Package 'MLSP'

May 7, 2026

Type Package

Title Machine Learning Models for Soil Properties

Version 0.1.0

Description Creates a spectroscopy guideline with a highly accurate prediction model for soil properties using machine learning or deep learning algorithms such as LASSO, Random Forest, Cubist, etc., and decide which algorithm generates the best model for different soil types.

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.3

Imports gsignal, pls, glmnet, Cubist, randomForest

NeedsCompilation no

Author Pengyuan Chen [aut, cre],
Christopher Clingensmith [aut],
Chenglong Ye [aut],
Sabine Grunwald [aut],
Katsutoshi Mizuta [aut]

Maintainer Pengyuan Chen <pch276@uky.edu>

Repository CRAN

Date/Publication 2025-10-08 19:40:21 UTC

Contents

colsMean 2
merge_of_lab_and_spectrum 2
ml_f 4
msd.comp 5
raw 6
soil_preprocess 7

Index **10**

colsMean *Compute Row-wise Means for Groups of Columns*

Description

This function computes the row-wise mean for consecutive groups of columns in a matrix or data frame.

Usage

```
colsMean(x, ncols)
```

Arguments

x	A numeric matrix or data frame.
ncols	An integer specifying the number of consecutive columns to group together.

Details

The function splits the columns of `x` into consecutive groups of `ncols` columns and calculates the mean of each row for each group. The number of columns in `x` must be divisible by `ncols`.

Value

A numeric matrix with the same number of rows as `x` and `ncol(x) / ncols` columns, where each column is the row-wise mean of a group of `ncols` columns.

Examples

```
mat <- matrix(1:12, nrow = 3)
colsMean(mat, 2)
```

merge_of_lab_and_spectrum *Merge Soil Laboratory Data with Spectral Data*

Description

This function merges soil laboratory data with cleaned spectral (VNIR) data, performs preprocessing, and prepares inputs for calibration and model building.

Usage

```
merge_of_lab_and_spectrum(soil_data, data_NaturaSpec_cleaned)
```

Arguments

`soil_data`	A data frame containing soil laboratory measurements (must include a column named `LAB_NUM`).
`data_NaturaSpec_cleaned`	A data frame containing cleaned spectral data with columns `Wavelength`, `LAB_NUM`, and reflectance values.

Details

The function performs the following steps:

- Aggregates spectral data by wavelength and computes mean reflectance values.
- Merges the soil and spectral datasets by `LAB_NUM`.
- Separates soil variables and VNIR spectral matrix.
- Creates calibration sample indices using random sampling.
- Defines spectral bands to remove (detector artifact areas) and indices to be used in modeling.

Value

A list with the following elements:

soil Data frame of soil laboratory data (first 8 columns of merged dataset).

vnir.matrix Matrix of VNIR spectral reflectance values (without metadata columns).

j List of calibration sample indices for cross-validation (4 sets).

rm1, rm2, rm3, rm4 Vectors of indices corresponding to spectral bands to be removed (detector artifact regions around 1000 nm and 1800 nm).

ind Indices of spectral bands used for aggregation (columns 7–2146).

rmove Indices of bands to be excluded from analysis.

vars Vector of spectral band names retained after removal.

Examples

```
merged <- merge_of_lab_and_spectrum(soil_data, data_NaturaSpec_cleaned)
str(merged)
```

`ml_f` *Machine Learning Function for Soil Spectral Data*

Description

This function applies several machine learning models (PCR, PLSR, Random Forest, LASSO, Cubist) to soil spectral data and compares their performance. Optionally, it can return the best-performing model.

Usage

```
ml_f(
  x,
  y,
  smoother_selection,
  type_of_soil,
  model_selection = TRUE
)
```

Arguments

`x`	A data frame or matrix containing spectral data.
`y`	A vector containing corresponding soil laboratory measurements.
`smoother_selection`	A parameter specifying the smoothing method to be applied during preprocessing.
`type_of_soil`	A character string indicating the soil type for model calibration.
`model_selection`	Logical; if 'TRUE' (default), the function returns only the best-performing model. If 'FALSE', it returns the results from all models.

Details

The function merges spectral and laboratory data, preprocesses the data, and evaluates the following models:

- PCR (Principal Component Regression)
- PLSR (Partial Least Squares Regression)
- RF (Random Forest)
- LASSO regression
- Cubist regression

Each model's performance results are combined into a single results object. If `model_selection = TRUE`, the function returns the model with the highest performance metric (based on the 11th column of the results table).

Value

A data frame:

If `model_selection = FALSE` Returns results for all models.

If `model_selection = TRUE` Returns only the best-performing model result.

Examples

```
# Example usage:
results <- ml_f(
  x,
  y,
  smoother_selection = "savitzky",
  type_of_soil = "loam",
  model_selection = TRUE
)
```

`msd.comp` *Compute Model Evaluation Metrics*

Description

This function computes various statistics for comparing observed values ‘y‘ with predicted values ‘yhat‘. It includes correlation, regression coefficients, bias, RMSE, MSE, and predictive performance metrics like RPD and RPIQ.

Usage

```
msd.comp(y, yhat)
```

Arguments

`y`	Numeric vector of observed values.
`yhat`	Numeric vector of predicted values (same length as ‘y‘).

Value

A named numeric vector with the following components:

r Pearson correlation between ‘y‘ and ‘yhat‘

int Intercept of regression of ‘y‘ on ‘yhat‘

slope Slope of regression of ‘y‘ on ‘yhat‘

r2 Coefficient of determination (R-squared)

bias Mean bias: mean(yhat) - mean(y)

rmse Root mean squared error

mse Mean squared error

sb Systematic bias component of MSE

nu Non-unity slope component of MSE

lc Lack-of-correlation component of MSE

rmse.c Corrected RMSE after removing bias

mse.c Corrected MSE after removing bias

rpd Ratio of standard deviation to RMSE (RPD)

rpiq Ratio of interquartile range to RMSE (RPIQ)

Examples

```
y_obs <- c(1.2, 3.4, 2.5, 4.1)
y_pred <- c(1.1, 3.5, 2.4, 4.0)
msd.comp(y_obs, y_pred)
```

`raw` *Aggregate VNIR Spectra by Columns*

Description

This function aggregates VNIR (Visible and Near-Infrared) spectral data by calculating the mean of every 10 columns while removing specific detector artifact regions (~1000nm and ~1800nm) and unwanted spectral bands.

Usage

```
raw(vnir.matrix)
```

Arguments

`vnir.matrix`	A numeric matrix or data frame containing VNIR spectral data. Each row corresponds to a sample, and each column corresponds to a spectral band.

Details

The function removes columns corresponding to detector artifacts: - rm1: bands 1–46 - rm2: bands 637–666 - rm3: bands 1437–1466 - rm4: bands 2127–2151 Additionally, columns 1:4, 64:66, 144:146, and 213:214 (after averaging) are removed.

Value

A data frame containing the aggregated VNIR spectra with cleaned band names. The number of columns is reduced by averaging over every 10 bands and removing artifact-prone regions.

Examples

```
raw_spectra <- raw(vnir_matrix)
```

`soil_preprocess` *Soil and Spectral Data Preprocessing for Model Training*

Description

These functions fit predictive models for soil properties using VNIR spectral data. Each function applies a specific machine learning method:

- `pcr_preprocess()` – Principal Component Regression (PCR)
- `plsr_preprocess()` – Partial Least Squares Regression (PLSR)
- `lasso_preprocess()` – LASSO regression
- `rf_preprocess()` – Random Forest regression
- `cubist_preprocess()` – Cubist regression

Computes mean performance metrics across multiple calibration and validation sets. Typically used to summarize the results of soil property prediction models generated by preprocessing functions such as `pcr_preprocess()`, `plsr_preprocess()`, `lasso_preprocess()`, `rf_preprocess()`, or `cubist_preprocess()`.

Usage

```
pcr_preprocess(soil, vnir.matrix, j, preprocess, type_of_soil)

plsr_preprocess(soil, vnir.matrix, j, preprocess, type_of_soil)

lasso_preprocess(soil, vnir.matrix, j, preprocess, type_of_soil)

rf_preprocess(soil, vnir.matrix, j, preprocess, type_of_soil)

cubist_preprocess(soil, vnir.matrix, j, preprocess, type_of_soil)

results(metric.list, soil_type)
```

Arguments

`soil`	A data frame of soil properties. Must include the target soil variable.
`vnir.matrix`	A numeric matrix of VNIR spectral data.
`j`	A list of index vectors specifying calibration sample sets (e.g., from `merge_of_lab_and_spectrum`).
`preprocess`	A preprocessing function to apply to the spectral data (e.g., smoothing, normalization).

`type_of_soil`	An integer index selecting which soil property column to model.
`metric.list`	A list of MSD metric objects returned by one of the preprocessing/model functions. Each element corresponds to a model fit on a calibration/validation split.
`soil_type`	Optional, an integer or string indicating which soil property was modeled (currently not used internally but kept for consistency).

Details

All functions use the same workflow:

1. Combine the selected soil property with preprocessed spectra.
2. Split data into calibration and validation sets (using sample indices).
3. Fit the chosen model across multiple calibration/validation partitions.
4. Generate predictions and compute performance metrics (MSD-based).

Value

A list of MSD metric objects for calibration and validation sets, specific to the fitted model.

A named numeric vector of mean performance metrics across all splits:

LV Latent variable / model index

cv-r2 Cross-validated R-squared for calibration set

cv-bias Bias in cross-validation for calibration set

cv-rmse Root mean squared error in cross-validation for calibration set

cal-mse Mean squared error for calibration set

cal-rpiq Ratio of performance to interquartile distance for calibration set

val-r2 R-squared for validation set

val-bias Bias for validation set

val-rmse Root mean squared error for validation set

val-mse Mean squared error for validation set

val-rpiq Ratio of performance to interquartile distance for validation set

See Also

`merge_of_lab_and_spectrum`, `ml_f`

Examples

```
# Example with PCR
results_pcr <- pcr_preprocess(soil, vnir.matrix, j, preprocess = scale, type_of_soil = 2)

# Example with Random Forest
results_rf <- rf_preprocess(soil, vnir.matrix, j, preprocess = scale, type_of_soil = 2)
```

```
msd_list <- pcr_preprocess(soil, vnir.matrix, j, preprocess = scale, type_of_soil = 2)
results_summary <- results(msd_list)
```

Index

colsMean, 2
cubist_preprocess *(*soil_preprocess*)*, 7

lasso_preprocess *(*soil_preprocess*)*, 7

merge_of_lab_and_spectrum, 2, *7*, *8*
ml_f, 4, *8*
msd.comp, 5

pcr_preprocess *(*soil_preprocess*)*, 7
plsr_preprocess *(*soil_preprocess*)*, 7

raw, 6
results *(*soil_preprocess*)*, 7
rf_preprocess *(*soil_preprocess*)*, 7

soil_preprocess, 7