Package 'DyMEP'

January 20, 2025

Type Package

Title Dynamic Multi Environment Phenology-Model

Version 0.1.2

Author Flavian Tschurr

Maintainer Flavian Tschurr <flavian.tschurr@usys.ethz.ch>

Description Mechanistically models/predicts the phenology (macrophases) of 10 crop plants (trained on a big dataset over 80 years derived from the German weather service (DWD) <https://opendata.dwd.de/>). Can be applied for remote sensing purposes, dynamically check the best subset of available covariates for the given dataset and crop.

License LGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Imports stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2024-05-02 11:53:05 UTC

Contents

`asymptotic_prediction` *DRC function: asymptotic_prediction*

Description

DRC function: asymptotic_prediction

Usage

```
asymptotic_prediction(x, params)
```

Arguments

`x`	input variable
`params`	list of input parameter; Asym a numeric parameter representing the horizontal asymptote on the right side (very large values of input). ; lrc a numeric parameter representing the natural logarithm of the rate constant; c0 a numeric parameter representing the x for which the response is zero.

Value

value with asymptotic response

Examples

```
asymptotic_prediction(5,list("Asym_value"=0.5,
                    "lrc_value"=0.2,
                      "c0_value"=4))
# visualization
asymptote <- lapply(seq(0, 10, 0.1),
asymptotic_prediction,
list("Asym_value"=0.5, "lrc_value"=0.2, "c0_value"=4))
plot(seq(0, 10, 0.1), asymptote)
```

available_crops_and_phases

available_crops_and_phases

Description

check what crops and corresponding phenology phases are available at the moment. Chose a crop (crop_abbrev), phenological phase from the output of this function further usage

Usage

```
available_crops_and_phases()
```

Value

dataframe with available crops and phenology phases

Examples

```
available_crops_and_phases()
```

available_environmental_covariates

available_environmental_covariates check what environmental covariates are implemented, use or alter prediction these abbreviations and the corresponding unit

Description

available_environmental_covariates check what environmental covariates are implemented, use or alter prediction these abbreviations and the corresponding unit

Usage

```
available_environmental_covariates()
```

Value

dataframe with available environmental covariates

Examples

```
available_environmental_covariates()
```

DyMEP *Dynamic Multi Environment Phenology-Model*

Description

Empirically models/predicts the phenology (macro-phases) of 10 crop plants (trained on a big dataset over 80 years derived from the German weather service (DWD)). Can be applied for remote sensing purposes, environmental inputs can be chosen from a range of pre-trained response curves and applied to the trained crops and phenological phases. No retraining is done within the use of this package.

Examples

```
available <- available_crops_and_phases()
#what is the best environmental covariates for one or multiple phases?
# check what covairates are implemented in the model
available_covariates <- available_environmental_covariates()

best_DyMEP_model(env_covariates = c("tas","tasmin","VPD","SPI",
"global_radiation","tasmax","RH"),
                pheno_phases = c("sowing-emergence","jointing-heading"),
                crop_abbrev = "WW")
# create a list of wanted phases and corresponding environmental covariates
phase_covariate_list <- list("sowing-emergence" = c("tasmin","VPD","SPI"),
                        "emergence-jointing"= c("tas","tasmin","VPD","SPI"),
                     "jointing-heading" = c("global_radiation","tas","SPI"))

# alternatively you can create this input list directly like this with the
# best available model:
phase_covariate_list <- best_DyMEP_model(env_covariates =
c("tas","tasmin","VPD","SPI","global_radiation","tasmax","RH"),
pheno_phases = c("sowing-emergence","emergence-jointing","jointing-heading"),
crop_abbrev = "WW",
output_list_for_prediction = TRUE)


# create dummy environmental data
environmental_data <- data.frame("DATE" = seq.Date(
              from = as.Date("2021-01-01"), to = as.Date("2023-12-31"),by=1),
                           "tas"=runif(1095,min=-10,max=40),
                            "RH"=runif(1095,min=0,max=100),
                            "tasmin"=runif(1095,min=-10,max=40),
                            "tasmax"=runif(1095,min=-5,max=40),
                            "VPD" = runif(1095,min=0,max=40),
                            "SPI"= runif(1095,min=-1,max=4),
                            "global_radiation"= runif(1095,min=0,max=3500))


pheno_phase_prediction(phase_covariate_list = phase_covariate_list,
                       environmental_data = environmental_data,
```

```
                    phase_starting_date =as.Date("2021-01-01"),
                    crop_abbrev = "WW")

# you can also get a more detailed output, containing detailed predictions
# and the parameters of the used DRC curves:
detailed_output <- pheno_phase_prediction(
           phase_covariate_list = phase_covariate_list,
           environmental_data = environmental_data,
           phase_starting_date =as.Date("2021-01-01"),
           crop_abbrev = "WW",
           output_type = "detailed_information")

#  this output can be visualised like:
# get overview plot of the prediction
DyMEP_prediction_visualizer(detailed_output)
# check the DRC curves of the used model
DyMEP_DRC_visualizer(detailed_output)
```

`DyMEP_DRC_visualizer` *DyMEP_DRC_visualizer*

Description

Visualizes the Dose-Response Curves (DRC) for each phenological phase and environmental covariate.

Usage

```
DyMEP_DRC_visualizer(detailed_output)
```

Arguments

`detailed_output`
: Output of the pheno_phase_prediction function with output_type = "detailed_information".

Value

Returns plots showing the DRC curves for each phenological phase and environmental covariate. Each row represents a phenology phase.

Examples

```
phase_covariate_list <- best_DyMEP_model(env_covariates =
c("tas","tasmin","VPD","SPI","global_radiation","tasmax","RH"),
pheno_phases = c("sowing-emergence","emergence-jointing","jointing-heading"),
crop_abbrev = "WW",
output_list_for_prediction = TRUE)
# create dummy environmental data
environmental_data<- data.frame("DATE"=seq.Date(from = as.Date("2021-01-01"),
```

```
                    to = as.Date("2023-12-31"),by=1),
                                "tas"=runif(1095,min=-10,max=40),
                                "RH"=runif(1095,min=0,max=100),
                                "tasmin"=runif(1095,min=-10,max=40),
                                "tasmax"=runif(1095,min=0,max=40),
                                "VPD" = runif(1095,min=0,max=40),
                                "SPI"= runif(1095,min=-1,max=4),
                              "global_radiation"= runif(1095,min=0,max=3500))

DyMEP_DRC_visualizer(detailed_output = pheno_phase_prediction(
phase_covariate_list = phase_covariate_list,
environmental_data = environmental_data,
phase_starting_date =as.Date("2021-01-01"),
crop_abbrev = "WW",
output_type = "detailed_information")
)
```

`DyMEP_prediction_visualizer`

DyMEP_prediction_visualizer

Description

Visualizes the predictions of the DyMEP model.

Usage

```
DyMEP_prediction_visualizer(detailed_output)
```

Arguments

`detailed_output`
: Output of the pheno_phase_prediction function with output_type = "detailed_information".

Value

A plot with one panel per phenology phase, showing the environmental covariate responses, the GLM prediction, and the phase prediction (points).

Fields

`timestamp` Description of timestamp column.

Examples

```
phase_covariate_list <- best_DyMEP_model(env_covariates =
             c("tas","tasmin","VPD","SPI","global_radiation","tasmax","RH"),
           pheno_phases = c("sowing-emergence","emergence-jointing",
           "jointing-heading"),
```

```
                                        crop_abbrev = "WW",
                                        output_list_for_prediction = TRUE)

# Create dummy environmental data
environmental_data <- data.frame("DATE" =
                                   seq.Date(from = as.Date("2021-01-01"),
                                     to = as.Date("2023-12-31"), by = 1),
                       "tas" = runif(1095, min = -10, max = 40),
                        "RH" = runif(1095, min = 0, max = 100),
                        "tasmin" = runif(1095, min = -10, max = 40),
                        "tasmax" = runif(1095, min = 0, max = 40),
                        "VPD" = runif(1095, min = 0, max = 40),
                        "SPI" = runif(1095, min = -1, max = 4),
                        "global_radiation" = runif(1095, min = 0, max = 3500))

DyMEP_prediction_visualizer(detailed_output = pheno_phase_prediction(
                                phase_covariate_list = phase_covariate_list,
                                environmental_data = environmental_data,
                                phase_starting_date = as.Date("2021-01-01"),
                                crop_abbrev = "WW",
                                output_type = "detailed_information"))
```

`envpredutils.pheno_phase_prediction_glm_model`
apply the prediction with glm model

Description

apply the prediction with glm model

Usage

```
envpredutils.pheno_phase_prediction_glm_model(
  env_data_pheno_phase,
  pheno_phase,
  crop_abbrev,
  model,
  output_type = "dates"
)
```

Arguments

`env_data_pheno_phase`	environmental data required to predict the phase
`pheno_phase`	phenological phase
`crop_abbrev`	abbreviation of the crop
`model`	the selected model to predict the wanted phenological phase

output_type — either "dates" or "detailed_information"; defines what output of the model they user wants to have as return, default is set to "dates". If a user wants to get the response parameters, curves, predictions and model thresholds, it should be chosen "detailed_information" output = "dates" will return a dataframe with the stages and according dates output = "detailed_information" will return a list with the dates, but also the corresponding dose response parameters and predictions

Value

final output, either detailed (if output_type = "detailed_information") as list, or data.frame with dates if output_type = "dates"

---

non_linear_prediction *DRC function: non_linear_prediction*

Description

broken stick model according to an env variable

Usage

```
non_linear_prediction(env_variate, params)
```

Arguments

env_variate	value of a environmental covariate
params	list of input parameter; base_value: minimal value; slope estimated value, slope of the linear phase

Value

value with non_linear response

Examples

```
non_linear_prediction(1,list("base_value"=5,"slope_value"=1))

# visualization
non_linear <- lapply(seq(0, 10, 0.1),
non_linear_prediction,
list("base_value"=5,"slope_value"=1))
plot(seq(0, 10, 0.1), non_linear)
```

pheno_phase_prediction *function to predict the a phenological phase in winter wheat*

Description

predict one or all phenological phases

Usage

```
pheno_phase_prediction(
  phase_covariate_list,
  environmental_data,
  phase_starting_date,
  crop_abbrev,
  output_type = "dates",
  external_params_path = NULL
)
```

Arguments

phase_covariate_list	list like: list("sowing-emergence" = c("tas","VPD","SPI"), "emergence-jointing" = c("tas")) indicating per phenological phase the covariates to use. List of of phenological phases must be consecutive!
environmental_data	data.frame with the necessary environmental data, one column must be "DATE" (as.Date format), the others with the names of the environmental covariates (e.g. tas, tasmin etc.)
phase_starting_date	starting date of the first phase which will be predicted (object of class "Date" (use as.Date()))
crop_abbrev	abbreviation of the crop to be modeled (valid crop_abbrevs can be found with available_crops_and_phases())
output_type	either "dates" or "detailed_information"; defines what output of the model they user wants to have as return, default is set to "dates". If a user wants to get the response parameters, curves, predictions and model thresholds, it should be chosen "detailed_information" output = "dates" will return a dataframe with the stages and according dates output = "detailed_information" will return a list with the dates, but also the corresponding dose response parameters and predictions
external_params_path	path where additional crop parameters should be stored if not possible to download in to the regular R repository. The default is NULL, which will use the regular R repository as path

Value

returns the end-date of each phase

either return an object of class 'DyMEP', if detailed_information is selected as output_type, from a dataframe containing phenology data, or

Examples

```
pheno_phase_prediction(phase_covariate_list = list(
        "sowing-emergence" = c("tasmin","VPD","SPI","tasmax","tas","RH",
        "global_radiation"),
        "emergence-jointing"= c("tasmin","VPD","SPI","tasmax","tas","RH",
        "global_radiation"),
        "jointing-heading" = c("tasmin","VPD","SPI","tasmax","tas","RH",
        "global_radiation")),
    environmental_data <- data.frame("DATE" = seq.Date(
              from = as.Date("2021-01-01"), to = as.Date("2023-12-31"),by=1),
                          "tas"=runif(1095,min=-10,max=40),
                           "RH"=runif(1095,min=0,max=100),
                           "tasmin"=runif(1095,min=-10,max=40),
                           "tasmax"=runif(1095,min=-5,max=40),
                           "VPD" = runif(1095,min=0,max=40),
                           "SPI"= runif(1095,min=-1,max=4),
                           "global_radiation"= runif(1095,min=0,max=3500)),
      phase_starting_date =as.Date("2021-01-01"),
      crop_abbrev = "WW")
```

---

reg_linear_prediction *DRC function: reg_linear_prediction*

Description

linear model according to an env variable

Usage

```
reg_linear_prediction(env_variate, params)
```

Arguments

env_variate	value of a environmental covariate
params	list of input parameter; intercept estimated value, slope of the linear phase

Value

value with reg_linear response

Examples

```
reg_linear_prediction(1,list("intercept_value"=1, "slope_value"=5))
# visualization
reg_linear <- lapply(seq(0, 10, 0.1),
reg_linear_prediction,
list("intercept_value"=-1,"slope_value"=1))
plot(seq(0, 10, 0.1), reg_linear)
```

`WangEngels_prediction` *DRC function: WangEngels_prediction*

Description

DRC function: WangEngels_prediction

Usage

```
WangEngels_prediction(x, params)
```

Arguments

`x`	effective env_variable value
`params`	list of input parameter; xmin_value represents the minimal env_variable value above which growth response will happen ; xopt_value: optimal growth point, env_variable values here have the highest response; xmax_value represents the maximal env_variable value above which no growth response will happen according to the wang engel model.

Value

value with WangEngels response

Examples

```
WangEngels_prediction(10, params = list("xmin_value"=1,
                                        "xopt_value"=25,
                                        "xmax_value"=35,
                                         "r_value"=0.5))
 # visualization
WangEngels <- lapply(seq(0, 40, 0.1),
WangEngels_prediction,
list("xmin_value"=1,
     "xopt_value"=25,
      "xmax_value"=35,
       "r_value"=0.5))
       plot(seq(0, 40, 0.1), WangEngels)
```

Index