simplerspec devtools::install_github("philipp-baumann/simplerspec")

Contact: philipp.baumann@usys.ethz.ch
Sustainable Agroecosystems group, ETH Zürich

R package that facilitates spectroscopy data handling, processing and modeling

Spectra, metadata, chemical reference data

`spectra and reference data tibble (spc_ref_tbl)`

simplerspec
spectroscopy modeling

fit_pls()

R list of model outputs

`pls_<response_variable>`

`$data`

`$predobs` data.frame with predicted and observed values

`$stats` data.frame

```
 model     dataType       rmse       rmsd        msd       sdev       rpd       rpiq         r2       bias
1   pls  Calibration 0.9915364 0.9836356 0.967539 5.733394 5.782333 9.272983 0.9700916 0.0000000
2   pls   Validation 1.3772712 1.3548750 1.835686 5.784446 4.199933 4.843031 0.9463123 0.1160055
         SB          NU          LC SB_prop NU_prop LC_prop  n         b ncomp variable min_obs
1 0.00000000 3.868190e-29 0.967539        0       0       100 63 1.00000     6        C   2.404
2 0.01345728 8.379626e-02 1.738433        1       5        95 31 1.05518     6        C   3.413
  median_obs max_obs  mean_obs       CV
1      9.191  24.558 10.332529 55.48877
2      7.450  24.687  9.115806 63.45512
```

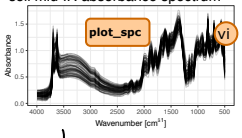
Model evaluation statistics including measures describing the distribution of chemical reference data

`$p_pc` ggplot2 graph

`$pls_model` caret::train() output of class "train"

`$p_model` ggplot2 graph

soil mid-IR absorbance spectrum

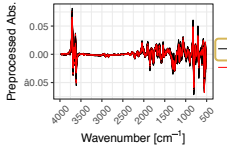


join_chem_spc

Join spectra tibble (spc_tbl) resulting from simplerspec data processing with reference data from traditional laboratory analyses

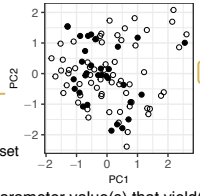
simplerspec
spectra processing

Processed spectra



Calibration sampling

PCA of preprocessed spectra (in addition center data)



> Kennard-Stone sampling algorithm or random data split

- calibration
- validation

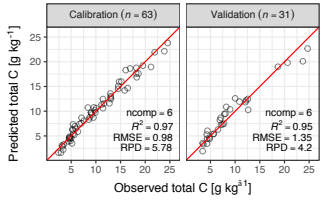
scale processed spectra

Perform model tuning on calibration (training) set

`spectra tibble (spc_tbl)`

Model tuning

Find model's parameter value(s) that yield(s) best and most realistic predictive performance

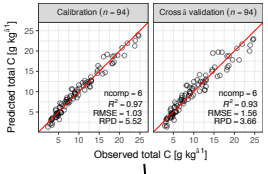


Derive estimates of model performance

6b Evaluation of final model using cross-validation resampling statistics

> Use the entire data set for model parameter tuning and model performance assessment

Calculate resampling statistics based on hold-out predictions (cross-validation at final model number of components)



6a Evaluation of final model based on test set (validation) predictions

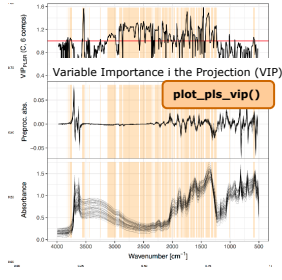
5 Refit the model using the entire calibration (training) data set with the final tuning parameter(s)

Model diagnostics and application

> Determine likelihood that the new samples follow data-generating mechanism exploited in calibration (training) model.

Interprete the model and understand mechanisms that allow predictions

> Detect extrapolation within or outside of predictor (preprocessed spectra) space and gain knowledge on the data generating mechanisms (consider factors associated with spatial and temporal variability that influence relation between spectra and sample properties to be predicted).



i **read_opus_univ()**
Read pectra data and metadata from single OPUS binary files

ii **gather_spc()**
Gather spectra from list of spectral data into a tibble object

iii **resample_spc()**
Resample spectra stored in tibble column after gathering spectra

iv **average_spc()**
Average spectra in tibble column by sample_id after resampling spectra

v **preprocess_spc()**
Preprocesses spectra in tibble column after averaging spectra (list-column <spc_pre>)

1 Define a candidate set of model parameter values

Selection of model tuning parameter sets

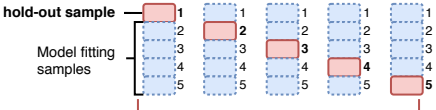
e.g. for PLS regression:
#Components (ncomp)
1 2 3 4 5 6 ...

2 Fit multiple predictive models (e.g. PLS regression) for all tuning candidate sets and k folds (split by resampling technique) using the model fitting samples

Resample the training set repeatedly (ncomp times)

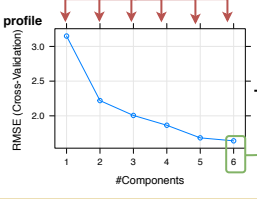
One of possible resampling techniques used to estimate model performance:

***K*-fold cross-validation** randomly split data into *K* subsets



Calculate performance measure (RMSE) on hold out samples using predictors (preprocessed spectra).

x (# tuning parameter sets)



3 Assess performance on hold-out samples by predicting hold-out samples for all k folds by model fitted on remaining model fitting samples

Summary of model fitting process

```
> print(pls_C$pls_model)
Partial Least Squares

  63 samples
1729 predictors

Pre-processing: centered (1729), scaled (1729)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 55, 58, 58, 59, 57, 58, ...
Resampling results across tuning parameters:

  ncomp  RMSE      Rsquared
  1      3.149334  0.7666746
  2      2.219748  0.8498336
  3      2.005840  0.8783200
  4      1.864283  0.9016300
  5      1.682929  0.9207747
  6      1.639983  0.9159574

RMSE was used to select the optimal model using the
smallest value.
The final value used for the model was ncomp = 6.
```

4 Choose the final model turning parameters