



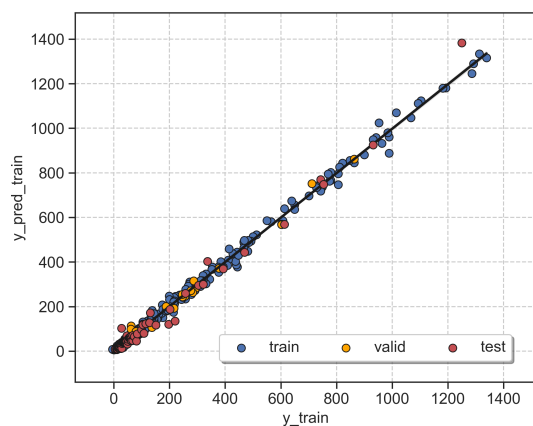
ROBERT v 1.2.1 2025/06/06 15:28:20

How to cite: Dalmau, D.; Alegre Requena, J. V. WIREs Comput Mol Sci. 2024, DOI: 10.1002/WCMS.1733**Section A. ROBERT Score***This score is designed to evaluate the models using different metrics.***No PFI (standard descriptor filter):**

Model = NN · Train:Validation:Test = 81:9:10

Points(train+valid.):descriptors = 597:198

Score = 9 / 10

**STRONG**

Train : $R^2 = 1.0$, MAE = 8.1, RMSE = 1.3e+01
 Valid. : $R^2 = 0.99$, MAE = 9.7, RMSE = 1.5e+01
 Test : $R^2 = 0.99$, MAE = 1.6e+01, RMSE = 2.8e+01

Severe warnings

- ☒ No severe warnings detected

Moderate warnings

- ☐ Uneven y distribution (Section C)
- ☐ Highly correlated features (Section D)
- ☐ Potential "faulty" outliers (Section E)

Overall assessment

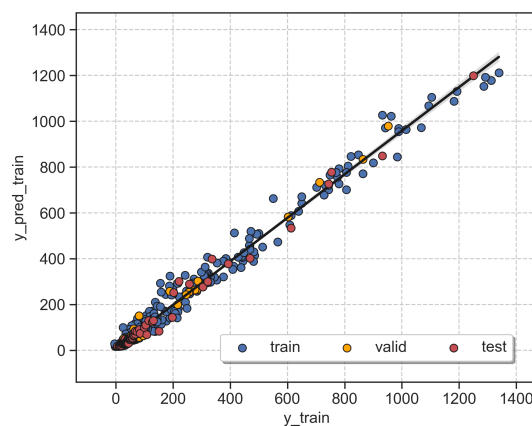
- ☐ Reliable model, but examine warnings

PFI (only most important descriptors):

Model = RF · Train:Validation:Test = 81:9:10

Points(train+valid.):descriptors = 597:68

Score = 10 / 10

**STRONG**

Train : $R^2 = 0.99$, MAE = 1.7e+01, RMSE = 2.9e+01
 Valid. : $R^2 = 0.99$, MAE = 1.2e+01, RMSE = 1.8e+01
 Test : $R^2 = 0.99$, MAE = 1.6e+01, RMSE = 2.7e+01

Severe warnings

- ☒ No severe warnings detected

Moderate warnings

- ☐ Uneven y distribution (Section C)
- ☐ Moderately correlated features (Section D)

Overall assessment

- ☒ The model seems reliable



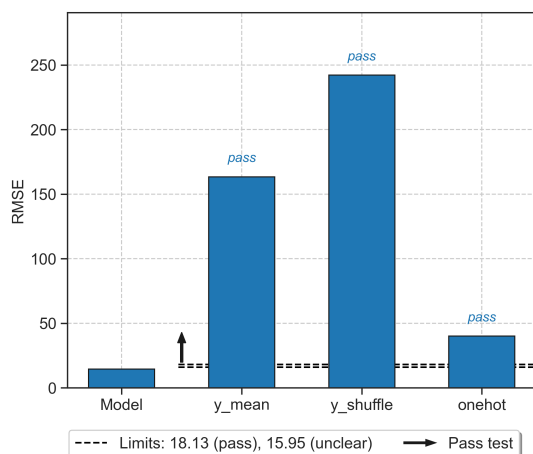
Section B. Advanced Score Analysis

This section explains each component that comprises the ROBERT score.

1. Model vs "flawed" models (3 / 3)

The model predicts right for the right reasons.

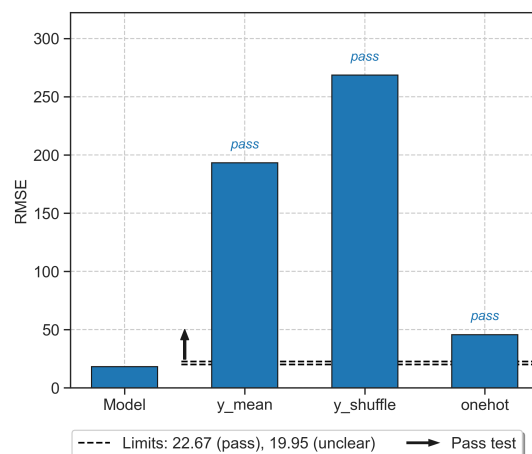
Pass: +1, Unclear: 0, Fail: -1. [Details here.](#)



1. Model vs "flawed" models (3 / 3)

The model predicts right for the right reasons.

Pass: +1, Unclear: 0, Fail: -1. [Details here.](#)



2. Predictive ability of the model (2 / 2)

Good predictive ability with R^2 (test) = 0.99.

R^2 0.70-0.85: +1, R^2 >0.85: +2.

2. Predictive ability of the model (2 / 2)

Good predictive ability with R^2 (test) = 0.99.

R^2 0.70-0.85: +1, R^2 >0.85: +2.

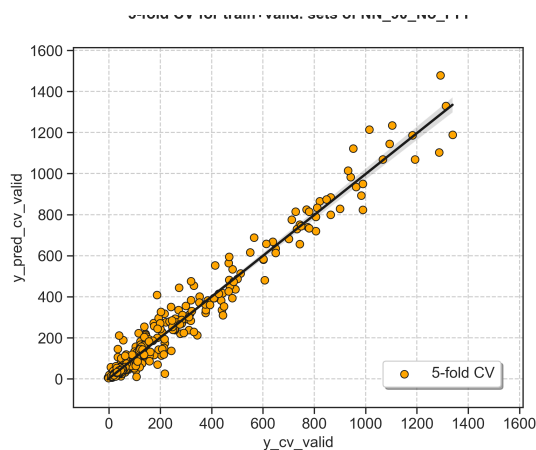
3. Cross-validation (5-fold CV) of the model

Overfitting analysis on the model with 3a and 3b:

3a. CV predictions train + valid. (2 / 2)

Good predictive ability with R^2 (5-fold CV) = 0.97.

R^2 0.70-0.85: +1, R^2 >0.85: +2.



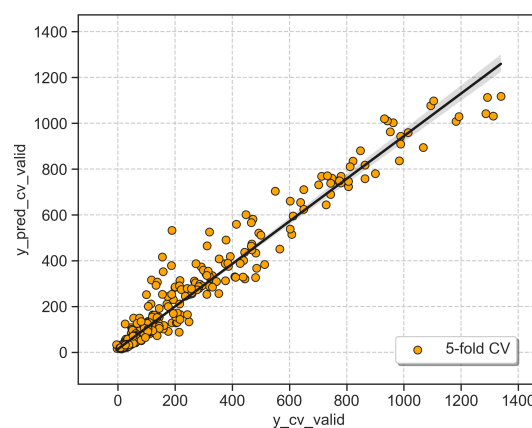
3. Cross-validation (5-fold CV) of the model

Overfitting analysis on the model with 3a and 3b:

3a. CV predictions train + valid. (2 / 2)

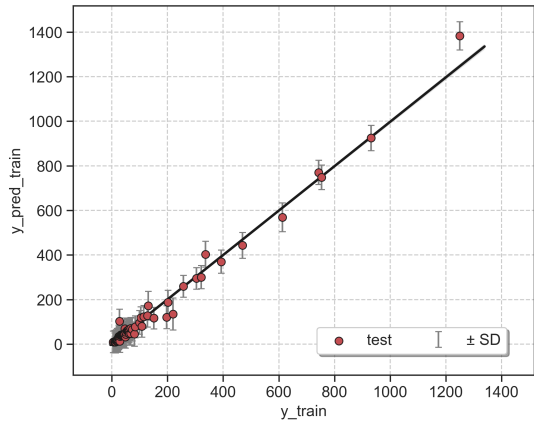
Good predictive ability with R^2 (5-fold CV) = 0.96.

R^2 0.70-0.85: +1, R^2 >0.85: +2.



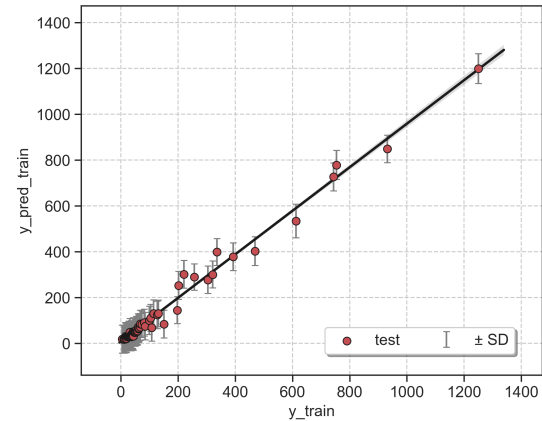
3b. Avg. standard deviation (SD) (2 / 2)

Low variation, 4*SD (test) = 205.2 (15% y-range).
4*SD 25-50% y-range: +1, 4*SD < 25% y-range: +2.
[Details here.](#)



3b. Avg. standard deviation (SD) (2 / 2)

Low variation, 4*SD (test) = 241.3 (18% y-range).
4*SD 25-50% y-range: +1, 4*SD < 25% y-range: +2.
[Details here.](#)



4. Points(train+valid.):descriptors (0 / 1)

Number of descps. could be lower (ratio 597:198).
5 or more points per descriptor: +1.

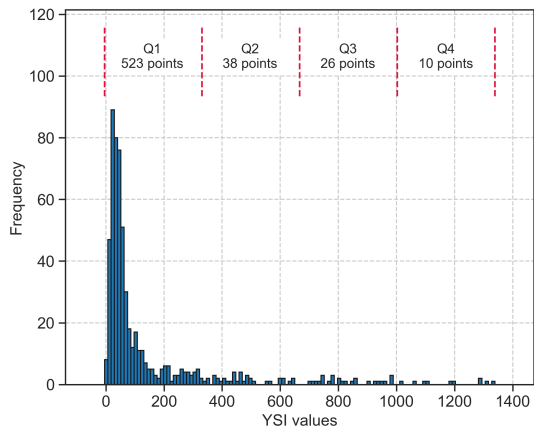
4. Points(train+valid.):descriptors (1 / 1)

Decent number of descps. (ratio 597:68).
5 or more points per descriptor: +1.



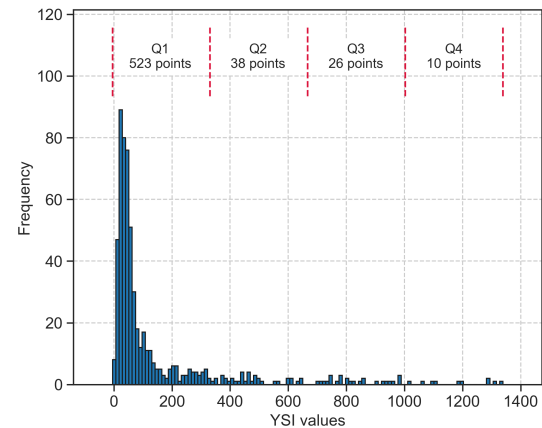
Section C. Distribution of y Values

This section shows the distribution of y values within the training and validation sets.



y distribution analysis

x WARNING! Your data is not uniform (Q4 has 10 points while Q1 has 523)



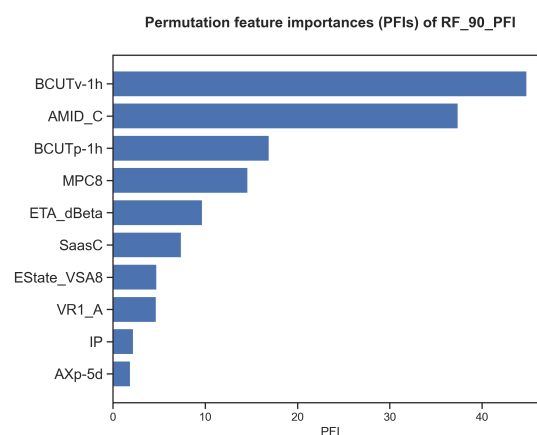
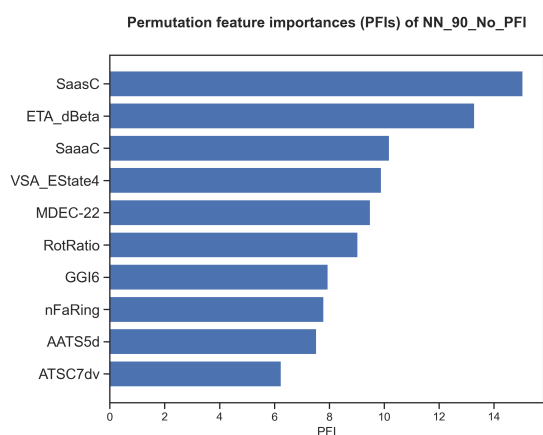
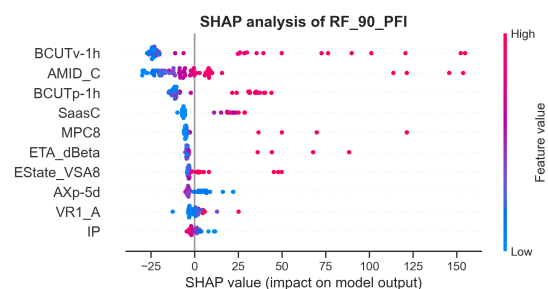
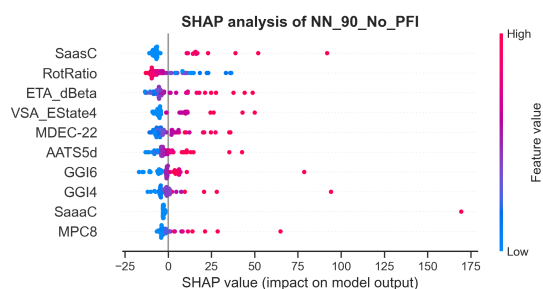
y distribution analysis

x WARNING! Your data is not uniform (Q4 has 10 points while Q1 has 523)



Section D. Feature Importances

This section presents feature importances measured using the validation set.



Pearson maps not created if >30 descriptors.
descriptors.

Pearson maps not created if >30

Correlation analysis

x WARNING! High correlations observed (up to $r = 0.85$ or $R^2 = 0.73$, for nFaRing and n10FaRing)

Correlation analysis

x WARNING! Noticeable correlations observed (up to $r = -0.83$ or $R^2 = 0.69$, for BCUTv-1l and RNCG)



Section E. Outlier Analysis

This section detects outliers using the standard deviation (SD) of errors from the training set.

No PFI (standard descriptor filter):

Outliers (max. 10 shown)

Train: 22 outliers out of 537 datapoints (4.1%)

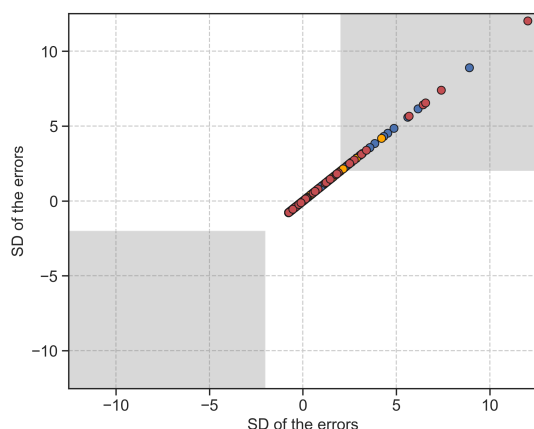
- M61 (3.2 SDs)
- M68 (4.5 SDs)
- M70 (8.9 SDs)
- M72 (2.2 SDs)
- M73 (6.2 SDs)
- M83 (4.9 SDs)
- M100 (2.6 SDs)
- M104 (2.3 SDs)
- M107 (2.6 SDs)
- M113 (2.4 SDs)

Validation: 6 outliers out of 60 datapoints (10.0%)

- M96 (3.1 SDs)
- M105 (2.6 SDs)
- M153 (2.0 SDs)
- M225 (4.2 SDs)
- M230 (2.9 SDs)
- M255 (2.2 SDs)

Test: 9 outliers out of 66 datapoints (13.6%)

- M247 (6.4 SDs)
- M419 (2.7 SDs)
- M187 (3.1 SDs)
- M256 (2.5 SDs)
- M370 (6.6 SDs)
- M355 (7.4 SDs)
- M137 (5.7 SDs)
- M102 (3.4 SDs)
- M62 (1.2e+01 SDs)



PFI (only most important descriptors):

Outliers (max. 10 shown)

Train: 31 outliers out of 537 datapoints (5.8%)

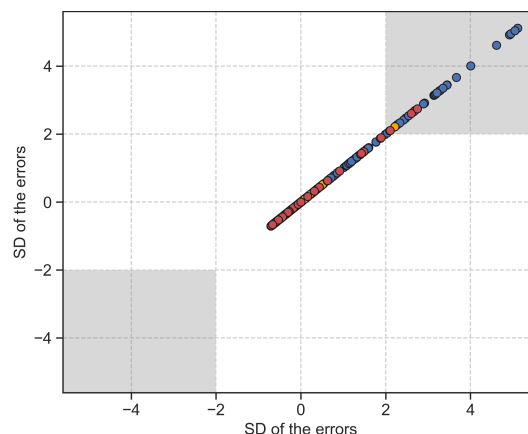
- M58 (4.6 SDs)
- M59 (4.9 SDs)
- M60 (3.4 SDs)
- M61 (4.9 SDs)
- M64 (3.3 SDs)
- M67 (3.3 SDs)
- M71 (5.1 SDs)
- M75 (3.3 SDs)
- M77 (2.7 SDs)
- M78 (3.1 SDs)

Validation: 2 outliers out of 60 datapoints (3.3%)

- M53 (2.2 SDs)
- M175 (2.2 SDs)

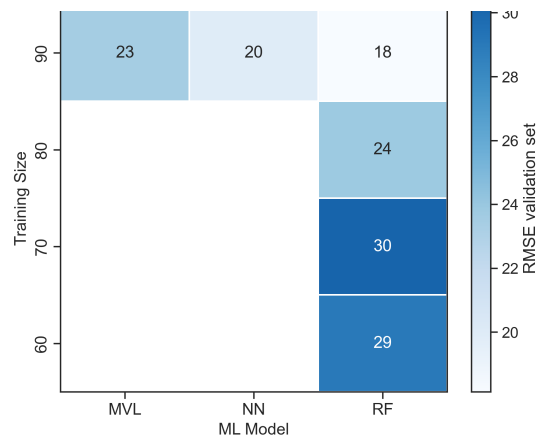
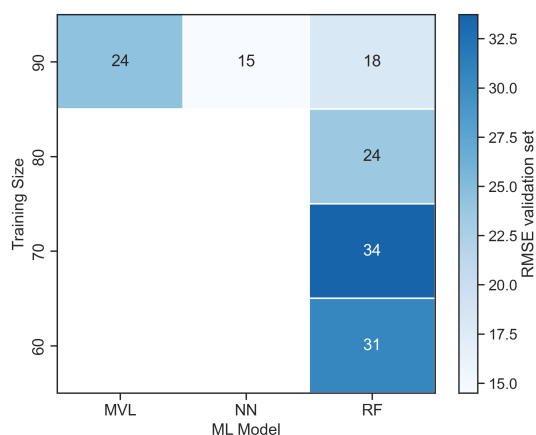
Test: 5 outliers out of 66 datapoints (7.6%)

- M256 (2.1 SDs)
- M355 (2.7 SDs)
- M115 (2.1 SDs)
- M102 (2.6 SDs)
- M76 (2.7 SDs)



Section F. Model Screening

This section compares different combinations of hyperoptimized algorithms and partition sizes.



Section G. Reproducibility

This section provides all the instructions to reproduce the results presented.

1. Download these files (*the authors should have uploaded the files as supporting information!*):

- CSV database (YSI-mordred_QM.csv)

2. Install and adjust the versions of the following Python modules:

- Install ROBERT and its dependencies: `conda install -y -c conda-forge robert`
- Adjust ROBERT version: `pip install robert==1.2.1`
- Install scikit-learn-intelex: `pip install scikit-learn-intelex==2024.7.0`

(if scikit-learn-intelex is not installed, slightly different results might be obtained)

3. Run ROBERT using this command line in the folder with the CSV database:

```
python -m robert --names "Name" --y "YSI" --csv_name "YSI-mordred_QM.csv"
```

4. Execution time, Python version and OS:

Originally run in Python 3.12.2 using Linux #1 SMP Fri Apr 20 16:44:24 UTC 2018

Total execution time: 3766.53 seconds (*the number of processors should be specified by the user*)



Section H. Transparency

This section contains important parameters used in scikit-learn models and ROBERT.

1. Parameters of the scikit-learn models (same keywords as used in scikit-learn):

No PFI (standard descriptor filter):

sklearn model: MLPRegressor
 random_state: 233
 names: Name
 batch_size: 16
 hidden_layer_sizes: [16, 16]
 learning_rate_init: 0.01
 max_iter: 50
 validation_fraction: 0.2
 alpha: 0.0001
 shuffle: True
 tol: 0.0001
 early_stopping: False
 beta_1: 0.999
 beta_2: 0.999
 epsilon: 1e-08

PFI (only most important descriptors):

sklearn model: RandomForestRegressor
 random_state: 19
 names: Name
 n_estimators: 60
 max_depth: 5
 max_features: 0.75
 min_samples_split: 2
 min_samples_leaf: 1
 min_weight_fraction_leaf: 0
 ccp_alpha: 0
 oob_score: False
 max_samples: 0.5

2. ROBERT options for data split (KN or RND), predict type (REG or CLAS) and hyperopt error (RMSE, etc.):

No PFI (standard descriptor filter):

split: KN
 type: reg
 error_type: rmse

PFI (only most important descriptors):

split: KN
 type: reg
 error_type: rmse



Section I. Abbreviations

Reference section for the abbreviations used.

ACC: accuracy	KN: k-nearest neighbors	REG: Regression
ADAB: AdaBoost	MAE: root-mean-square error	RF: random forest
CSV: comma separated values	MCC: Matthew's correl. coefficient	RMSE: root mean square error
CLAS: classification	ML: machine learning	RND: random
CV: cross-validation	MVL: multivariate lineal models	SHAP: Shapley additive explanations
F1 score: balanced F-score	NN: neural network	VR: voting regressor
GB: gradient boosting	PFI: permutation feature importance	
GP: gaussian process	R2: coefficient of determination	

Miscellaneous

General tips to improve the models and instructions to predict new values.

Some general tips to improve the score

1. Adding meaningful datapoints might help to improve the model. Also, using a uniform population of datapoints across the whole range of y values usually helps to obtain reliable predictions across the whole range. More information about the range of y values used is available in Section C.
2. Adding meaningful descriptors or replacing/deleting the least useful descriptors used might help. Feature importances are gathered in Section D.

How to predict new values with these models?

1. Create a CSV database with the new points, including the necessary descriptors.
 2. Place the CSV file in the parent folder (i.e., where the module folders were created)
 3. Run the PREDICT module as 'python -m robert --predict --csv_test FILENAME.csv'.
 4. The predictions will be shown at the end of the resulting PDF report and will be stored in the last column of two CSV files called MODEL_SIZE_test(_No)_PFI.csv, which are in the PREDICT folder.
-