PNAS
www.pnas.org

Supplementary Information for

Crop variety management for climate adaptation supported by citizen science

Jacob van Etten*, Kauê de Sousa, Amilcar Aguilar, Mirna Barrios, Allan Coto, Matteo Dell'Acqua, Carlo Fadda, Yosef Gebrehawaryat, Jeske van de Gevel, Arnab Gupta, Afewerki Y. Kiros, Brandon Madriz, Prem Mathur, Dejene K. Mengistu, Leida Mercado, Jemal Nurhisen Mohammed, Ambica Paliwal, Mario Enrico Pè, Carlos F. Quiros, Juan Carlos Rosas, Neeraj Sharma, S.S. Singh, Iswhar S. Solanki and Jonathan Steinke

***Corresponding author**
E-mail: j.vanetten@cgiar.org

This PDF file includes:

Supporting Information Text

Supplementary Methods

Crop trials. We focused on farmers' overall evaluation of varieties to derive recommendations. In all cases, overall performance ranking had a strong correlation with farmers' ranking of yield (Kendall rank correlation coefficient of 0.96-0.97). Varieties were included in the trials on the basis of previous on-farm trials (India), consultation with breeders (Nicaragua) and a pilot participatory on-farm trial (Ethiopia (1)). Details of the varieties included in the trials are given as part of the full dataset on Dataverse (2). SI Appendix Fig. S5 and Table S2 contain details on the questionnaire applied.

Environmental data. We used free, publicly available environmental data with coverage across the tropics to make the three case studies comparable and to ensure the methods can be applied to future studies across the tropics. For rainfall, we used the Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS) dataset, which provides daily precipitation estimates with a 0.05 degree resolution, based on satellite data and weather stations (3). To obtain day and night temperatures for each trial location during the growth period of the crop, composites of 8-day land surface temperature MODIS (MYD11A2) were used (4). To reduce noise and fill gaps, the adaptive Savitzky-Golay filter was used with a window size of 12 for the polynomial smoothing (5).

Fourteen environmental variables were extracted for the vegetative, flowering and grain filling periods starting on the planting date of each observation point. These climatic variable were previously used for the environmental analysis of wheat trial data (6). Variables extracted from MODIS (4) and CHIRPS (3) data were: (i) maxDT, maximum day temperature (°C); (ii) minDT, minimum day temperature (°C); (iii) maxNT, maximum night temperature (°C); (iv) minNT, minimum night temperature (°C); (v) DTR, diurnal temperature range, mean difference between day temperature and night temperature (°C); (vi) SU, summer days, number of days with maximum temperature 30 °C; (vii) TR, tropical nights, number of nights with maximum temperature $>$ 25 °C; (viii) MLDS, maximum length of consecutive dry days ($<$ 1 mm); (ix) MLWS, maximum length of consecutive wet days ($\geq$ 1 mm); (x) R5mm, days with rainfall between 5 and 10 mm; (xi) R20mm, days with rainfall higher than 20 mm; (xii) SDII, simple rainfall intensity index (mean of wet days / total rainfall); (xiii) Rx1day, maximum 1-day rainfall (mm); and (xiv) Rx5day, maximum 5-day rainfall (mm).

To represent geographic structure, we included the longitude and latitude of trial locations in the analysis, as well as longitude+latitude and longitude-latitude (coordinates on 45° rotated axes). Soil data was obtained from the Harmonized World Soil Database (7).

Data analysis. For recursive partitioning with the Plackett-Luce model we used the covariates described in the previous section. Each PLT model selected optimal variables to partition the data, using a cut-off value of p = 0.01 for variable selection and a minimal group size of 30 percent of the total dataset of each country. We evaluated the PLT models with 10-fold cross-validation. We used cross-validated (out-of-sample) deviance rather than the equivalent Akaike Information Criterion (AIC) values, because cross-validated deviance better reflects the complexity of the full modeling procedure (8). To generate Table 1, we calculated pseudo-R^2 values that represent the relative reduction in deviance of the model (9).

To create models that provide generalizable predictions across seasons, we used blocked cross-validation (with seasons as blocks) combined with a forward variable selection procedure (10). For each season used for testing, we predicted variety scores with a PLT model using the data from the other seasons as training data and calculated the deviance of this prediction. For both model training and testing, seasonal climate variables were used, which is equivalent to having a perfect seasonal forecast for the test season. We used the deviance values of each validation season to calculate an Akaike weight, which is the probability that a given variable combination represents the best model (11). We combined the Akaike weights across the testing seasons calculating a weighted mean, using as weights the square root of the sample size of each testing season (12). We performed forward variable selection, using this combined Akaike weight as our selection criterion.

We evaluated if the models obtained with the variable selection procedure retained predictive power when no seasonal climate data was available for the testing season. We prepared representative seasonal scenarios of past climate conditions of each site by extracting the last 15 years of seasonal climate data derived from the MODIS dataset (2002-2016). We determined ten planting dates for each growing season as the midpoints of ten equiprobable quantile intervals estimated with a survival random forest fitted on the planting dates of the trial data using climatic covariates (13). We predicted variety performance for 15 seasons * 10 planting dates = 150 seasonal scenarios. We averaged variety probability of winning across these scenarios for each planting date interval, excluding the seasons used as testing data. We then compared these predictions with variety performance in all observations, matching with the planting date interval. We followed an equivalent procedure to generate a model without covariates and a perfect forecast (using the observed seasonal climate conditions) (Table 2). To compare the models, we calculated a weighted average of pseudo-R^2 (deviance reduction) values across testing seasons (9), using again the square root of the sample size as weights (12).

To examine the origin of cold tolerance in Ethiopian farmer varieties, we created a group with cold-adapted and cold-sensitive varieties from the results of the generalizable PLT model shown in Fig. 2a (excluding modern varieties and farmer varieties from unknown origin). Farmer varieties were classified as cold-adapted if above-average in node 2 and below-average in node 3 ($n = 10$) and as cold-sensitive if below-average in node 2 and above-average in node 3 ($n = 16$). We performed a one-tailed, unequal-variance t-test to determine differences in elevation between the two groups (14).

Organizing the datasets relied on R packages Matrix (15), tidyverse (16) and RCurl (17). Statistical analysis was performed using packages caret (18), partykit (19), PlackettLuce (20), survival (21) and qvcalc (22). Forward variable selection depended on packages abind (23), doParallel (24) and foreach (25). Geospatial analysis was done with packages alphahull (26), dismo (27), gstat (28), raster (29) and rgeos (30). To produce Fig. 2, package ggplot2 (31) was used.

References

1. Mancini C, et al. (2017) Joining smallholder farmers' traditional knowledge with metric traits to select better varieties of Ethiopian wheat. *Scientific Reports* 7(1):9120.
2. van Etten J, et al. (2018) Replication data for: "Crop variety management for climate adaptation supported by citizen science" [Harvard Dataverse] (https://doi.org/10.7910/DVN/4ICF6W).
3. Funk C, et al. (2015) The climate hazards infrared precipitation with stations—a new environmental record for monitoring extremes. *Scientific Data* 2:150066.
4. Wan Z, Hook S, Hulley G (2015) MYD11A1 MODIS/Aqua Land Surface Temperature/Emissivity 8-Day L3 Global 1km SIN Grid V006 [data set] (http://dx.doi.org/10.5067/MODIS/MYD11A2.006).
5. Chen J, et al. (2004) A simple method for reconstructing a high-quality NDVI time-series data set based on the Savitzky–Golay filter. *Remote Sensing of Environment* 91(3):332–344.
6. Kehel Z, Crossa J, Reynolds M (2016) Identifying Climate Patterns during the Crop-Growing Cycle from 30 Years of CIMMYT Elite Spring Wheat International Yield Trials in *Applied Mathematics and Omics to Assess Crop Genetic Resources for Climate Change Adaptive Traits*, eds. Bari A, Damania AB, Mackay M, Dayanandan S. (CRC Press), pp. 151–174.
7. Nachtergaele F, et al. (2009) Harmonized world soil database. *Wageningen: ISRIC*.
8. Elder IV JF (2003) The generalization paradox of ensembles. *Journal of Computational and Graphical Statistics* 12(4):853–864.
9. Agresti A (2003) *Categorical Data Analysis*. (John Wiley & Sons).
10. Meyer H, Reudenbach C, Hengl T, Katurji M, Nauss T (2018) Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation. *Environmental Modelling & Software* 101:1–9.
11. Wagenmakers EJ, Farrell S (2004) AIC model selection using Akaike weights. *Psychonomic Bulletin & Review* 11(1):192–196.
12. Whitlock MC (2005) Combining probability from independent tests: the weighted Z-method is superior to Fisher's approach. *Journal of Evolutionary Biology* 18(5):1368–1373.
13. Therneau TM, Grambsch PM (2000) *Modeling Survival Data: Extending the Cox Model*. (Springer-Verlag New York, New York), 1 edition, p. 350.
14. Welch BL (1947) The generalization of 'Student's' problem when several different population variances are involved. *Biometrika* 34(1/2):28–35.
15. Bates D, Maechler M (2017) *Matrix: Sparse and Dense Matrix Classes and Methods*. R package version 1.2-12.
16. Wickham H (2017) *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1.
17. Lang DT, the CRAN team (2018) *RCurl: General Network (HTTP/FTP/...) Client Interface for R*. R package version 1.95-4.10.
18. Kuhn M, et al. (2018) *caret: Classification and Regression Training*. R package version 6.0-79.
19. Hothorn T, Zeileis A (2015) partykit: a modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research* 16:3905–3909.
20. Turner H, Kosmidis I, Firth D, van Etten J (2017) PlackettLuce: Plackett-Luce models for rankings.
21. Therneau TM (2015) *A Package for Survival Analysis in S*. version 2.38.
22. Firth D (2017) *qvcalc: Quasi Variances for Factor Effects in Statistical Models*. R package version 0.9-1.
23. Plate T, Heiberger R (2016) *abind: Combine Multidimensional Arrays*. R package version 1.4-5.
24. Microsoft Corporation, Weston S (2017) *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.11.
25. Microsoft Corporation, Weston S (2017) *foreach: Provides Foreach Looping Construct for R*. R package version 1.4.4.
26. Pateiro-Lopez B, Rodriguez-Casal A (2016) *alphahull: Generalization of the Convex Hull of a Sample of Points in the Plane*. R package version 2.1.
27. Hijmans RJ, Phillips S, Leathwick J, Elith J (2017) *dismo: Species Distribution Modeling*. R package version 1.1-4.
28. Gräler B, Pebesma E, Heuvelink G (2016) Spatio-temporal interpolation using gstat. *The R Journal* 8(1):204–218.
29. Hijmans RJ, van Etten J (2017) *raster: Geographic Data Analysis and Modeling*. R package version 2.6-7.
30. Bivand R, Rundel C (2018) *rgeos: Interface to Geometry Engine - Open Source ('GEOS')*. R package version 0.3-28.
31. Wickham H (2009) *ggplot2: Elegant Graphics for Data Analysis*. (Springer-Verlag New York).

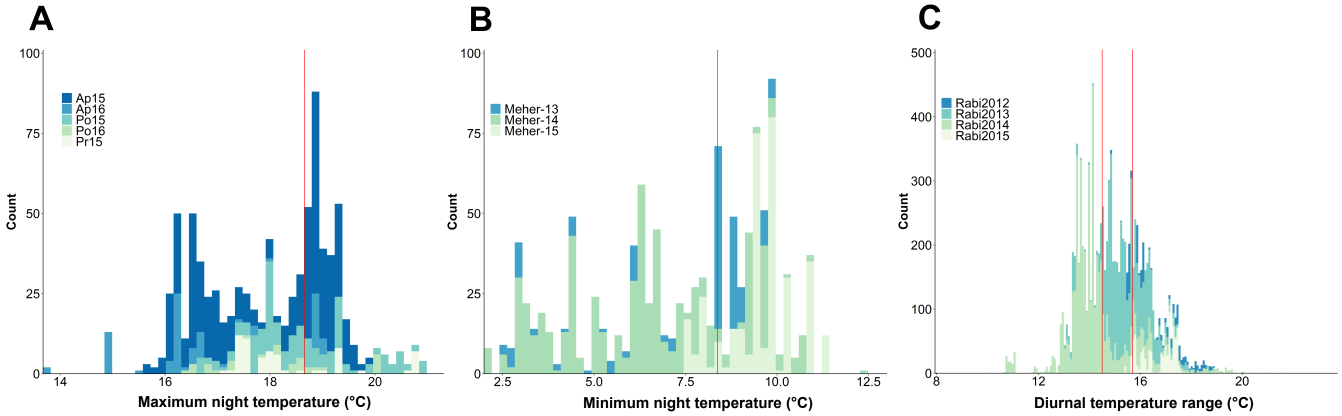


Fig. S1. Distribution of retained variables for the generalizable Plackett-Luce models among different seasons for *(A)* Nicaragua, *(B)* Ethiopia, and *(C)* India. Vertical red lines represent the cutting off point defined for classification of nodes in the Plackett-Luce trees algorithm.

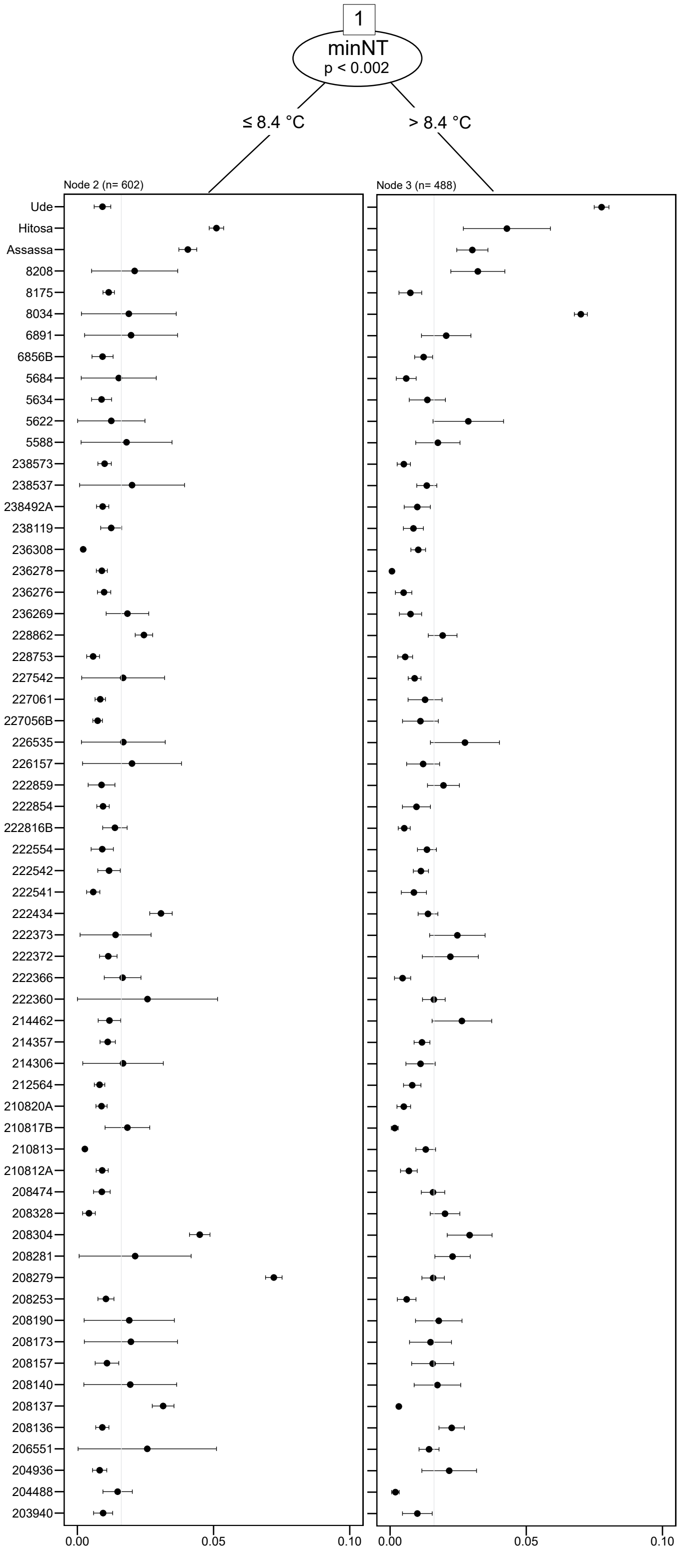


Fig. S2. Plackett-Luce Tree of tricot trial data and associated climatic data for durum wheat in Ethiopia. Intervals show quasi-standard errors. The grey vertical lines indicate the average probability of winning (1 / number of varieties). In this case, the model selected minNT, the minimum night temperature (°C) during the vegetative period, as the covariate.

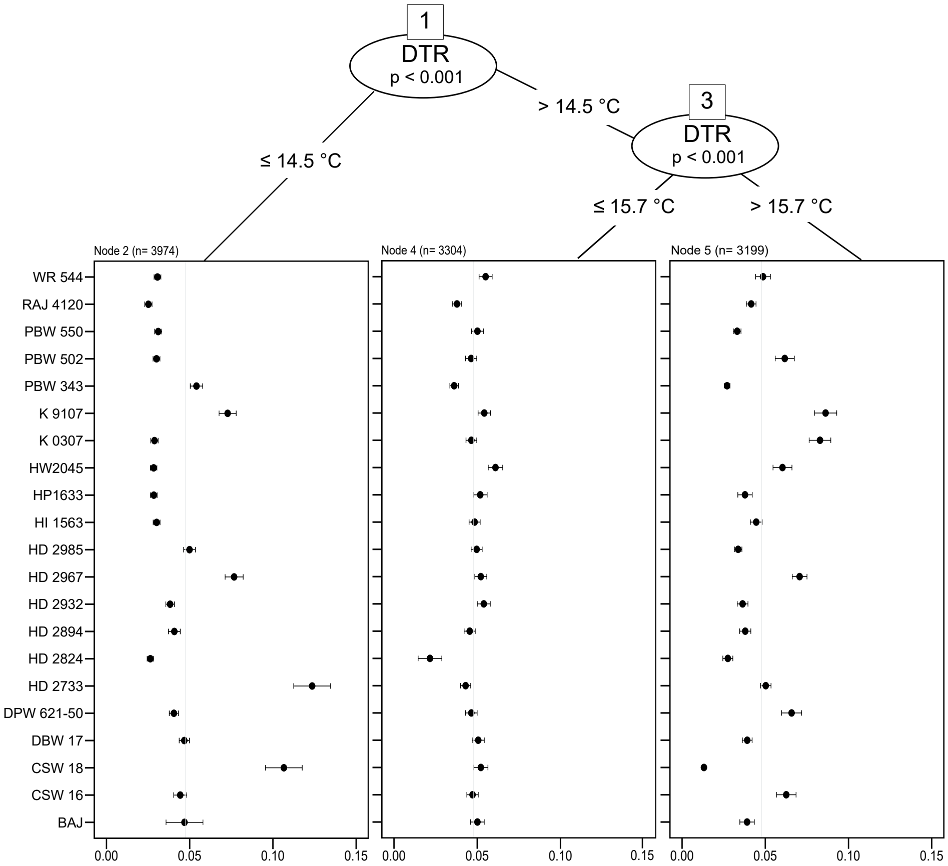


Fig. S3. Plackett-Luce Tree of tricot trial data and associated climatic data for bread wheat in India. Intervals show quasi-standard errors. The grey vertical lines indicate the average probability of winning (1 / number of varieties). In this case, the model selected DTR, the diurnal temperature range (°C) during the vegetative period.

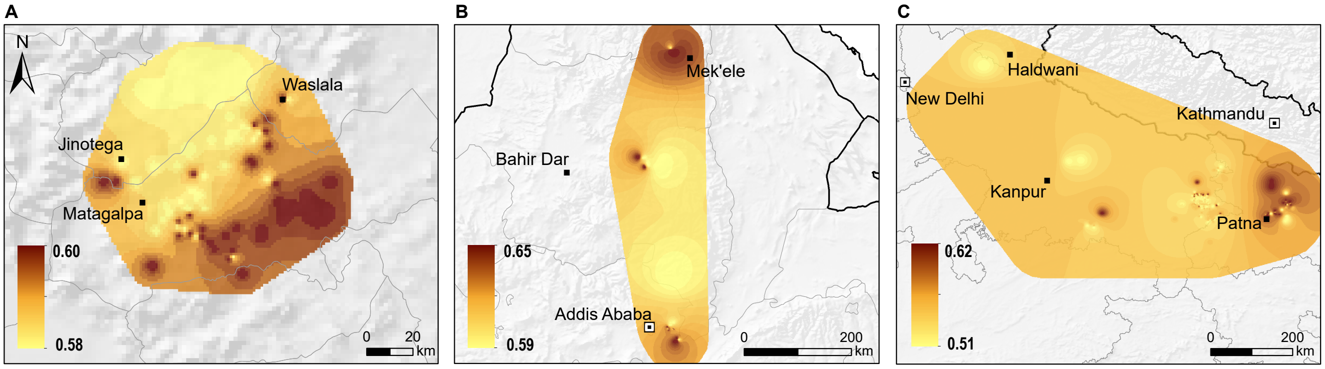
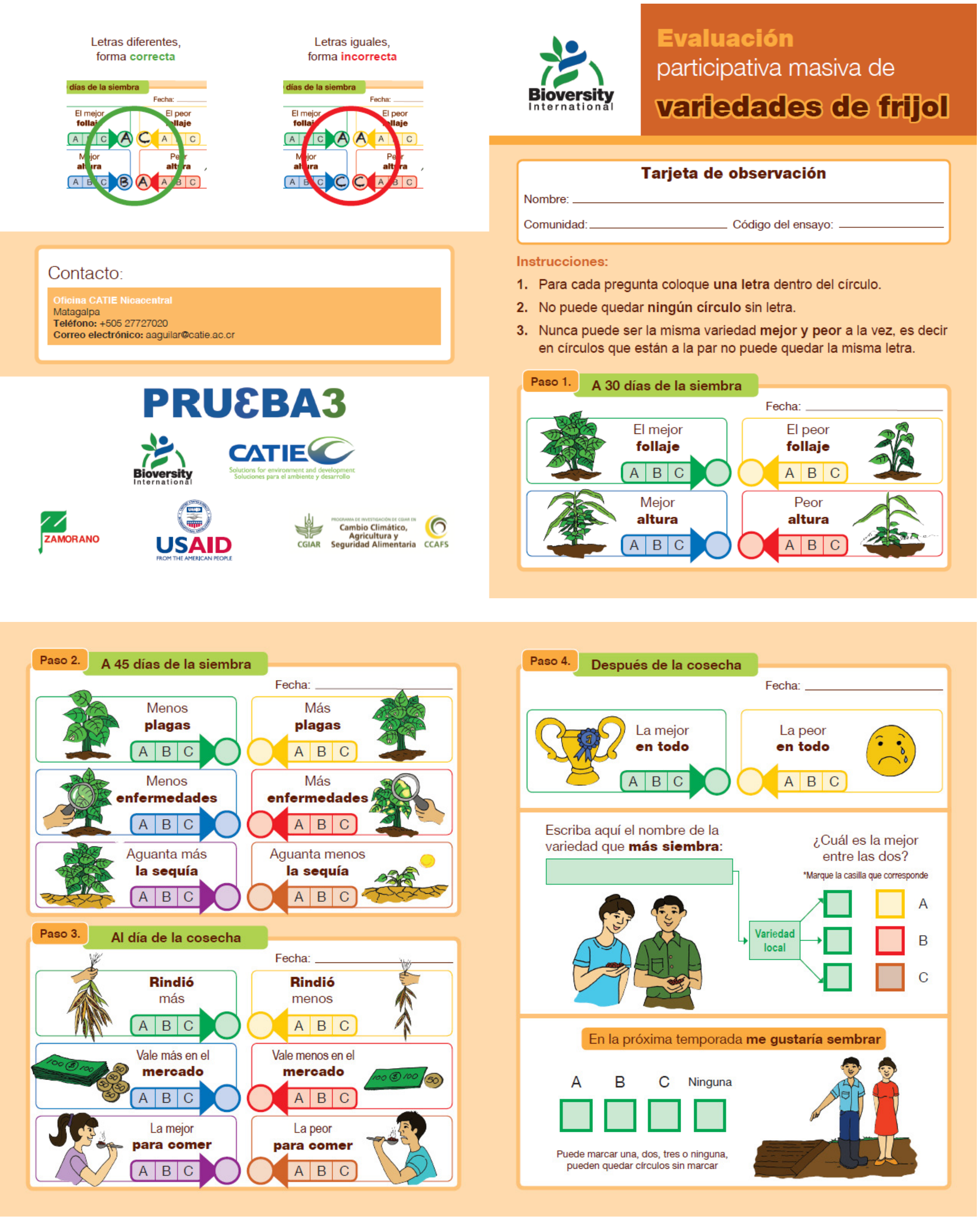


Fig. S4. Probability of outperforming (reliability) existing varietal recommendations by using crop varieties recommendations generated with the tricot citizen science approach. Panel *(A)* shows the probability of INTA Fuerte Sequia and INTA Centro Sur outperform varieties INTA Matagalpa and INTA Rojo in Nicaragua. Panel *(B)* shows the probability of varieties 208279 and 8034 outperform varieties Assassa, Hitosa and Ude in Ethiopia. Panel *(C)* shows the probability of varieties HD 2733 and K 9107 outperform HD 2967 in India.


Letras diferentes,
forma correcta
Letras iguales,
forma incorrecta
días de la siembra
Fecha:
El mejor follaje
El peor follaje
Mejor altura
Peor altura
A B C
PRUEBA3
Bioversity International
CATIE
Solutions for environment and development
Soluciones para el ambiente y desarrollo
ZAMORANO
USAID
FROM THE AMERICAN PEOPLE
CGIAR
PROGRAMA DE INVESTIGACIÓN DE CGIAR EN
Cambio Climático, Agricultura y Seguridad Alimentaria
CCAFS

Contacto:

Oficina CATIE Nicacentral
Matagalpa
Teléfono: +505 27727020
Correo electrónico: aaguilar@catie.ac.cr

Evaluación
participativa masiva de
variedades de frijol

Tarjeta de observación

Nombre: ____________________

Comunidad: ____________ Código del ensayo: ____________

Instrucciones:

1. Para cada pregunta coloque **una letra** dentro del círculo.
2. No puede quedar **ningún círculo** sin letra.
3. Nunca puede ser la misma variedad **mejor y peor** a la vez, es decir en círculos que están a la par no puede quedar la misma letra.

Paso 1. A 30 días de la siembra

Fecha: ____________

El mejor **follaje** A B C ○ | ○ A B C El peor **follaje**

Mejor **altura** A B C ○ | ○ A B C Peor **altura**

Paso 2. A 45 días de la siembra

Fecha: ____________

Menos **plagas** A B C ○ | ○ A B C Más **plagas**

Menos **enfermedades** A B C ○ | ○ A B C Más **enfermedades**

Aguanta más **la sequía** A B C ○ | ○ A B C Aguanta menos **la sequía**

Paso 3. Al día de la cosecha

Fecha: ____________

Rindió más A B C ○ | ○ A B C **Rindió** menos

Vale más en el **mercado** A B C ○ | ○ A B C Vale menos en el **mercado**

La mejor **para comer** A B C ○ | ○ A B C La peor **para comer**

Paso 4. Después de la cosecha

Fecha: ____________

La mejor **en todo** A B C ○ | ○ A B C La peor **en todo**

Escriba aquí el nombre de la variedad que **más siembra**: □

Variedad local

¿Cuál es la mejor entre las dos?
*Marque la casilla que corresponde

□ □ A
□ □ B
□ □ C

En la próxima temporada **me gustaría sembrar**

A □ B □ C □ Ninguna □

Puede marcar una, dos, tres o ninguna, pueden quedar círculos sin marcar

Fig. S5. Data collection card for crowdsourcing trials (example from Nicaragua). Participating farmer fills it with her or his observations about the varieties. The format given to farmers is full colour. The idea is that farmers write the selected letter in the circle. Questions were applied as ultra-short statements (e.g. ‘Highest yield’, ‘Lowest yield’). We also added illustrations for each of the characteristics to make the format easier to use for users with low levels of literacy (see van Etten et al (2016) for further details on the tests made to define this format).

Table S1. Number of tricot trials per cropping season used to model the performance of durum wheat (Ethiopia; Meher season), bread wheat (India; Rabi season), and common beans (Nicaragua; Primera, Apante and Postrera seasons).

Year	**Ethiopia**	**India**	**Nicaragua**		
			Primera	Apante	Postrera
2012	–	562	–	–	–
2013	176	4134	–	–	–
2014	578	4947	–	–	–
2015	336	834	–	481	177
2016	–	–	64	87	33

Table S2. Core questionnaire applied with farmers to access the performance of crop varieties. Some questions on this questionnaire may change between countries/crops (e.g., tiller capacity and lodging for wheat, and dry tolerance and color for common beans) and the objectives of the local organization involved in the trial. However, core questions are made (e.g., yield, overall performance) to maintain the uniformity of all datasets and ensure comparisons among trials.

Label	Description
farmer_name	The name of the farmer who received the seeds and will conduct the trial
husband-wife_name	The husband/wife name
village	The name of the village the farmer lives and will conduct the trial
district	The district (municipality, etc.) of the village
gender(M/F)	The gender of the farmer (male or female)
age	The farmer age (years)
planting_date	The date farmer started the trial (same day for all three varieties)
best_variety_characteristic_1_pest_resistance	Which variety performed better for pest resistance (A, B, or C)?
worst_variety_characteristic_1_pest_resistance	Which variety performed worse for pest resistance (A, B, or C)?
best_variety_characteristic_2_yield	Which variety performed better for yield (A, B, or C)?
worst_variety_characteristic_2_yield	Which variety performed worse for yield (A, B, or C)?
best_variety_characteristic_3_grow_again	Which variety you would like to grow again in the next season (A, B, or C or none)?
worst_variety_characteristic_3_grow_again	Which variety you would not like to grow again in the next season (A, B, or C or none)?
best_variety_characteristic_4_overall_performance	Which variety performed better among all characteristics (A, B or C)?
worst_variety_characteristic_4_overall_performance	Which variety performed worse among all characteristics (A, B or C)?
overall_vs_local_a	How was the performance of the variety A compared to the local variety (better, worse)?
overall_vs_local_b	How was the performance of the variety B compared to the local variety (better, worse)?
overall_vs_local_c	How was the performance of the variety C compared to the local variety (better, worse)?
best_variety_characteristic_x_define*	Which variety performed better in terms of [characteristic x] (A, B or C)?
worst_variety_characteristic_x_define*	Which variety performed worse in terms of [characteristic x] (A, B or C)?

*Question(s) to be defined according the crop characteristics and the objectives of the crowdsourcing trial as defined by the local organization running the project.

Supplementary Code S1

R code used to model the performance of crop varieties with Plackett-Luce model

```
# ------------------------------------------------------------------
# ------------------------------------------------------------------

library(here)
library(tidyverse)
library(svglite)
library(Matrix)
library(reshape2)
library(magrittr)
library(caret)
library(RCurl)
library(psychotools)
library(PlackettLuce)
library(partykit)
library(BradleyTerryScalable)
library(igraph)
library(foreach)
library(doParallel)
library(abind)
library(dismo)
library(rgeos)
library(gstat)
library(raster)
library(alphahull)

# For updates visit: https://github.com/kauedesousa/ClimMobTools

# ------------------------------------------------------------------
# ------------------------------------------------------------------
# Read data and load additional functions ####

#add ClimMob tools for data learning
tools <- RCurl::getURL("https://raw.githubusercontent.com/
kauedesousa/ClimMobTools/pilot/ClimMobLearning.R",
                       ssl.verifypeer = FALSE)
eval(parse(text = tools))

#Define number of cores for parallelisation
n_cpu <- 3

#Read data
filename <- "tricot_data.csv"
df <- read_csv( here(filename), na = c("NA",""))

#Crop that will be analysed
crop <-  sort(unique(df$crop))


#This script will evaluate the performance of Plackett-Luce models
# under following models:
# PLT-nocov without explanatory variables
# PLT-design  includes season, planting date, location, soil
# PLT-climate  with climatic explanatory variables
# PLT-clim+loc with explanatory variables from climate and location
approach <- c("PLT-nocov","PLT-design","PLT-climate","PLT-clim+loc")

#-----------------------------------------------------------------
#-----------------------------------------------------------------
# Run models ####
```

```
set.seed(123)

for(m in seq_along(crop)){

#--------------------------------------------------
#--------------------------------------------------
# Subset data and check rankings and explanatory variables ####

cat("####################################␣\n␣Starting␣analysis:",
    toupper(crop[m]), "␣\n␣Time:", date(), "\n")

#Read arguments for this crop
args <- read.table(here(paste0(crop[m],"_args.txt")),
                   header = TRUE, stringsAsFactors = FALSE)

overallvslocal <- as.logical(args[match("overallVSlocal", args[,1]), 2])
overallvsyield <- as.logical(args[match("overallVSyield", args[,1]), 2])
dropcovar <- args[match("dropCovar", args[,1]), 2]
dropitem <- args[match("dropItem", args[,1]), 2]
dropseason <- args[match("dropSeason", args[,1]), 2]
minplots <- as.integer(args[match("minPlots", args[,1]), 2])
averageseason <- as.logical(args[match("avgSeason", args[,1]), 2])
labels <- as.character(args[match("labels", args[,1]), 2])
if(labels=="NULL") {labels <- NULL} else {labels <- strsplit(labels, ",")[[1]]}
#Get arguments for Plackett-Luce model
minsize <- as.numeric(args[match("minsize", args[,1]), 2])
bonferroni <- as.logical(args[match("bonferroni", args[,1]), 2])
alpha <- as.numeric(args[match("alpha", args[,1]), 2])
npseudo <- as.numeric(args[match("npseudo", args[,1]), 2])
mean.method <- args[match("mean.method", args[,1]), 2]

#Subset data for the m crop
mydata <- df[df$crop == crop[m] , ]

#Create folder for outputs
output <- "output/"
dir.create(here(output), showWarnings = FALSE)

#An output folder for the crop
output <- paste0(output, unique(mydata$crop))
dir.create(here(output), showWarnings = FALSE)

#If there is any season to drop, this is to avoid Cholmod error
#https://github.com/hturner/PlackettLuce/issues/24
#mydata <- mydata[!mydata$season %in% dropseason, ]

#Reclassify factors levels to fit in the subset data
mydata[c("soil","season")] <- lapply(mydata[c("soil","season")],
                                     function(x) as.factor(as.character(x)))

#Get colnames with varieties info
vars <- grepl("variety", names(mydata))

#Remove items (varieties) tested in only 1 season
# here we remove items, not the entire row, rows with more
# than 1 NA will be removed next
rm_item <- cbind(vars = unlist(mydata[vars]),
                 season = rep(as.character(mydata$season), sum(vars)))
rm_item <- as.data.frame(table(rm_item[,1], rm_item[,2]))
rm_item$Freq[rm_item$Freq > 0] <- 1
rm_item <- aggregate(rm_item[,"Freq"], by=list(rm_item[,"Var1"]), sum)
rm_item <- as.vector(c(as.character(rm_item[rm_item$x<2, 1 ])))

#Remove selected items
mydata[vars] <- lapply(mydata[vars], function(X){
  ifelse(X %in% rm_item, NA, X)
})

#Remove entries (rows) with more than 1 NA per observer or duplicated items
keep <- apply(mydata[,vars], 1, function(X){
  k <- ifelse(sum(is.na(X)) > 1 | any(duplicated(X)), FALSE, TRUE)
})

cat("\n␣Removing:", sum(!keep), "of:", length(keep),
    "observations␣\n␣reason:␣varieties␣tested␣in␣only␣one␣season␣\n" )

#Remove wrong evaluations best == worst in overall_performance
keep <- ifelse(mydata$best==mydata$worst, FALSE, keep)

mydata <- mydata[keep,]

cat("\n␣Removing:", sum(!keep), "of:", length(keep),
    "observations␣\n␣reason:␣varieties␣tested␣in
␣␣␣␣only␣one␣season␣and␣inconsistent␣rankings␣\n" )

#Then look for varieties with low representativity
# defined by minplots
# this is to avoid Cholmod error
# https://github.com/hturner/PlackettLuce/issues/24
rm_item <- as.data.frame(sort(table(unlist(mydata[,vars]))))
rm_item <- as.character(rm_item$Var1[rm_item$Freq < minplots])
rm_item <- sort(unique(c(rm_item, dropitem)))

#Remove selected items
mydata[vars] <- lapply(mydata[vars], function(X){
  ifelse(X %in% rm_item, NA, X)
})

#Remove entries (rows) with more than 1 NA per observer or duplicated items
keep <- apply(mydata[,vars], 1, function(X){
  k <- ifelse(sum(is.na(X)) > 1 , FALSE, TRUE)
})

#Remove all data with issues described above
mydata <- mydata[keep, ]

#Get name of retained items
itemnames <- sort(unique(unlist(mydata[,vars])))

#Compute correlation between overall rankings and yield rankings
if(overallvsyield){
  #rankings from overall performance
  overall <- mydata[,c("variety_a","variety_b","variety_c","characteristic","best","worst")]
  #rankings from yield
  yield <- mydata[,c("variety_a","variety_b","variety_c","yield","best_yield","worst_yield")]
  #make sure that both datasets has the same names
  names(yield) <- names(overall)

  #remove NA's and inconsistent rankings in yield
  ykeep <- !is.na(yield$best) & !is.na(yield$worst) &  yield$best != yield$worst
  #keep those rows with no NA's
  yield <- yield[ykeep,]
  overall <- overall[ykeep,]

  #get rankings for yield
  yield <- grouped_rankings(as.PL(yield, local = FALSE, additional.rank = FALSE ),
  seq_len(nrow(yield)))
  yield <- yield[1:length(yield),, as.grouped_rankings = FALSE]

  #get rankings for overall performance
  overall <- grouped_rankings(as.PL(overall, local = FALSE, additional.rank = FALSE ),
  seq_len(nrow(overall)))
  overall <- overall[1:length(overall),, as.grouped_rankings = FALSE]

  #Calculate Kendall's correlation
  #export this output
  capture.output(kendallTau_plt(yield, overall)[1] * -1 ,
                 file = here(output, "Kendall_tau_yield_vs_overall.txt"))
}


#Take explanatory variables in a separate dataframe
covar <- cbind(mydata [, c("season","lat","lon","xy","yx","planting_day")],
               mydata [, c(23:(ncol(mydata)-4))])

#Check if there is any explanatory variable to drop
varout <- grepl(dropcovar, names(covar))
covar <- covar[,!varout]

#Check variance in explanatory variables
varout <- caret::nearZeroVar(covar)
cat("\n␣Removing␣these␣variables␣with␣near␣zero␣variance:␣\n",
    sort(names(covar)[varout]),"\n")
#Remove variables with near zero variance
covar <- covar[,-varout]

#Take rankings from mydata
mydata <- mydata[, c("variety_a","variety_b","variety_c",
                     "characteristic","best","worst")]
#Take the number of rows in this dataset
n <- nrow(mydata)

#Generate a Plackett-Luce grouped rankings
if(overallvslocal){
  R <- as.PL(mydata, local = TRUE, additional.rank = TRUE)
  G <- grouped_rankings(R, rep(seq_len(n), 4) )
}
if(!overallvslocal){
  R <- as.PL(mydata, local = FALSE)
  G <- grouped_rankings(R, seq_len(n))
}


cat("This␣analysis␣will␣use",n,"observations.\n")

#Merge grouped rankings with explanatory variables
mydata <- cbind(G, covar)

#----------------------------------------------------
#----------------------------------------------------
# Set parameters for forward selection ####

#Define folds based on the season where this crop was evaluated
```

```
folds <- as.integer(as.factor(as.character(mydata$season)))
#number of folds
k <- max(folds)


#Perform a forward variable selection on explanatory variables and select
## those who better contribute to improve predictions between seasons
## also compare the performance of explanatory variables with a null model
## to be able to predict a PLtree object from a null model we create a NULL
## variable (variable with no variance) and added to the main dataset
mydata$P1 <- 1

#Select explanatory variables
expvar <- names(mydata)[8:ncol(mydata)]

#Calculate the weights of each season based on the square root of
# n observations per season divided by total n
wseason <- as.vector(summary(mydata$season))
wseason <- sqrt(wseason/n)
wseason <- wseason/sum(wseason)

#Define initial parameters for forward selection
## baseline deviance
par_n <-  0
## vector to keep best explanatory variables
var_keep <- NULL
## if TRUE the routine will keep running, this parameters is updated
# at the end of each "while" routine
best <- TRUE
## number of runs
runs <- FALSE
## vector with best parameters (loglik, vars and lambda) in each run
best_parameters <- NULL
## minimum size for node split
minsize <- round((n*minsize), -1)
#Remove unused objects and reduce size of globalenv() in parallel export
rm(covar, yield, ykeep, overall)

#---------------------------------------------------
#---------------------------------------------------
#---------------------------------------------------
#---------------------------------------------------
# Run forward selection ####

# Create cluster to do parallelisation
cluster <- parallel::makeCluster(n_cpu)
doParallel::registerDoParallel(cluster)

# keep running until the model reach its best performance
while(best){

  cat("\n␣Starting␣Forward␣Selection.␣Run␣", sum(runs)+1, "\n␣Time:␣", date(), "\n")

  fs <- length(expvar)

  #get predictions from nodes and put in matrix (foreach)
  models <- foreach(i = 1:fs,
                    .combine = acomb,
                    .packages = c("PlackettLuce","psychotree"),
                    .export = ls(globalenv())) %dopar% (
                      f1(formula = as.formula(paste0("G␣~␣", paste(unique(c(var_keep,
                     expvar[i])), collapse = "␣+␣"))),
                        d = mydata,
                        k = k, folds = folds, minsize = minsize, npseudo = npseudo,
                        mean.method = mean.method,
                        alpha = alpha, bonferroni = bonferroni)
                   )

#calculate Akaike weights along explanatory variables per season
AW <- as.matrix(models[,2:(k+1)])
AW <- apply(AW, 2, function(x) round(as.numeric(x), digits = 10))
AW <- apply(AW, 2, function(X) AkaikeWeights(X)[[3]] )

#then take the weighted average per season Stouffer's method with weights
for(i in seq_len(k)) AW[,i] <- AW[,i] * wseason[i]

#take the Stouffer's weighted mean
meanAW <- rowSums(AW)

#take estimators
estimators <- models[,(k+2):ncol(models)]
dimnames(estimators)[[2]] <- c("AIC","McFadden","r2ML","r2CU","kendallTau")

#take the model call from each model
call <- models[,1]

#take maximum parameter from Akaike weights
par_max <- max(meanAW)

#take the position of par_max in expvar vector
index_par_max <- which.max(meanAW)

#Is par_max best (higher) than par_n?
best <- par_max > par_n #if not, the forward selection will stop

#if best, save the outputs
if(best){

  #take the name of best variable
  best_var <- expvar[index_par_max]
  cat("###␣Best␣covariate␣identified:", best_var, "\n")

  #sum runs
  runs <- c(runs, best)

  #take outputs from this run and export to a .csv file
  out <- as_tibble(cbind(call, models[,c(2:(k+1))], meanAW = meanAW, estimators))
  names(out)[2:(k+1)] <- paste0(rep("Deviance",k), 1:k)

  #remove best_var from this run
  expvar <- expvar[!grepl(best_var, expvar)]

  #remove null var from the first run, no longer necessary
  expvar <- expvar[!grepl("P1", expvar)]

  #keep this model for the next run
  var_keep <- c(var_keep, best_var)

  #add best model to the next round
  expvar <- c(expvar, paste(var_keep, collapse = '␣+␣'))

  #change the base for par_n (minimun accepted value)
```

```
    par_n <- par_max

    #keep the best parameters form this run
    best_parameters <- rbind(best_parameters, cbind(par_max, model = toString(var_keep)))

    write.csv(out, here(output, "model_parameters.csv" )  )
    cat("␣##########␣End␣run", sum(runs), "\n" )
  }

}

#Stop cluster connection
stopCluster(cluster)

#take the best model
best_model <- var_keep

#Save parameters used in this analysis as a .txt file
write.table(rbind(n = n, minsize = minsize, bonferroni = bonferroni,
                  alpha = alpha, npseudo = npseudo,
                  mean.method = mean.method,
                  covar = toString(best_model)),
            file = here(output, "PLT_parameters.txt" ))

cat("End␣Forward␣Selection.␣\n␣Time:␣", date(),
    "\n␣Best␣model␣will␣use:",  best_model ,"\n")

# Define list of explanatory variables to use in each approach
attrib <- list(zero = c("P1"),
               desi = c("lon","lat","planting_day"),
               clim = c(best_model),
               clsp = c(best_model, c("lon","lat","yx") ))


#--------------------------------------------------
#--------------------------------------------------
#--------------------------------------------------
#--------------------------------------------------
# Run blocked cross-validation for the different model approaches ####

# Run the best model against the PLT-null, PLT-clim+loc and PLT-design
cat("Starting␣blocked␣cross-validation", date() ,"\n")

blockedfolds <-  matrix(NA, nrow = length(approach), ncol = (8+(k*6)),
                        dimnames = list(seq_along(approach),
                                        c("Approach","Model","AIC","Deviance",
                                          "McFadden","r2ML","r2CU","kendallTau",
                                          paste0(rep("AIC",times = k), 1:k),
                                          paste0(rep("Deviance",times = k), 1:k),
                                          paste0(rep("McFadden",times = k), 1:k),
                                          paste0(rep("r2ML",times = k), 1:k),
                                          paste0(rep("r2CU",times = k), 1:k),
                                          paste0(rep("kendallTau",times = k), 1:k))))

blockedfolds[,1] <- approach


# Run blocked cross-validation over approaches
for(a in 1:length(approach)){

  cat("####␣blocked␣cross-validation␣in", approach[a], "\n")
  formula_a <- as.formula(paste0("G␣~␣", paste(attrib[[a]], collapse = "␣+␣") ))

  model <- crossvalidation_PLT(formula_a,
                               d = mydata, k = k, folds = folds, minsize = minsize,
                               alpha = alpha, bonferroni = bonferroni, npseudo = npseudo,
                               verbose = FALSE, mean.method = mean.method)

  estimators <- c(model$AIC, model$Deviance, model$McFadden, model$r2ML, model$r2CU,
  model$kendallTau, as.vector(model$estimators[,-4]))

  blockedfolds[a,3:ncol(blockedfolds)] <- estimators

  blockedfolds[a,2] <- paste0("G␣~␣", paste(attrib[[a]], collapse = "␣+␣") )

}

#-------------------------------------------------
#-------------------------------------------------
#-------------------------------------------------
#-------------------------------------------------
# Run average season using historical data ####
if(averageseason){

  cat("Starting␣average␣season␣with␣blocked␣cross-validation", date() ,"\n")

  #load climatology data
  load(here("processing", paste0(crop[m],"_climatology.RData")))
  #define number of predictions to be generated n.years * n.estimated planting dates
  npreds <- length(climatology) * length(climatology[[1]])

  #list into array
  nr <- nrow(climatology[[1]][[1]])
  nc <- ncol(climatology[[1]][[1]])

  climatology <- unlist(climatology)
  climatology <- array(climatology, dim = c(nr, nc, npreds),
                       dimnames = list(1:nr, names(climatology[[1]][[1]]), 1:npreds ))

  #matrix to keep results
  avgseason <-  matrix(NA, nrow = npreds, ncol = (8+(k*6)),
                       dimnames = list(seq_len(npreds),
                                       c("Approach","Model","AIC","Deviance",
                                         "McFadden","r2ML","r2CU","kendallTau",
                                         paste0(rep("AIC",times = k), 1:k),
                                         paste0(rep("Deviance",times = k), 1:k),
                                         paste0(rep("McFadden",times = k), 1:k),
                                         paste0(rep("r2ML",times = k), 1:k),
                                         paste0(rep("r2CU",times = k), 1:k),
                                         paste0(rep("kendallTau",times = k), 1:k))))


  pb <- txtProgressBar(min = 1, max = npreds, style = 3)

  for(a in seq_len(npreds)){

    #temporary dataframe with G and explanatory variables
    a_df <- cbind(G, as.data.frame(climatology[keep, , a]))

    model <- crossvalidation_PLT(as.formula(paste0("G␣~", paste(attrib$clim, collapse = "+"))),
                                 d = a_df, k = k, folds = folds, minsize = minsize,
                                       alpha = alpha, bonferroni = bonferroni, npseudo = npseudo,
                                       verbose = FALSE, mean.method = mean.method)

    estimators <- c(model$AIC, model$Deviance, model$McFadden, model$r2ML,
                    model$r2CU, model$kendallTau, as.vector(model$estimators[,-4]))

    avgseason[a,3:ncol(avgseason)] <- estimators

    setTxtProgressBar(pb, a)

  }

  close(pb)

  cat("End␣of␣average␣season␣\n")

  #export data
  write.csv(avgseason, here(output, "average_season.csv"), row.names = FALSE)

  avgseason <- t(as.matrix(colMeans(avgseason), byrow = TRUE))

  avgseason[1:2] <-c("avgseason", "G␣~␣averageseason")

  blockedfolds <- as.matrix(rbind(blockedfolds, avgseason))
}

#write matrix as a .csv file
write.csv(blockedfolds, here(output, "performance_PLT_blocked_crossvalidation.csv"),
          row.names = FALSE)


#------------------------------------------------------------
#------------------------------------------------------------
#------------------------------------------------------------
#------------------------------------------------------------
# Run k-fold cross-validation for the different model approaches ####
cat("Starting␣k-fold␣cross-validation",date(),"\n")
#Define number of folds
k <- 10
#Define samples
folds <- sample(rep(1:k, times = ceiling(n/k), length.out=n), replace=FALSE)

#matrix to keep the outputs
kfolds <-   matrix(NA, nrow = length(approach), ncol = (8+(k*6)),
                   dimnames = list(seq_along(approach),
                                   c("Approach","Model","AIC","Deviance",
                                     "McFadden","r2ML","r2CU","kendallTau",
                                     paste0(rep("AIC",times = k), 1:k),
                                     paste0(rep("Deviance",times = k), 1:k),
                                     paste0(rep("McFadden",times = k), 1:k),
                                     paste0(rep("r2ML",times = k), 1:k),
                                     paste0(rep("r2CU",times = k), 1:k),
                                     paste0(rep("kendallTau",times = k), 1:k))))

kfolds[,1] <- approach

#run k-fold over approaches
for(a in 1:length(approach)){
  cat("####␣k-fold␣in", approach[a], "\n")

  formula_a <- as.formula(paste0("G␣~␣", paste(attrib[[a]], collapse = "␣+␣") ))
  model <- crossvalidation_PLT(formula_a,
                               d = mydata, k = k, folds = folds, minsize = minsize,
                               alpha = alpha, bonferroni = bonferroni, npseudo = npseudo,
                               verbose = FALSE)

  estimators <- c(model$AIC, model$Deviance, model$McFadden, model$r2ML, model$r2CU,
  model$kendallTau, as.vector(model$estimators[,-4]))

  kfolds[a,3:ncol(kfolds)] <- estimators

  kfolds[a,2] <- paste0("G␣~␣", paste(attrib[[a]], collapse = "␣+␣") )
}

#write matrix as a csv file
write.csv(kfolds, here(output, "performance_PLT_10fold_crossvalidation.csv"),
          row.names = FALSE)


#------------------------------------------------
#------------------------------------------------
#------------------------------------------------
#------------------------------------------------
# Fit the best model from k-fold and generate Plackett-Luce plots ####

#Fit pltree with best model
cat("Fit␣pltree␣with␣best␣model␣\n")
tree <- pltree(as.formula(paste0(c("G␣~␣"), paste(attrib$clim, collapse = "␣+␣"))),
               data = mydata, alpha = alpha,
               minsize = minsize, npseudo = npseudo)

#export summary of fitted pltree
capture.output(tree, summary(tree), file = here(output, "pltree_clim.txt") )

#Write this tree as a .svg file
svg(filename = here(output, "PLTree.svg"),
    width=15,
    height=15,
    pointsize=12)
partykit::plot.modelparty(tree)
dev.off()

#Generate plots with error bars using qvcal
plots <- plot_nodes(tree, labels = labels, font.size = c(20, 23))
#define file names
nodepaths <- paste0(here(output, paste0("PLTree_", names(plots) ,".svg")))
#put names in a list
nodepaths <- as.list(nodepaths)
#define names of each element in this list
names(nodepaths) <- names(plots)

#write plots as .svg file
h <- length(itemnames) + 1.5
mapply(function(X, Y){

  ggsave(filename = Y, plot = X,
         dpi = 600, width = 15, height = h, units = "cm")

}, X = plots, Y = nodepaths )

#------------------------------------------------
```

```
#----------------------------------------
#----------------------------------------
#----------------------------------------
#Make a histogram to show the distribution of ####
## explanatory variables among blocks (seasons)

#get names of vars within nodes
vars <- partykit:::.list.rules.party(tree)
vars <- strsplit(vars, "&|<=|[>]")
mvalue <- which.max(as.vector(unlist(lapply(vars, length))))
vars <- vars[[mvalue]]
vars <- vars[seq(1, length(vars)-1, 2)]
vars <- gsub("␣", "", vars)

#get split values
ni <- nodeids(tree)
ni_terminal <- nodeids(tree, terminal = TRUE)
ni_inner <- ni[!ni %in% ni_terminal]
breaks <- sapply(ni_inner, function(X){
  split_node(node_party(tree[[X]]))$breaks} )

for(i in unique(vars)){

  b <- breaks[ vars%in% vars[i] ]

  h <- ggplot(mydata, aes(mydata[, vars[i] ], fill = season)) +
    geom_histogram(bins = 50) +
    geom_vline(xintercept = b, col = "red") +
    labs(x = "", y = "Count") +
    scale_fill_brewer(palette = "GnBu", direction = -1, name = "") +
    scale_x_continuous(expand = expand_scale(mult = c(0, .05))) +
    scale_y_continuous(expand = expand_scale(mult = c(0, .01))) +
    theme_classic() +
    theme(axis.text = element_text(size=20, colour="black"),
          axis.text.x = element_text(size=20, angle = 0,
                                     hjust=0.5,vjust=1, face="plain"),
          axis.text.y = element_text(size=20, angle = 0,
                                     hjust=1, vjust=0.5, face="plain"),
          axis.title=element_text(size=20,face="bold"),
          axis.line = element_line(),
          legend.position=c(.15, .8),
          legend.text=element_text(size=20),
          legend.background = element_rect(colour = NA),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          plot.background = element_blank())

  ggsave(here(output, paste0("histogram_",vars[i],".svg")),
         plot = h, dpi = 600,
         width = 25, height = 20, units = "cm")

}


#----------------------------------------
#----------------------------------------
#----------------------------------------
#----------------------------------------
# Calculate worst regret ####
#Regret is the difference with the best variety in each node
WR <- worstRegret(tree)
```

```
if(!is.null(labels)) {WR$items <- labels}

write.csv(WR, here(output, "worst_regret.csv"))


#--------------------------------------------------------------
#--------------------------------------------------------------
#--------------------------------------------------------------
#--------------------------------------------------------------
# Plot network connections among items ####

adj <- PlackettLuce::adjacency(R)

adj <- as.vector(adj)

adj <- t(matrix(adj, nrow = length(labels), ncol = length(labels)))

dimnames(adj) <- list(labels, labels)

adj <- btdata(adj, return_graph = TRUE)

svg(filename = here(output, "connections.svg"),
    width=30,
    height=30,
    pointsize=55)
plot.igraph(adj$graph, vertex.size = 30, edge.arrow.size = 0.1)
dev.off()
```