a diluted solution of water (H_2O) which gives the pH (soil-H_2O) value.
Environment	Total N (soil)	g kg^{-1}	The fraction of Nitrogen in a soil of *P. edulis* forests.
Environment	Total P (soil)	g kg^{-2}	The fraction of Phosphorus in a soil of *P. edulis* forests.
Environment	Total K (soil)	g kg^{-3}	The fraction of Potassium in a soil of *P. edulis* forests.
Environment	Available P_2O_5 (soil)	mg kg^{-1}	The Phosphorus pentoxide (P_2O_5) in soil which is available for absorption by roots in *P. edulis forests*.
Environment	Available SiO_2 (soil)	mg kg^{-1}	The Silicon dioxide (SiO_2) in soil which is available for absorption by roots in *P. edulis* forests.
Environment	Cation exchange capacity, CEC (soil)	cmole kg^{-1}	The total capacity of a soil to hold exchangeable cations in *P. edulis* forests.
Environment	K^+ (soil)	me $100g^{-1}$	The amount of Potassium ions in a soil which is available in *P. edulis* forests.
Environment	Ca^{2+} (soil)	me $100g^{-1}$	The amount of Calcium ions in a soil which is available in *P. edulis* forests.
Environment	Mg^{2+} (soil)	me $100g^{-1}$	The amount of Magnesium ions in a soil which is available in *P. edulis* forests.

Environment	Sand (soil)	%	The fraction of soil particles that the size range between 2 and 0.063 mm.
Environment	Silt (soil)	%	The fraction of soil particles that the size range between 0.063 and 0.002 mm.
Environment	Clay (soil)	%	The fraction of soil particles that the size is smaller than 0.002 mm.
Environment	N (litter)	kg ha^{-1}	The amount of nitrogen in a litter layer of *P. edulis* forests per hectare.
Environment	Ca (litter)	kg ha^{-1}	The amount of Calcium in a litter layer of *P. edulis* forests per hectare.
Environment	K (litter)	kg ha^{-1}	The amount of Potassium in a litter layer of *P. edulis* forests per hectare.
Environment	Mg (litter)	kg ha^{-1}	The amount of Magnesium in a litter layer of *P. edulis* forests per hectare.
Environment	P (litter)	kg ha^{-1}	The amount of Phosphorus in a litter layer of *P. edulis* forests per hectare.
Environment	Si (storage in aboveground biomass)	kg ha^{-1}	The amount of Silicon which storaged in aboveground biomass of *P. edulis* forests per hectare.
Environment	Si (storage in belowground biomass)	kg ha^{-1}	The amount of Silicon which storaged in belowground biomass of *P. edulis* forests per hectare.
Environment	Si (storage in soil)	kg ha^{-1}	The amount of Silicon which storaged in a soil of *P. edulis* forests per hectare.
Environment	Si (primary sink in Plant annually)	kg ha^{-1} yr^{-1}	The total amount of Silicon which sinked in *P. edulis* forests per hectare annually.
Environment	Si (net sink in Plant annually)	kg ha^{-1} yr^{-1}	The primary sink of Silicon in *P. edulis* forests minus Silicon which returns to soils
Environment	Si (return to soil)	kg ha^{-1} yr^{-1}	The primary sink of Silicon minus the net sink in *P. edulis* forests per hectare annually.

Environment	Relative luminosity	%	The relative brightness of spaces in *P. edulis* forests, normalised to 0 for darkest space in a black chamber and 1 for the highest luminosity in the open space outside the *P. edulis* forest.
Growth Trait	Culm density	culm ha^{-1}	The number of living culms in *P. edulis* forests per hectare.
Growth Trait	Mean DBH	cm	The mean of living culms' diameter at breast height (DBH) in a *P. edulis* forest.
Growth Trait	Mean height	m	The mean of living culms' height in a *P. edulis* forest.
Growth Trait	Basal area (b.a.)	m^2 ha^{-1}	The average amount of an area (usually a hectare) occupied by basal area of culms in *P. edulis* forests.
Growth Trait	LAI (Fisheye lens)	m^2 m^{-2}	The one-sided leaf area per unit ground surface area estimated by fish eye lens in *P. edulis* forests. [leaf area index (LAI) = leaf area / ground area]
Growth Trait	LAI (leaf area scanner)	m^2 m^{-2}	The one-sided leaf area per unit ground surface area estimated by scanner for cauculiating leaf area in *P. edulis* forests. [leaf area index (LAI) = leaf area / ground area]
Growth Trait	Leaves C	%	The carbon fraction of leaf dry matter in *P. edulis* forests.
Growth Trait	Branches C	%	The carbon fraction of branch dry matter in *P. edulis* forests.
Growth Trait	Culms C	%	The carbon fraction of culm dry matter in *P. edulis* forests.
Growth Trait	Fine roots C	%	The carbon fraction of fine root dry matter in *P. edulis* forests.
Growth Trait	Coarse root C	%	The carbon fraction of coarse root dry matter in *P. edulis* forests.
Growth Trait	Rhizomes C	%	The carbon fraction of rhizome dry matter in *P. edulis* forests.
Growth Trait	Stump C	%	The carbon fraction of stump dry matter in *P. edulis* forests.
Growth Trait	Soil C (0-10cm)	%	The carbon fraction of dry soil from the deepth of 0cm to 10cm in *P. edulis* forests.
Growth Trait	Soil C (10-30cm)	%	The carbon fraction of dry soil from the deepth of 10cm to 30cm in *P. edulis* forests.
Growth Trait	Foliage	Mg C ha^{-1}	The leaf biomass (carbon equivalent) in *P. edulis* forests.
Growth Trait	Branches	Mg C ha^{-1}	The branch biomass (carbon equivalent) in *P. edulis* forests.

Growth Trait	Culms	Mg C ha^{-1}	The culm biomass (carbon equivalent) in *P. edulis* forests.
Growth Trait	AGC	Mg C ha^{-1}	The aboveground biomass (carbon equivalent) in *P. edulis* forests.
Growth Trait	Root_shoot ratio		The ratio of underground divided by aboveground biomass (carbon equivalent).
Growth Trait	Roots	Mg C ha^{-1}	The root biomass (carbon equivalent) in *P. edulis* forests.
Growth Trait	Rhizomes	Mg C ha^{-1}	The rhizome biomass (carbon equivalent) in *P. edulis* forests.
Growth Trait	Stumps	Mg C ha^{-1}	The stump biomass (carbon equivalent) in *P. edulis* forests.
Growth Trait	BGC	Mg C ha^{-1}	The belowground biomass (carbon equivalent) in *P. edulis* forests.
Growth Trait	TC	Mg C ha^{-1}	The toral biomass (carbon equivalent) in *P. edulis* forests.
Growth Trait	Litter layer	Mg C ha^{-1}	The litter mass (carbon equivalent) in *P. edulis* forests.
Growth Trait	Soil carbon	Mg C ha^{-1}	The soil carbon storage in *P. edulis* forests.
Growth Trait	Undergrowth	Mg C ha^{-1}	The undergrowth biomass (carbon equivalent) in *P. edulis* forests.
Growth Trait	TEC	Mg C ha^{-1}	The total ecosystem carbon which includes above- and below-ground live components, dead biomass, and soil carbon.
Growth Trait	LNP	Mg C ha^{-1} yr^{-1}	The leaf net primary productivity in *P. edulis* forests.
Growth Trait	BNP	Mg C ha^{-1} yr^{-1}	The branch net primary productivity in *P. edulis* forests.
Growth Trait	CNP	Mg C ha^{-1} yr^{-1}	The culm net primary productivity in *P. edulis* forests.
Growth Trait	litterfall	Mg C ha^{-1} yr^{-1}	The litter fall in *P. edulis* forests.
Growth Trait	ANPP	Mg C ha^{-1} yr^{-1}	The aboveground net primary productivity in *P. edulis* forests. (equal to LNP+BNP+CNP+ litter fall)
Growth Trait	RoNP	Mg C ha^{-1} yr^{-1}	The root net primary productivity in *P. edulis* forests.
Growth Trait	RhNP	Mg C ha^{-1} yr^{-1}	The rhizome net primary productivity in *P. edulis* forests.
Growth Trait	SNP	Mg C ha^{-1} yr^{-1}	The stump net primary productivity in *P. edulis* forests.
Growth Trait	BNPP	Mg C ha^{-1} yr^{-1}	The belowground net primary productivity in *P. edulis* forests. (equal to RoNP+RhNP+SNP)
Growth Trait	TNPP	Mg C ha^{-1} yr^{-1}	The total net primary productivity in *P. edulis* forests. (equal to ANPP+BNPP)

Growth Trait	Soil respiration (SR)	Mg C ha^{-1} yr^{-1}	The respiration of soil in *P. edulis* forests. (equal to belowground autotrophic respiration + heterotrophic respiration)
Growth Trait	Heterotrophic respiration (HR)	Mg C ha^{-1} yr^{-1}	Total respiration excluded autotrophic respiration. (also equal to TNPP-NEP)
Growth Trait	Net ecosystem production (NEP)	Mg C ha^{-1} yr^{-1}	The difference between organic carbon fixed by photosynthesis and total ecosystem respiration including autotrophic and heterotrophic respiration. (equal to gross primary production minus ecosystem respiration, or to TNPP minus heterotrophic respiration)

Supplementary Table 2. Pseudocode of the wandering through the random forests (WTF) sub-algorithm in the knife set (KS) approach

Algorithm I the WTF sub-algorithm in the KS

1: **initialise** $\mathbf{X}^s \sim \boldsymbol{X}^{(0)}$

2: **for** duplicate s = 0, 1, 2, … **do**

1: **initialise** $\boldsymbol{X}^{(0)} \sim q(\boldsymbol{X})$

2: **for** iteration i = 0, 1, 2, … **do**

1. $X_1^{(i)} \sim p(\boldsymbol{X}_1 = x_1 | \boldsymbol{X}_2 = x_2^{(i-1)}, \boldsymbol{X}_3 = x_3^{(i-1)}, \ldots, \boldsymbol{X}_D = x_D^{(i-1)})$

1. **procedure** random forest
2. **for** 1 to T **do**
3. Bootstrap **D** for forming a new **Di**

[The rest of points from **D** for out of bag (OOB) test]

4. Build full classification and regression trees on **Di**
5. **end for**
6. **ensemble** all T trees
7. **end procedure**

2. $X_2^{(i)} \sim p(X_2 = x_2 | X_1 = x_1^{(i)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$

1. **procedure** random forest
2. **for** 1 to T **do**
3. Bootstrap **D** for forming a new **Di**

[The rest of points from **D** for out of bag (OOB) test]

4. Build full classification and regression trees on **Di**
5. **end for**
6. **average** all T trees
7. **end procedure**

3. …

4. $X_D^{(i)} \sim p(\boldsymbol{X}_D = x_D | \boldsymbol{X}_1 = x_1^{(i)}, \boldsymbol{X}_2 = x_2^{(i)}, \ldots, \boldsymbol{X}_D = x_{D-1}^{(i)})$

1. **procedure** random forest
2. **for** 1 to T **do**
3. Bootstrap **D** for forming a new **Di**

[The rest of points from **D** for out of bag (OOB) test]

4. Build full classification and regression trees on **Di**
5. **end for**
6. **average** all T trees
7. **end procedure**

3. **end for**

3: **average** all $\mathbf{X}^s$

4: **end for**

Supplementary Table 3. Ratio of below-ground biomass to above-ground biomass (R) in *Phyllostachys edulis* forests worldwide

Domain	Ecological zone	Above-ground biomass	R [tonne root d.m. (tonne shoot d.m.)$^{-1}$]	References
Subtropical	Subtropical humid forest	above-ground biomass < 80 tonnes ha^{-1}	0.27 (0.13-0.34)	Wang et al. (2009); Tang et al. (2015); Zhuang et al. (2015); This study (2021)
		above-ground biomass > 80 tonnes ha^{-1}	0.51 (0.34-0.63)	Huang et al. (1993); Wang et al. (2013); Zhuang et al. (2015)
	Subtropical mountain systems		0.80 (0.24-1.98)	Li et al. (2006); Chen et al. (2011); Fan et al. (2011); Chen et al. (2012); Wang et al. (2013); Lin et al. (2017)
Temperate	Temperate oceanic forest	above-ground biomass < 80 tonnes ha^{-1}	1.32 (0.44-2.18)	Park and Ryu (1996); Zhang et al. (2005); Goto et al. (2008); Umemura and Takenaka (2014)
		above-ground biomass > 80 tonnes ha^{-1}	0.48 (0.28-0.66)	Isagi et al. (1997); Umemura and Takenaka (2014); Fukushima et al. (2015); This study (2021)
	Temperate continental forest		0.37 (0.09-0.92)	Hwang et al. (2005); Lee et al. (2012); Kim et al. (2018)

Supplementary Table 4. Tier 1 estimated biomass values of *P. edulis* obtained from Supplementary Dataset 1 for improving table 4.7-4.10 (values are approximate; use only for Tier 1) in 2006 IPCC Guidelines for National Greenhouse Gas Inventories

Climate domain	Ecological zone	Above-ground biomass in *Phyllostachys edulis* forests (tonnes d.m. ha^{-1})	Above-ground biomass in forests plantation of *Phyllostachys edulis* (tonnes d.m. ha^{-1})	Above-ground biomass growth in natural *Phyllostachys edulis* forests (tonnes d.m. ha^{-1} yr^{-1})	Above-ground biomass growth in forests plantation of *Phyllostachys edulis* (tonnes d.m. ha^{-1} yr^{-1})
Subtropical	Subtropical humid forest	110.13 (28.58-221.35)	69.00 (16.00-185.65)	6.60 (1.09-12.14)	10.30 (5.71-18.80)
	Subtropical mountain systems	90.63 (43.30-162.27)	71.47 (44.78-105.11)	13.04 (7.50-17.92)	17.09 (6.90-46.71)
Temperate	Temperate continental forest	57.77	73.37 (52.66-97.80)	-	-
	Temperate oceanic forest	125.02 (23.31-296.50)	114.07 (45.20-217.30)	12.60 (2.6-23.6)	27.59 (15.72-51.00)

Supplementary References

1. van Buuren, S. and Groothuis-Oudshoorn, K. mice: multivariate imputation by chained equations in R. *J. Stat. Softw.*, **45(3)**, 1-67 (2011). https://doi.org/10.18637/jss.v045.i03
2. Shah, A. D., Bartlett, J. W., Carpenter, J., Nicholas, O. and Hemingway, H. Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study. *Am. J. Epidemiol.*, **179(6)**, 764–774 (2014). https://doi.org/10.1093/aje/kwt312
3. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, **2(11)**, 559-572 (1901). http://pca.narod.ru/pearson1901.pdf
4. Spearman, C. The proof and measurement of association between two things. *Am. J. Psychol.*, **15**, 72–101 (1904). https://www.jstor.org/stable/pdf/1422689.pdf
5. Kendall, M. G. A new measure of rank correlation. *Biometrika*, **30(1-2)**, 81–93 (1938). https://watermark.silverchair.com/30-1-2-81.pdf
6. Royston, P. Algorithm AS 181: The W test for Normality. *J. R. Stat. Soc. Series C*, **31**, 176-180 (1982). https://doi.org/10.2307/2347986
7. Breiman, L. Random Forests. *Mach. Learn.* **45**, 5–32 (2001). https://doi.org/10.1023/A:1010933404324