Package 'expDB'

May 8, 2026

Type Package

Title Database for Experiment Dataset

Version 0.3.0

Maintainer Bangyou Zheng <bangyou.zheng@csiro.au>

Description A 'SQLite' database is designed to store all information of experiment-based data including metadata, experiment design, managements, phenotypic values and climate records. The dataset can be imported from an 'Excel' file.

License MIT + file LICENSE

URL https://expdb.bangyou.me/, https://github.com/byzheng/expdb

BugReports https://github.com/byzheng/expdb/issues

Encoding UTF-8

Imports DBI, RSQLite, utils, readxl, png, grid, tibble, lubridate, dplyr, methods, weaana (>= 0.1.1), tidyr, magrittr, stats, reshape2, stringr, rlang

RoxygenNote 7.3.2

Suggests testthat

NeedsCompilation no

Author Bangyou Zheng [aut, cre]

Repository CRAN

Date/Publication 2025-09-23 12:00:02 UTC

Contents

`approx_zadoks` *Approximate Date for a Target Zadoks Stage*

Description

This function estimates the date corresponding to a specified target Zadoks stage by interpolating between known Zadoks stages and their associated dates. Phenology Analysis Functions

Usage

```
approx_zadoks(zadoks, date, target)
```

Arguments

`zadoks`	A numeric vector of Zadoks growth stages (should be between 51 and 69).
`date`	A vector of dates (class `Date`) corresponding to each Zadoks stage.
`target`	A single numeric value indicating the target Zadoks stage for which to estimate the date.

Details

This module provides methods for analyzing phenological data, particularly focusing on the Zadoks scale.

The following two methods are implemented:

1. **Relative Date for Observed Target (Zadoks Rounded):** - If the target phenological stage is observed in the dataset (after rounding Zadoks values), the function will use the relative date corresponding to this observation. This allows for precise tracking of phenological events based on actual recorded data.

2. **Linear Interpolation for Target within Observed Range:** - If the target Zadoks stage is not directly observed but falls within the range of observed stages, the function estimates the date using linear interpolation between the nearest observed stages. This provides an approximate date for the target stage based on the trend of observed data.

Value

A `Date` object representing the interpolated date for the target Zadoks stage, or `NA` if the target is outside the observed range.

Note

Ensure that the input data includes Zadoks stage observations and corresponding dates for accurate analysis.

Examples

```
zadoks <- c(51, 54, 55, 59)
date <- as.Date(c("2020-08-15", "2020-08-17", "2020-08-18", "2020-08-21"))
target <- 55
approx_zadoks(zadoks, date, target)
approx_zadoks(zadoks[-3], date[-3], target)
```

---

dbAddDesigns *Add design for a trial*

Description

Add design for a trial

Usage

```
dbAddDesigns(con, data, extra_design = NULL)
```

Arguments

con	a connection object as produced by dbConnect
data	Trial design
extra_design	The extra columns for design

Value

no return values

---

dbAddFertilization *Insert or Update fertilization into expDB*

Description

Insert or Update fertilization into expDB

Usage

```
dbAddFertilization(con, data)
```

Arguments

con	a connection object as produced by dbConnect
data	A data frame includes all columns

Value

no return values

dbAddGene *Add gene information into database*

Description

Add gene information into database

Usage

```
dbAddGene(con, genes)
```

Arguments

con	a connection object as produced by dbConnect
genes	A data.frame of genes

Value

No return values

dbAddGeneAllele *Add gene allele information into database*

Description

Add gene allele information into database

Usage

```
dbAddGeneAllele(con, genes)
```

Arguments

con	a connection object as produced by dbConnect
genes	A data.frame of genes

Value

No return values

dbAddGenotype *Add genotypes into expDB*

Description

Add genotypes into expDB

Usage

```
dbAddGenotype(con, genotypes)
```

Arguments

con	a connection object as produced by dbConnect
genotypes	A string vector of genotypes

Value

No return values

dbAddIrrigatons *Insert or Update irrigation into expDB*

Description

Insert or Update irrigation into expDB

Usage

```
dbAddIrrigatons(con, data)
```

Arguments

con	a connection object as produced by dbConnect
data	A data frame includes all columns

Value

no return values

dbAddLog *Add log from expDB*

Description

Add log from expDB

Usage

```
dbAddLog(con, msg, date = format(Sys.time(), format = "%Y-%m-%d"))
```

Arguments

con	a connection object as produced by dbConnect
msg	Add message into expdb
date	Create time of message

Value

No return values

dbAddMethods *Insert or Update methods into expDB*

Description

Insert or Update methods into expDB

Usage

```
dbAddMethods(con, data)
```

Arguments

con	a connection object as produced by dbConnect
data	A data frame includes all columns

Value

no return values

dbAddMets *Insert and update met into expDB*

Description

Insert and update met into expDB

Usage

```
dbAddMets(con, data)
```

Arguments

con	a connection object as produced by dbConnect
data	Met design

Value

no return values

dbAddNodes *Add nodes into expDB*

Description

Add nodes into expDB

Usage

```
dbAddNodes(con, data)
```

Arguments

con	a connection object as produced by dbConnect
data	phenotype value

Value

no return values

dbAddPhenotype *Add design for a trial*

Description

Add design for a trial

Usage

```
dbAddPhenotype(con, data)
```

Arguments

con	a connection object as produced by dbConnect
data	phenotype value

Value

no return values

dbAddResearcher *Insert and update researcher into expDB*

Description

Insert and update researcher into expDB

Usage

```
dbAddResearcher(con, data)
```

Arguments

con	A connection object as produced by dbConnect
data	A data frame includes all columns

Value

no return values

dbAddSites *Insert or update site into expDB*

Description

Insert or update site into expDB

Usage

```
dbAddSites(con, data)
```

Arguments

con	a connection object as produced by dbConnect
data	A data frame includes all columns

Value

no return values

dbAddSource *Insert and update source into expDB*

Description

Insert and update source into expDB

Usage

```
dbAddSource(con, data)
```

Arguments

con	A connection object as produced by dbConnect
data	A data frame includes all columns

Value

no return values

dbAddTraits *Insert or update trait into expDB*

Description

Insert or update trait into expDB

Usage

```
dbAddTraits(con, data)
```

Arguments

con	a connection object as produced by dbConnect
data	A data frame includes all columns

Value

no return values

dbAddTrials *Insert or Update trial into expDB*

Description

Insert or Update trial into expDB

Usage

```
dbAddTrials(con, data)
```

Arguments

con	a connection object as produced by dbConnect
data	A data frame includes all columns

Value

no return values

dbAddTrialSoil *Add soil for a trial*

Description

Add soil for a trial

Usage

```
dbAddTrialSoil(
  con,
  data,
  units = list(thickness = "cm", no3 = "kg/ha", nh4 = "kg/ha")
)
```

Arguments

con	a connection object as produced by dbConnect
data	Soil profiles for trials
units	a list for the unit

Value

no return values

dbAddWeather *Add weather records into expDB*

Description

Add weather records into expDB

Usage

```
dbAddWeather(con, data, name = NULL)
```

Arguments

con	a connection object as produced by dbConnect
data	A string character for the path to met file, a WeaAna object, or a data frame.
name	The met name in the database if data is a data frame.

Value

no return values

dbAppendTable *Append a table into db and check the column name*

Description

Append a table into db and check the column name

Usage

```
dbAppendTable(con, table, data)
```

Arguments

con	A connection object as produced by dbConnect
table	The target table name
data	A data frame to write into table

dbExportMet *Export trials weather records to met file*

Description

Export trials weather records to met file

Usage

```
dbExportMet(con, output, na = NA, ...)
```

Arguments

con	a connection object as produced by dbConnect
output	The folder of output files
na	The character for missing value with default NA
...	All other arguments to define range of export trials. All trials will be export if there are not arguments. Supported arguments include trial (or trialcode)

Value

Write weather records into files. No return values.

dbGenotypeCheckName *Check genotype names*

Description

Check genotype names

Usage

```
dbGenotypeCheckName(con, genotype)
```

Arguments

con	a connection object as produced by dbConnect
genotype	The genotype name will be checked

Value

A vector with check genotype names

dbGetDryWeightPerStem *Get the dry weight per stem*

Description

Get the dry weight per stem

Usage

```
dbGetDryWeightPerStem(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected dry weight per stem

dbGetFertilization *Get fertilization from database*

Description

Get fertilization from database

Usage

```
dbGetFertilization(con, ...)
```

Arguments

con	a connection object as produced by dbConnect
...	Other arguments to specify meta data

Value

a data.frame for fertilization information

dbGetFieldMaturity *Estimation of maturity*

Description

Estimation of maturity

Usage

```
dbGetFieldMaturity(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected maturity time

dbGetFieldPopulation *Estimation of plant populations*

Description

Estimation of plant populations

Usage

```
dbGetFieldPopulation(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected field population

dbGetFieldStemNumber *Estimation of stem number per unit area*

Description

Estimation of stem number per unit area

Usage

```
dbGetFieldStemNumber(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected stem number

dbGetFieldTillerNumber *Estimation of tiller number per unit area*

Description

Estimation of tiller number per unit area

Usage

```
dbGetFieldTillerNumber(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected tiller number

dbGetGene *Get the gene information*

Description

Get the gene information

Usage

```
dbGetGene(con)
```

Arguments

con	a connection object as produced by dbConnect

Value

a data.frame with all gene information

dbGetGenotype *Get the genotype information*

Description

Get the genotype information

Usage

```
dbGetGenotype(con, name_only = FALSE)
```

Arguments

con	a connection object as produced by dbConnect
name_only	Only return the name of genotypes

Value

data.frame with genotype information or a vector with genotype name if name_only = TRUE.

dbGetIrrigation *Get irrigation from database*

Description

Get irrigation from database

Usage

```
dbGetIrrigation(con, ...)
```

Arguments

con	a connection object as produced by dbConnect
...	Other arguments to specify meta data

Value

a data frame for irrigation information

dbGetLog *Get log from expDB*

Description

Get log from expDB

Usage

```
dbGetLog(con)
```

Arguments

con	a connection object as produced by dbConnect

Value

A data.frame with all logs

dbGetMetInfo *Get met information*

Description

Get met information

Usage

```
dbGetMetInfo(con, name)
```

Arguments

con	a connection object as produced by dbConnect
name	The met name

Value

a data.frame for met information

dbGetOrganFinalLeafNumber *Get the final leaf number*

Description

The final leaf number is first retrived from trait "O_FinalLeafNumber", then calculated from trait "O_HaunIndex" if "O_FinalLeafNumber" is not observed. Final leaf number equals the maximum value of "O_HaunIndex", which should be an integer.

Usage

```
dbGetOrganFinalLeafNumber(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments
...	Arguments to specific trials.

Value

A data.frame for selected final leaf number

dbGetOrganHaunIndex *Get Haun Index*

Description

The Haun Index is retrived from trait "O_HaunIndex", extending the final observationa

Usage

```
dbGetOrganHaunIndex(con, trials = NULL, avg = TRUE, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments
avg	Whether to calculate the average value
...	Arguments to specific trials.

Value

A data.frame for selected Haun Index

dbGetPhenotype *Get phenotype values through a group of conditions*

Description

Get phenotype values through a group of conditions

Usage

```
dbGetPhenotype(
  con,
  traits = NULL,
  direction = "long",
  tt = FALSE,
  gene = FALSE,
  ...
)
```

Arguments

con	a connection object as produced by dbConnect
traits	A list of traits. All traits will be returned if NULL
direction	One of 'long' or 'wide' for reshape function
tt	Whether to calculate thermal time
gene	Whether to get gene information
...	All other arguments to define range of export trials

Value

a data.frame for selected phenotypic values

dbGetPlantFlowering *Estimation of flowering time*

Description

Estimation of flowering time

Usage

```
dbGetPlantFlowering(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected flowering time

---

dbGetPlantHeading *Estimation of heading time*

Description

Estimation of heading time

Usage

```
dbGetPlantHeading(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected heading time

---

dbGetPlantStemElongation *Estimation of stem elongation*

Description

Estimation of stem elongation

Usage

```
dbGetPlantStemElongation(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected stem elongation stage

---

dbGetPlantStemNumber *Estimation of stem number per plant*

Description

Estimation of stem number per plant

Usage

```
dbGetPlantStemNumber(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected stem number for individual plant

---

dbGetPlantTillerNumber *Estimation of tiller number per plant*

Description

Estimation of tiller number per plant

Usage

```
dbGetPlantTillerNumber(con, trials = NULL, ...)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
...	Arguments pass to dbGetTrials

Value

A data.frame for selected tiller number for individual plant

---

dbGetSites *Get site into expDB*

Description

Get site into expDB

Usage

```
dbGetSites(con)
```

Arguments

con	a connection object as produced by dbConnect

Value

a data.frame for all sites in the data base

---

dbGetSource *Get source from expDB*

Description

Get source from expDB

Usage

```
dbGetSource(con)
```

Arguments

con	a connection object as produced by dbConnect

Value

A data.frame for all source in the data base

dbGetTraits *Get trait list*

Description

Get trait list

Usage

```
dbGetTraits(con)
```

Arguments

con	a connection object as produced by dbConnect

Value

a data.frame for all traits in the data base

dbGetTrials *Get trials by a groups of conditions.*

Description

Get trials by a groups of conditions.

Usage

```
dbGetTrials(con, design = TRUE, ...)
```

Arguments

con	a connection object as produced by dbConnect
design	Whether include design
...	All other arguments to define range of export trials. All trials will be export if there are not arguments. Supported arguments include trial (or trialcode)

Value

A data.frame for selected trials

dbGetWeather *Get weather records from expDB*

Description

Get weather records from expDB

Usage

```
dbGetWeather(con, name, format = "data_frame", na = NA_character_, tz = "UTC")
```

Arguments

con	a connection object as produced by dbConnect
name	The met name
format	The format of export dataset.
na	The character for missing value with default NA
tz	Time zone applied for hourly temperature

Value

a data.frame for all weather records

dbGetZadoksStage *Obtain the key phenology stage*

Description

Obtain the key phenology stage

Usage

```
dbGetZadoksStage(con, trials, key_stage)
```

Arguments

con	a connection object as produced by dbConnect
trials	A data.frame to specify trials. If not NULL, other arguments will be ignored.
key_stage	The key zadoks stage

Value

A data.frame for selected Zadoks stage

dbImportXLSX *Import data from excel file*

Description

Import data from excel file

Usage

```
dbImportXLSX(con, xlsx, ignore_genotype = TRUE, ignore_trait = TRUE, ...)
```

Arguments

con	a connection object as produced by dbConnect
xlsx	The path to excel file
ignore_genotype	Ignore genotype tables when importing
ignore_trait	Ignore trait table when importing
...	Other arguments. Supported arguments include • extra_design: Extra columns in the experiment design. • tz: The time zone for the hourly climates.

Value

No return values

dbInsertUpdateByRow *Insert new rows or update existing rows to a specific table according to a specific column (unique) by each row*

Description

Insert new rows or update existing rows to a specific table according to a specific column (unique) by each row

Usage

```
dbInsertUpdateByRow(con, table, data, unique_col = "name")
```

Arguments

con	A connection object as produced by dbConnect
table	The target table name
data	A data frame to write into table
unique_col	A character vector to indentify each row in the table

dbListTrials *List all trials*

Description

List all trials

Usage

```
dbListTrials(con)
```

Arguments

con	a connection object as produced by dbConnect

Value

A data.frame for all trials in the data base

expdbConnect *Connect to expDB*

Description

Connect to expDB

Usage

```
expdbConnect(filename)
```

Arguments

filename	The filename of SQLite

Value

a connection object as produced by dbConnect

Examples

```
## Not run:
con <- connect('filename')

## End(Not run)
```

expdbCreateDB *create to expDB*

Description

create to expDB

Usage

```
expdbCreateDB(filename, system_traits = TRUE)
```

Arguments

filename	The filename of new expDB
system_traits	Whether to import system traits

Value

a connection object as produced by dbConnect

expdbDisconnect *Didconnect to expDB*

Description

Didconnect to expDB

Usage

```
expdbDisconnect(con)
```

Arguments

con	a connection object as produced by dbConnect.

Value

no return values

Examples

```
## Not run:
con <- connect('filename')
disconnect(con)

## End(Not run)
```

getIdByUniqueIndex *Get index id by unique_columns.*

Description

Get index id by unique_columns.

Usage

```
getIdByUniqueIndex(
  con,
  table,
  data,
  unique_col = "name",
  data_col = unique_col,
  ignore_case = FALSE
)
```

Arguments

con	a connection object as produced by dbConnect
table	the table name
data	A data frame to write into table
unique_col	A character vector to identify each row in the table
data_col	The column names in the data.frame
ignore_case	Whether ignore_case

Value

id of unique_col

harvestQuadratDetail *Process quadrat (detail) harvestQuadratDetail*

Description

Process quadrat (detail) harvestQuadratDetail

Usage

```
harvestQuadratDetail(con, records)
```

Arguments

con	a connection object as produced by dbConnect
records	Phenotype records

Value

no return values

Index