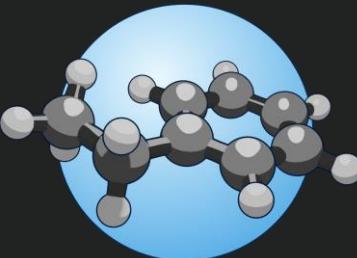


ROBERT: Bringing Machine Learning Closer to Chemists

David Dalmau Ginesta

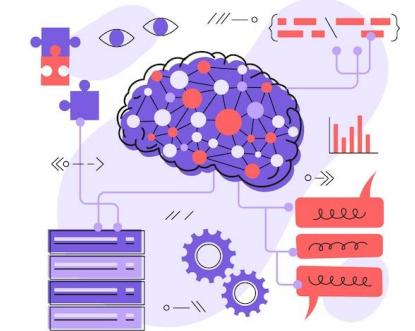
06/19/2024

CAMLC24



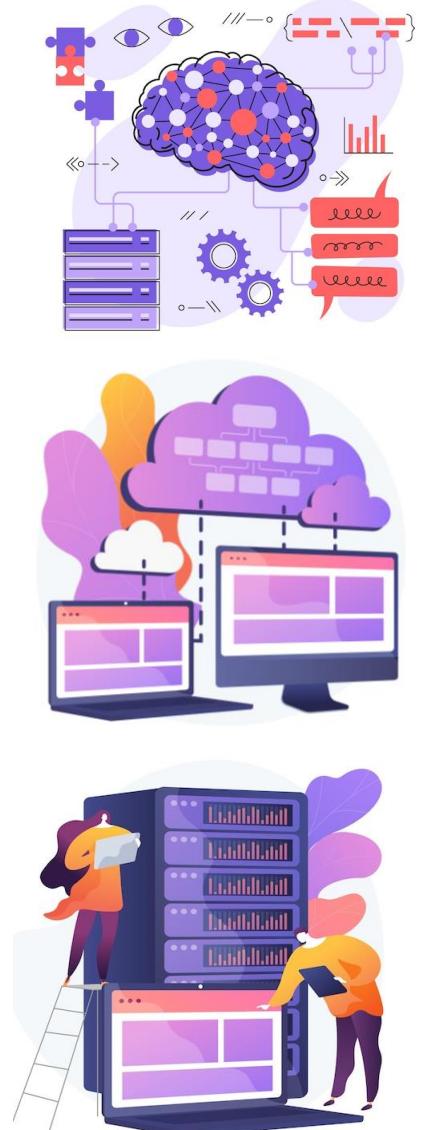
Why should I automate ML protocols?

- **Algorithmic innovation**
- **Data availability**
- **Computer power**

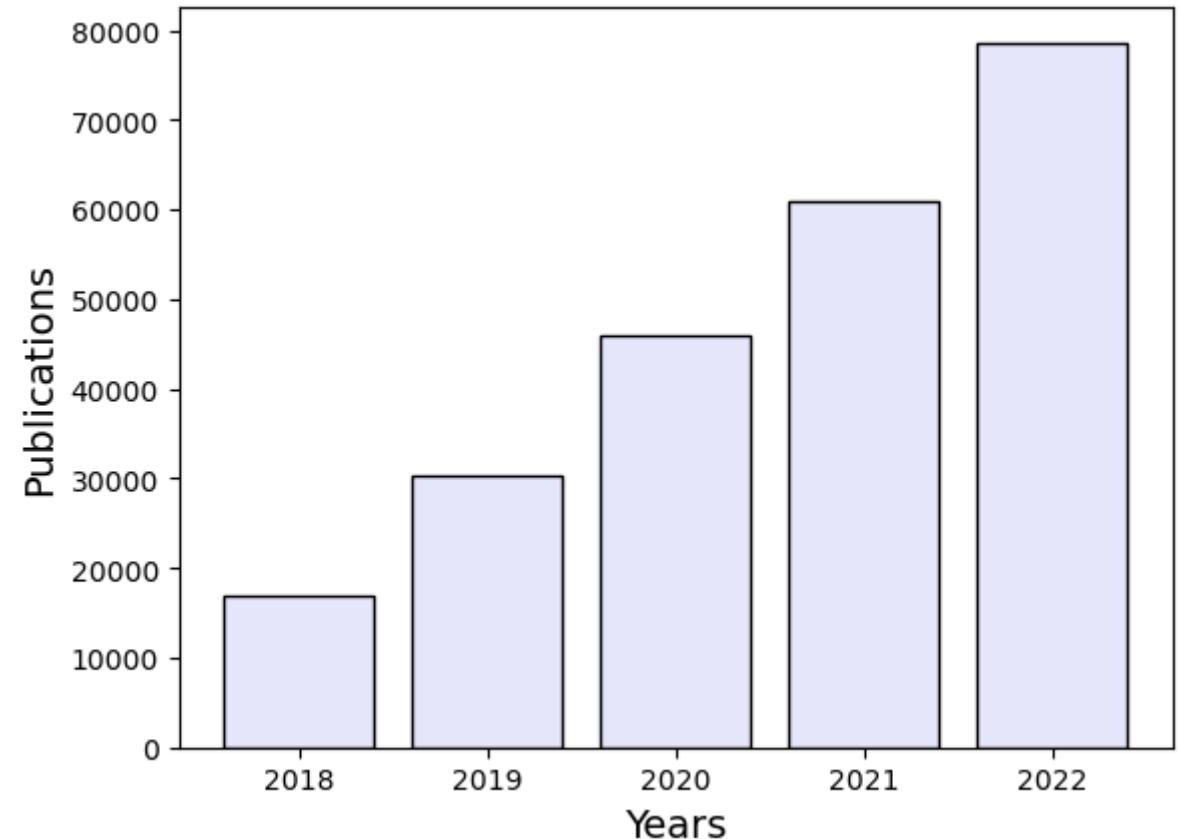


Why should I automate ML protocols?

- **Algorithmic innovation**
- **Data availability**
- **Computer power**



“Machine Learning” search in Reaxys



Year 2022: 78622 publications (≈ 215 /day)

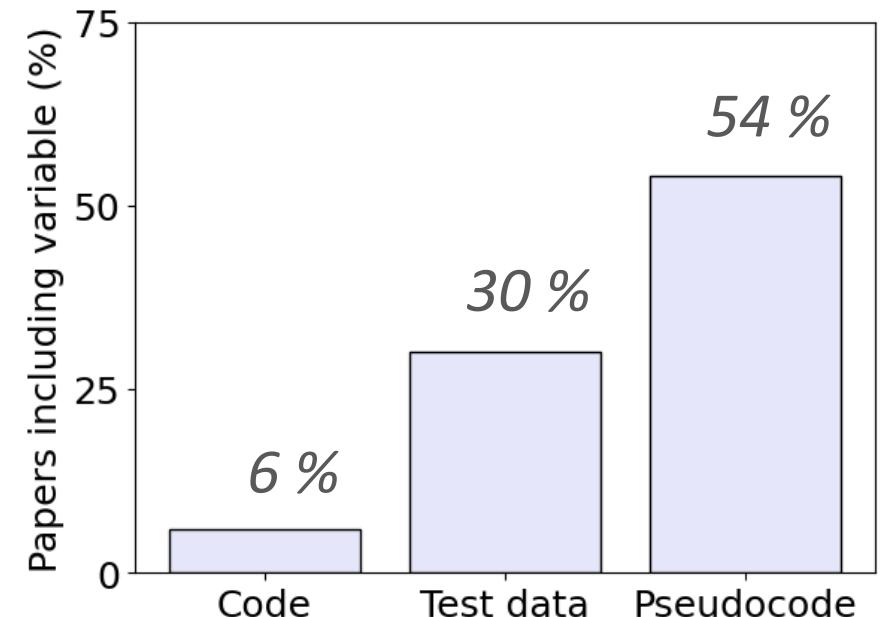
Why should I automate ML protocols?

“The growth of machine learning and making it FAIR”

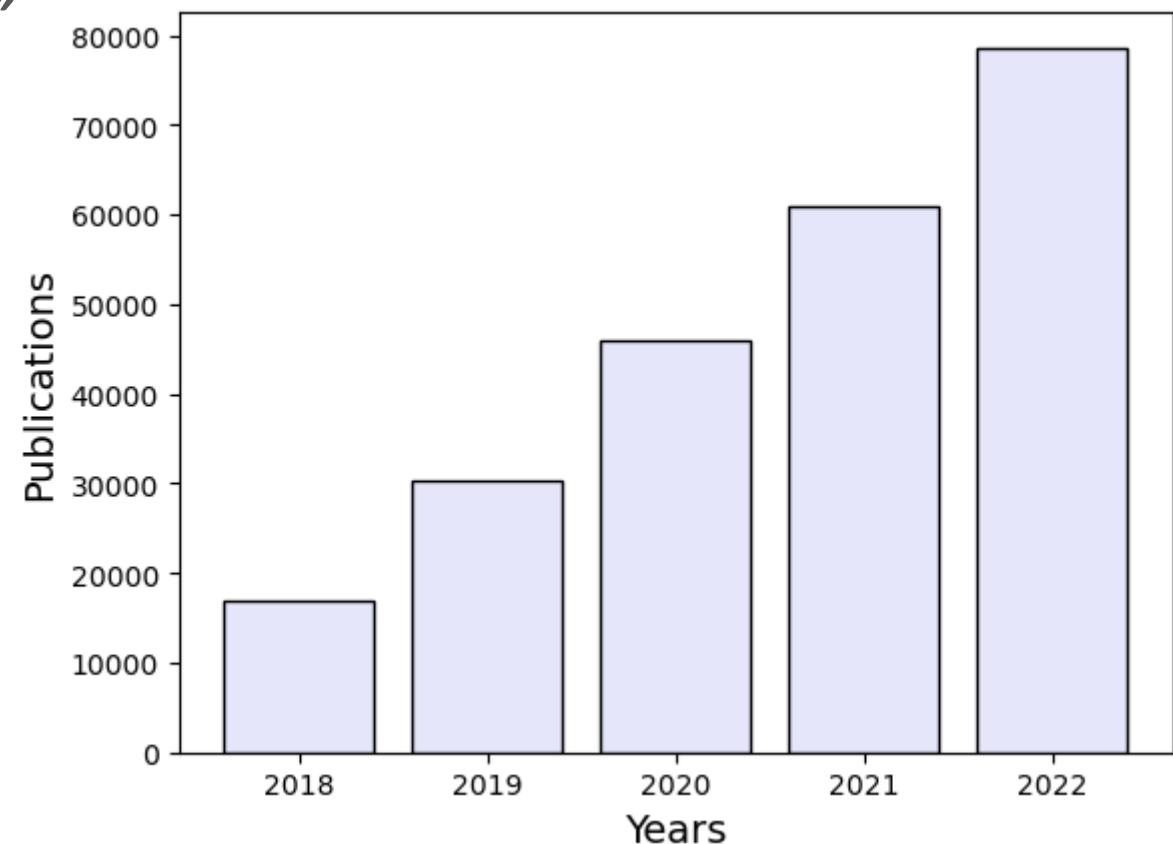
Nat. Chem. **13**, 505–508 (2021).

“Artificial intelligence faces reproducibility crisis”

Science **359**, 725-726(2018).



“Machine Learning” search in Reaxys

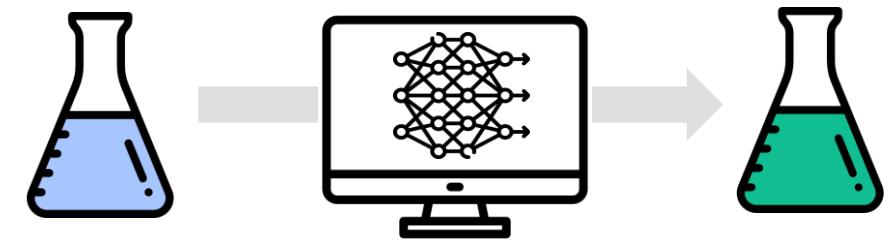


Year 2022: 78622 publications (≈ 215 /day)

Why should I automate ML protocols?

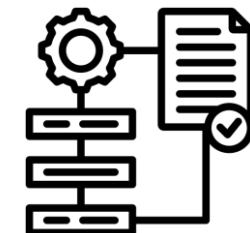
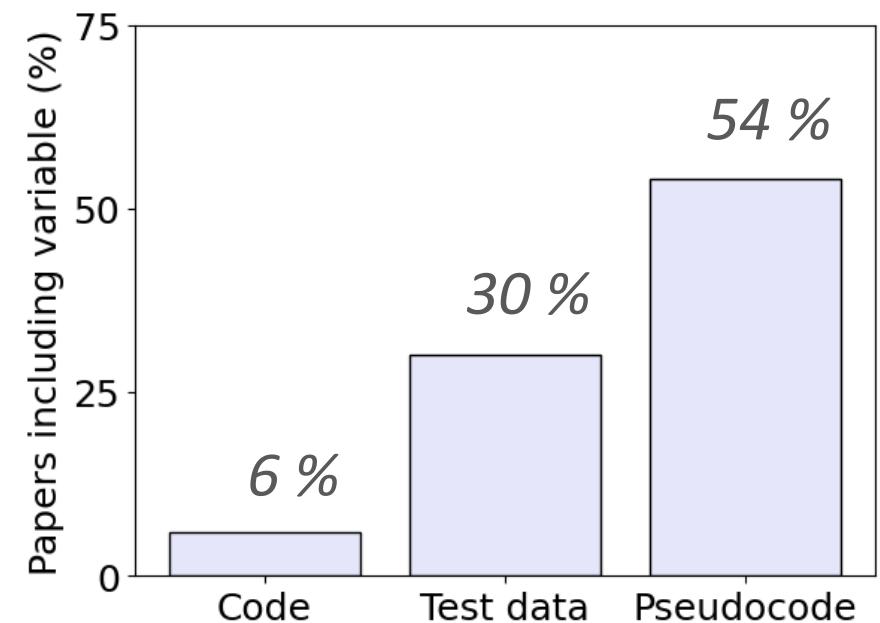
“The growth of machine learning and making it FAIR”

Nat. Chem. **13**, 505–508 (2021).



“Artificial intelligence faces reproducibility crisis”

Science **359**, 725-726(2018).



Protocols

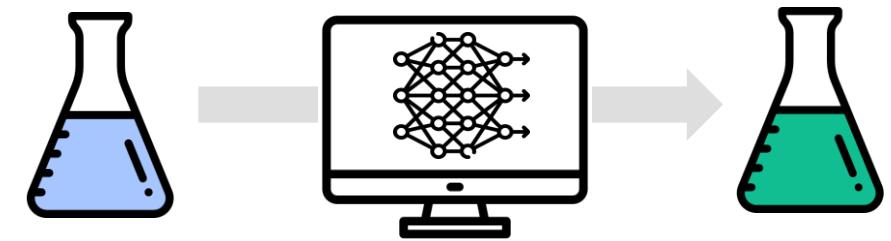


Standards

Why should I automate ML protocols?

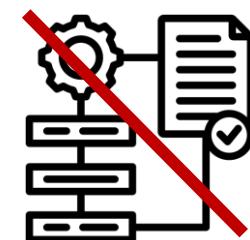
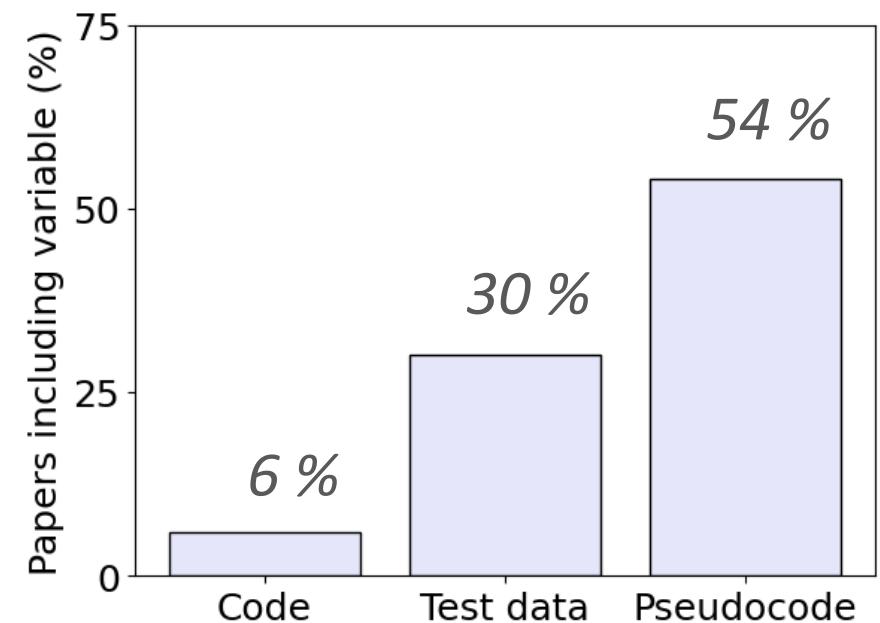
“The growth of machine learning and making it FAIR”

Nat. Chem. **13**, 505–508 (2021).



“Artificial intelligence faces reproducibility crisis”

Science **359**, 725-726(2018).



Protocols

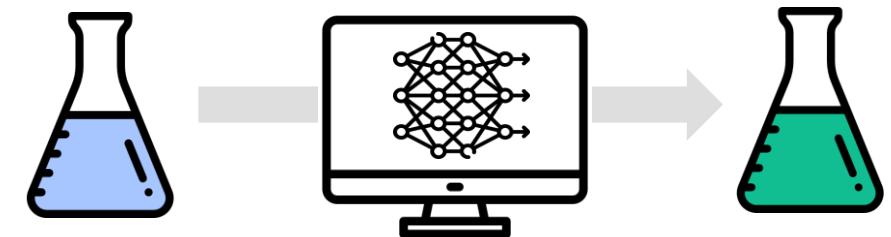


Standards

Why should I automate ML protocols?

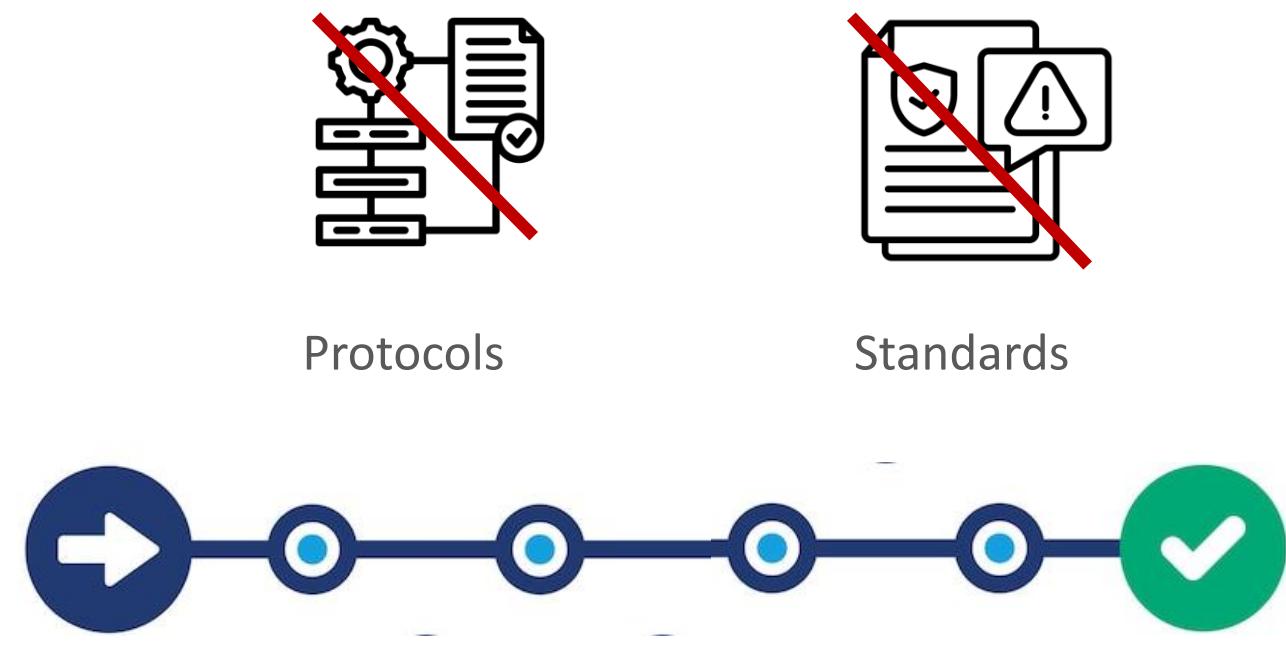
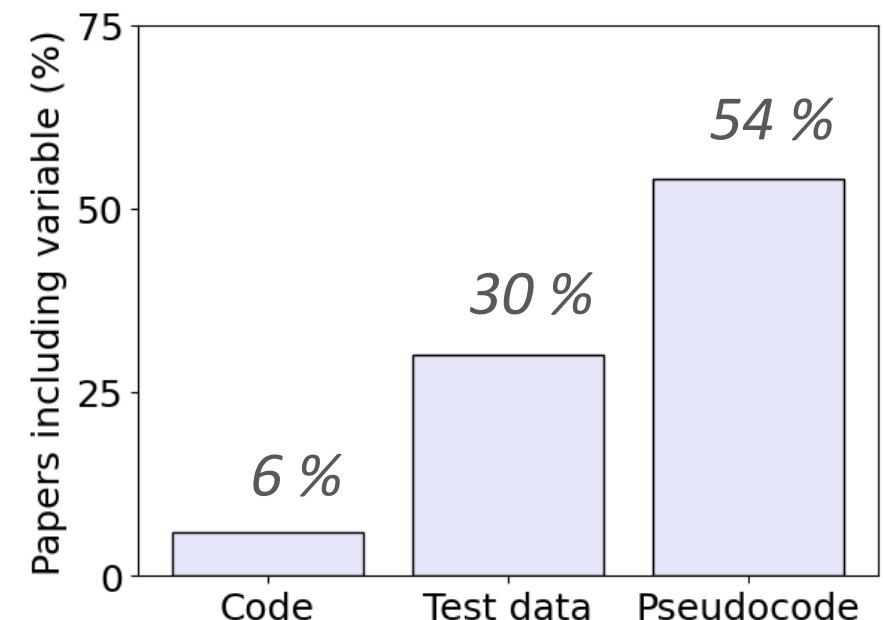
“The growth of machine learning and making it FAIR”

Nat. Chem. **13**, 505–508 (2021).

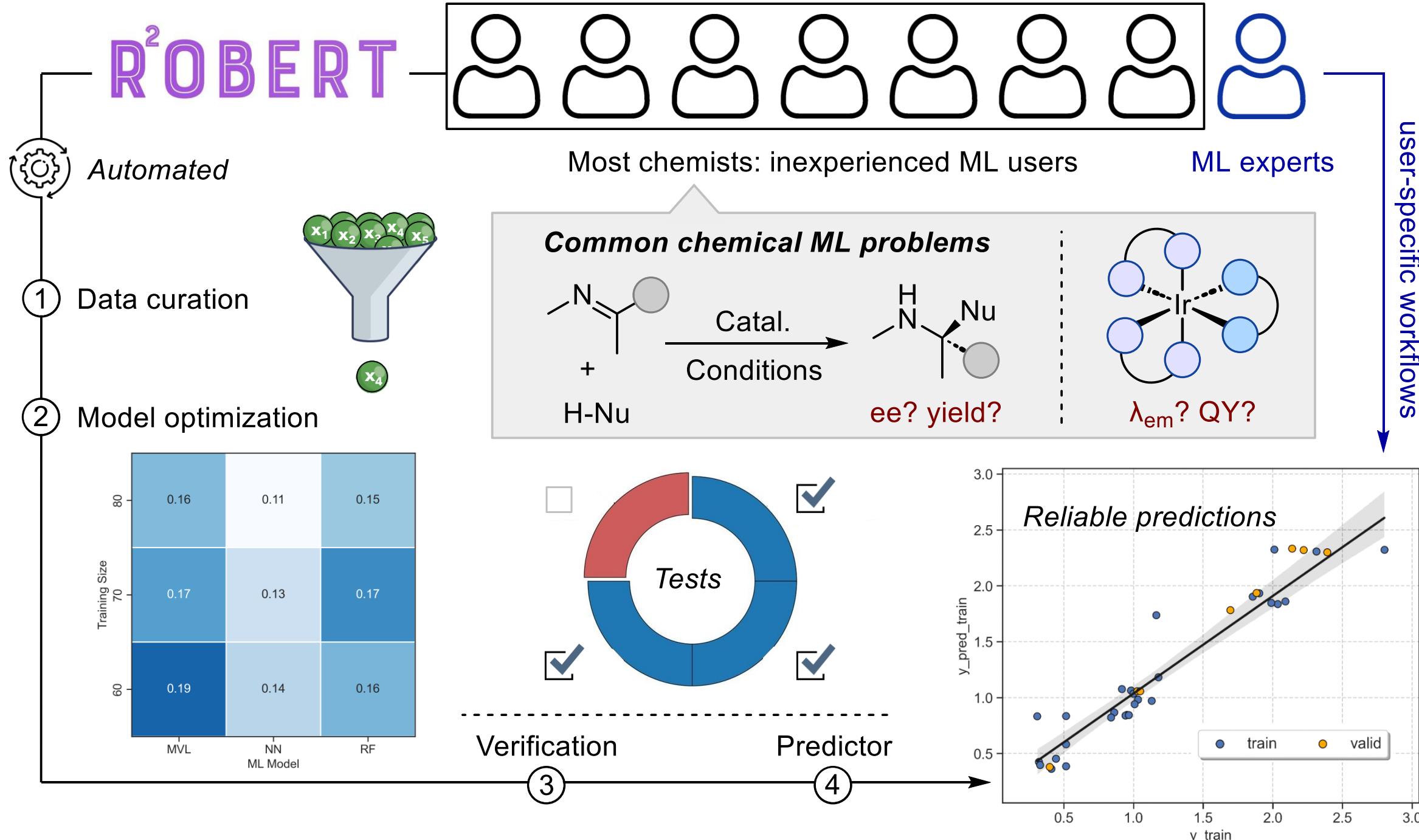


“Artificial intelligence faces reproducibility crisis”

Science **359**, 725-726(2018).



Introduction



Introduction

	Proton Number	First	Second	Third	Fourth
H	1	1310	-	-	-
He	2	2370	5250	-	-
Li	3	519	7300	11800	-
Be	4	900	1760	14800	21000
B	5	799	2420	3660	25000
C	6	1090	2350	4610	6220
N	7	1400	2860	4590	7480
O	8	1310	3390	5320	7450
F	9	1680	3370	6040	8410
Ne	10	2080	3950	6150	9290
Na	11	494	4560	6940	9540
Mg	12	736	1450	7740	10500
Al	13	577	1820	2740	11600
Si	14	786	1580	3230	4360
P	15	1060	1900	2920	4960
S	16	1000	2260	3390	4540
Cl	17	1260	2300	3850	5150
Ar	18	1520	2660	3950	5770
K	19	418	3070	4600	5860
Ca	20	590	1150	4940	6480
Sc	21	632	1240	2390	7110
Ti	22	661	1310	2720	4170
V	23	648	1370	2870	4600
Cr	24	653	1590	2990	4770
Mn	25	716	1510	3250	5190
Fe	26	762	1560	2960	5400
Co	27	757	1640	3230	5100
Ni	28	736	1750	3390	5400
Cu	29	745	1960	3350	5690



Database

Introduction

	Proton Number	First	Second	Third	Fourth
H	1	1310	-	-	-
He	2	2370	5250	-	-
Li	3	519	7300	11800	-
Be	4	900	1760	14800	21000
B	5	799	2420	3660	25000
C	6	1090	2350	4610	6220
N	7	1400	2860	4590	7480
O	8	1310	3390	5320	7450
F	9	1680	3370	6040	8410
Ne	10	2080	3950	6150	9290
Na	11	494	4560	6940	9540
Mg	12	736	1450	7740	10500
Al	13	577	1820	2740	11600
Si	14	786	1580	3230	4360
P	15	1060	1900	2920	4960
S	16	1000	2260	3390	4540
Cl	17	1260	2300	3850	5150
Ar	18	1520	2660	3950	5770
K	19	418	3070	4600	5860
Ca	20	590	1150	4940	6480
Sc	21	632	1240	2390	7110
Ti	22	661	1310	2720	4170
V	23	648	1370	2870	4600
Cr	24	653	1590	2990	4770
Mn	25	716	1510	3250	5190
Fe	26	762	1560	2960	5400
Co	27	757	1640	3230	5100
Ni	28	736	1750	3390	5400
Cu	29	745	1960	3350	5690

Database



ML model selection



Support Vector Machine
Decision Tree
Ensemble Trees
Generalized Additive Model
Neural Network
Linear Regression
Gaussian Process Regression



Introduction

	Proton Number	First	Second	Third	Fourth
H	1	1310	-	-	-
He	2	2370	5250	-	-
Li	3	519	7300	11800	-
Be	4	900	1760	14800	21000
B	5	799	2420	3660	25000
C	6	1090	2350	4610	6220
N	7	1400	2860	4590	7480
O	8	1310	3390	5320	7450
F	9	1680	3370	6040	8410
Ne	10	2080	3950	6150	9290
Na	11	494	4560	6940	9540
Mg	12	736	1450	7740	10500
Al	13	577	1820	2740	11600
Si	14	786	1580	3230	4360
P	15	1060	1900	2920	4960
S	16	1000	2260	3390	4540
Cl	17	1260	2300	3850	5150
Ar	18	1520	2660	3950	5770
K	19	418	3070	4600	5860
Ca	20	590	1150	4940	6480
Sc	21	632	1240	2390	7110
Ti	22	661	1310	2720	4170
V	23	648	1370	2870	4600
Cr	24	653	1590	2990	4770
Mn	25	716	1510	3250	5190
Fe	26	762	1560	2960	5400
Co	27	757	1640	3230	5100
Ni	28	736	1750	3390	5400
Cu	29	745	1960	3350	5690

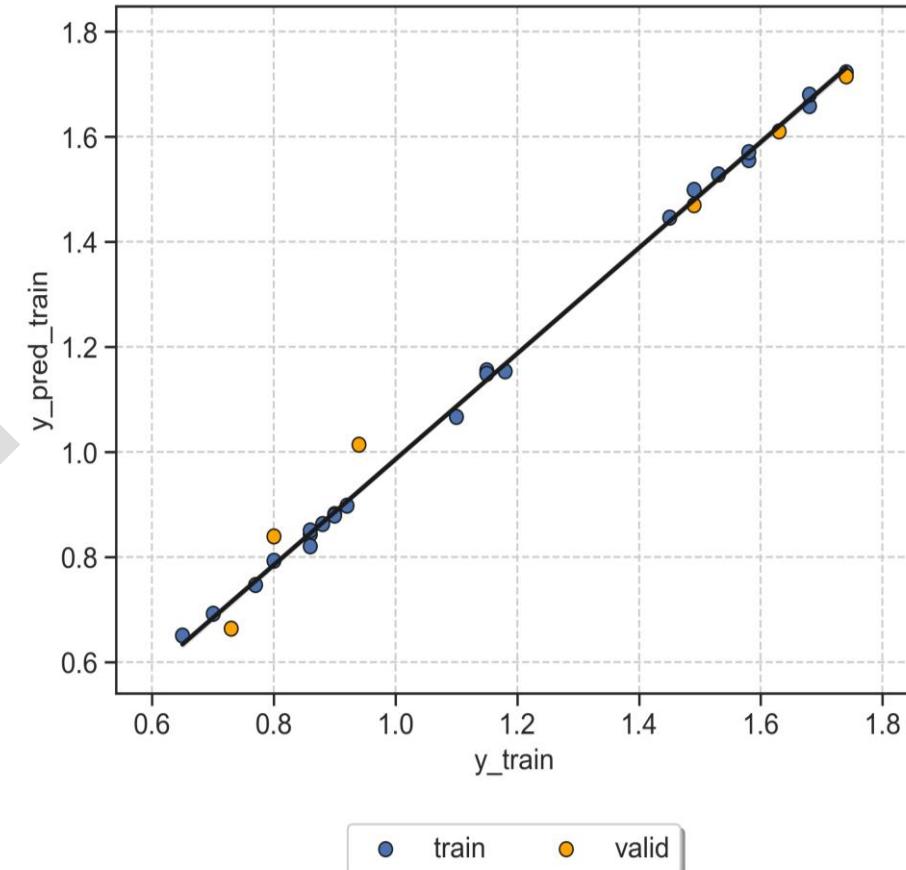
Database



ML model selection



Support Vector Machine
Decision Tree
Ensemble Trees
Generalized Additive Model
Neural Network
Linear Regression
Gaussian Process Regression



Results

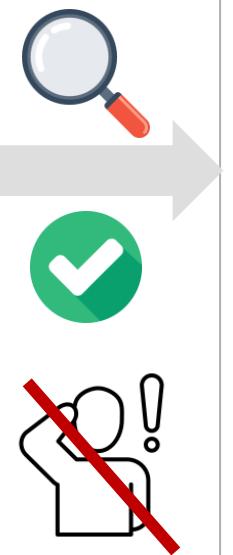
Introduction

	Proton Number	First	Second	Third	Fourth
H	1	1310	-	-	-
He	2	2370	5250	-	-
Li	3	519	7300	11800	-
Be	4	900	1760	14800	21000
B	5	799	2420	3660	25000
C	6	1090	2350	4610	6220
N	7	1400	2860	4590	7480
O	8	1310	3390	5320	7450
F	9	1680	3370	6040	8410
Ne	10	2080	3950	6150	9290
Na	11	494	4560	6940	9540
Mg	12	736	1450	7740	10500
Al	13	577	1820	2740	11600
Si	14	786	1580	3230	4360
P	15	1060	1900	2920	4960
S	16	1000	2260	3390	4540
Cl	17	1260	2300	3850	5150
Ar	18	1520	2660	3950	5770
K	19	418	3070	4600	5860
Ca	20	590	1150	4940	6480
Sc	21	632	1240	2390	7110
Ti	22	661	1310	2720	4170
V	23	648	1370	2870	4600
Cr	24	653	1590	2990	4770
Mn	25	716	1510	3250	5190
Fe	26	762	1560	2960	5400
Co	27	757	1640	3230	5100
Ni	28	736	1750	3390	5400
Cu	29	745	1960	3350	5690

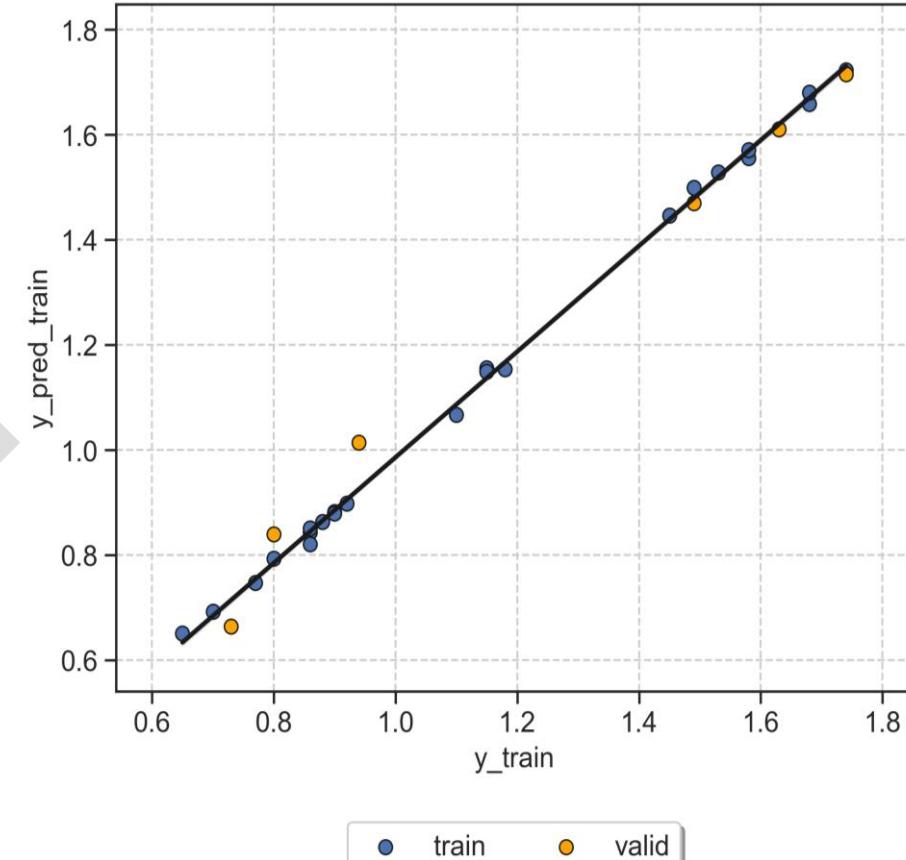
Database



ML model selection



Support Vector Machine
Decision Tree
Ensemble Trees
Generalized Additive Model
Neural Network
Linear Regression
Gaussian Process Regression



Results

Installation (instructions in Read the Docs)

- **Windows, macOS and Linux:**

1. Download Anaconda

2. Open a terminal with Anaconda

3. Copy/paste commands from Read the Docs

Quick note for users with no Python experience

You need a terminal with Python to install and run ROBERT. These are some suggested first steps:

For Windows users:

1. Install [Anaconda with Python 3](#).
2. Open an Anaconda prompt.
3. Create a conda environment called "robert" with Python (`conda create -n robert python=3.10`).
You only need to do this once.
4. Activate the conda environment called "robert" (`conda activate robert`).
5. Install ROBERT as defined in the "Installation" section (`conda install -c conda-forge robert`).
6. Go to the folder with your CSV database (using the "cd" command, i.e. `cd C:/Users/test_robert`).
7. Run ROBERT as explained in the Examples section.

Installation (instructions in Read the Docs)

- **Windows, macOS and Linux:**

1. Download Anaconda

2. Open a terminal with Anaconda

3. Copy/paste commands from Read the Docs



- **Nine testers with varying Python level:** *between 30 s - 2 min*

* *No Python experience at all: install Anaconda +5 min*

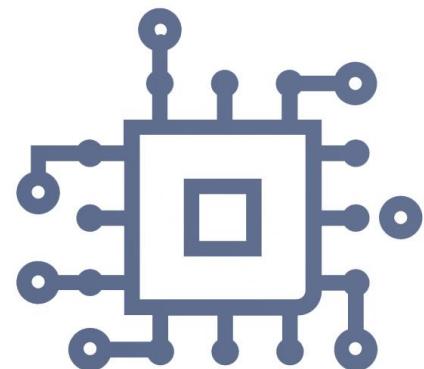
Quick note for users with no Python experience

You need a terminal with Python to install and run ROBERT. These are some suggested first steps:

For Windows users:

1. Install [Anaconda with Python 3](#).
2. Open an Anaconda prompt.
3. Create a conda environment called "robert" with Python (`conda create -n robert python=3.10`).
You only need to do this once.
4. Activate the conda environment called "robert" (`conda activate robert`).
5. Install ROBERT as defined in the "Installation" section (`conda install -c conda-forge robert`).
6. Go to the folder with your CSV database (using the "cd" command, i.e. `cd C:/Users/test_robert`).
7. Run ROBERT as explained in the Examples section.

ChemRxiv®

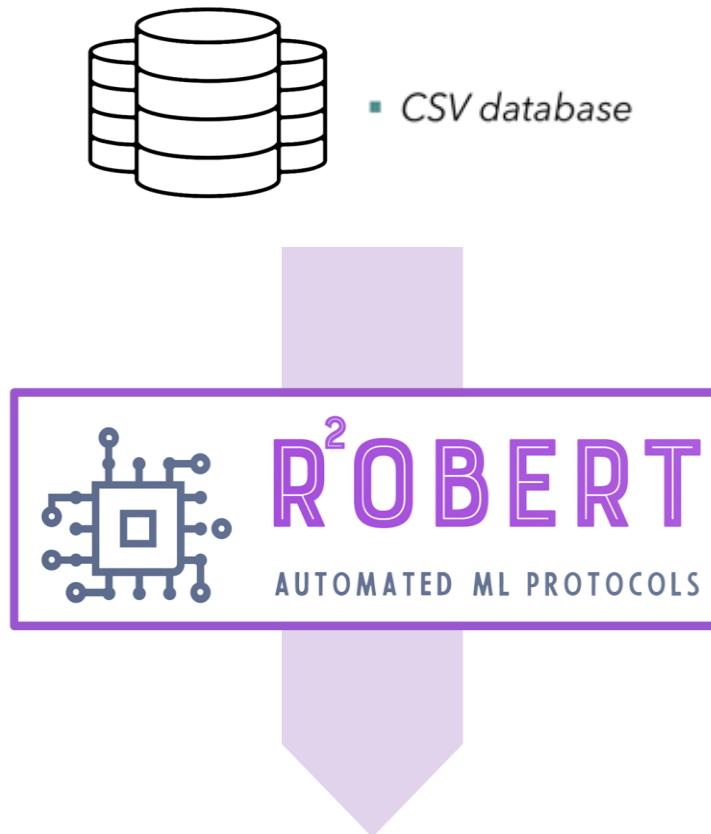


R²OBERT
AUTOMATED ML PROTOCOLS

Refiner and Optimizer of a Bunch of Existing Regression Tools

Submitted for publication

ROBERT: automation of ML protocols



- Data curation ML model scan
- Statistical tests Predictor model
- Outlier analysis AQME tandem

Input: CSV file with database or SMILES

User-defined descriptors

	X _i		y	
code_name	charge	LUMO	...	solub.
mol_5	0	-6,581	...	-13,3
mol_13	0	-5,375	...	-2,68
mol_16	0	-4,349	...	-1,41
mol_34	0	-6,557	...	-1,64

Automated: SMILES to descriptors

SMILES	solub.
c1ccsc1	-13,3
CCCC=C	-2,68
CC(C)Cl	-1,41
ClC(=C)Cl	-1,64

or

- Charge
 - Chi_n
 - FOD
 - V_{bur}
 - ...
- 200+ descriptors
(RDKit, xTB & DBSTEP)

Executing ROBERT: only one command line needed

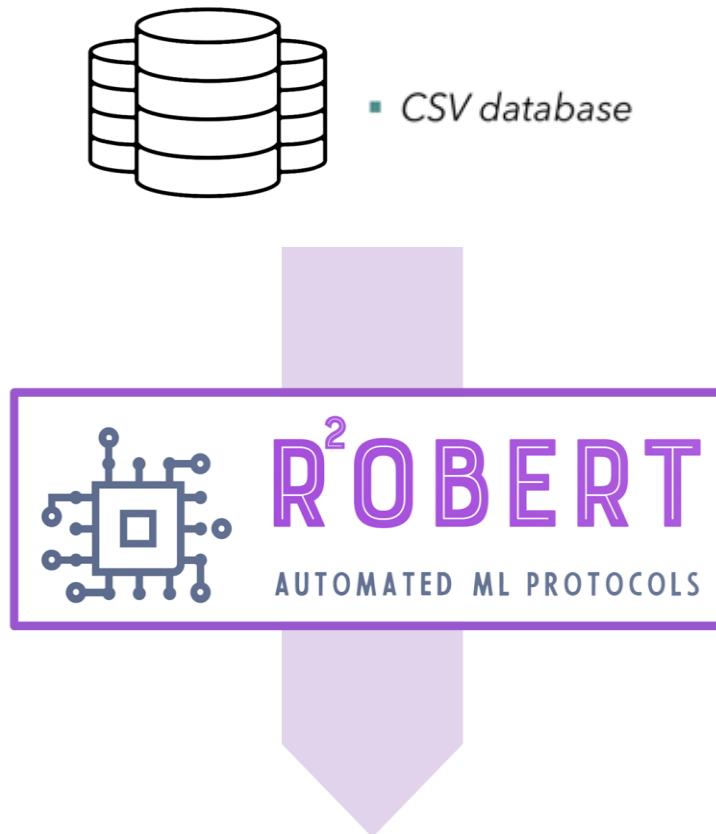


python -m robert --csv_name FILENAME.csv --y y_NAME (--ARG1 --ARG2 ...)



- ① Data curation
- ② Model optimization
- ③ Verification
- ④ Predictor

ROBERT: automation of ML protocols



- Data curation ML model scan
- Statistical tests Predictor model
- Outlier analysis AQME tandem

Input: CSV file with database or SMILES

User-defined descriptors

		X _i		y
code_name	charge	LUMO	...	solub.
mol_5	0	-6,581	...	-13,3
mol_13	0	-5,375	...	-2,68
mol_16	0	-4,349	...	-1,41
mol_34	0	-6,557	...	-1,64

Automated: SMILES to descriptors

SMILES	y
c1ccsc1	-13,3
CCCC=C	-2,68
CC(C)Cl	-1,41
CIC(=C)Cl	-1,64

or

Charge	FOD
Chi _n	V _{bur}
...	...
200+ descriptors	(RDKit, xTB & DBSTEP)

Executing ROBERT: only one command line needed

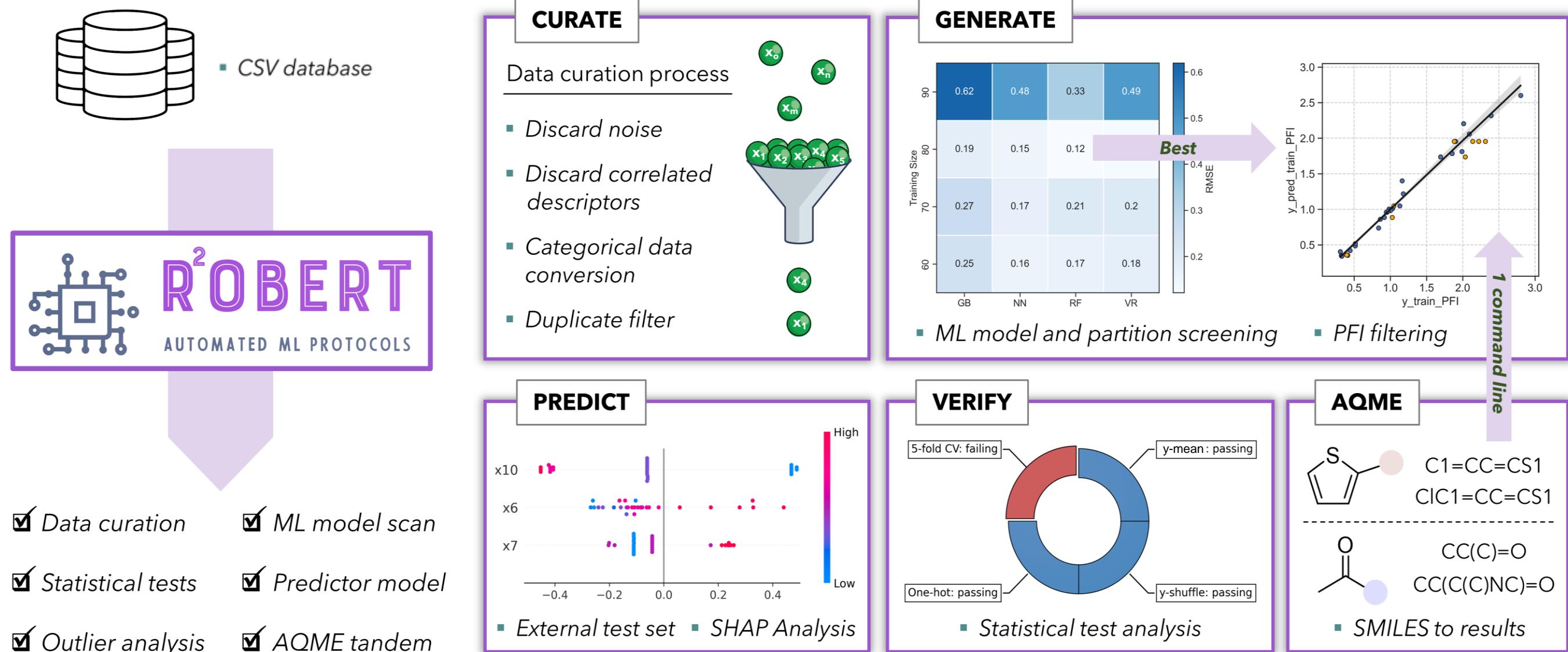


python -m robert --csv_name FILENAME.csv --y y_NAME (--ARG1 --ARG2 ...)



- ① Data curation
- ② Model optimization
- ③ Verification
- ④ Predictor

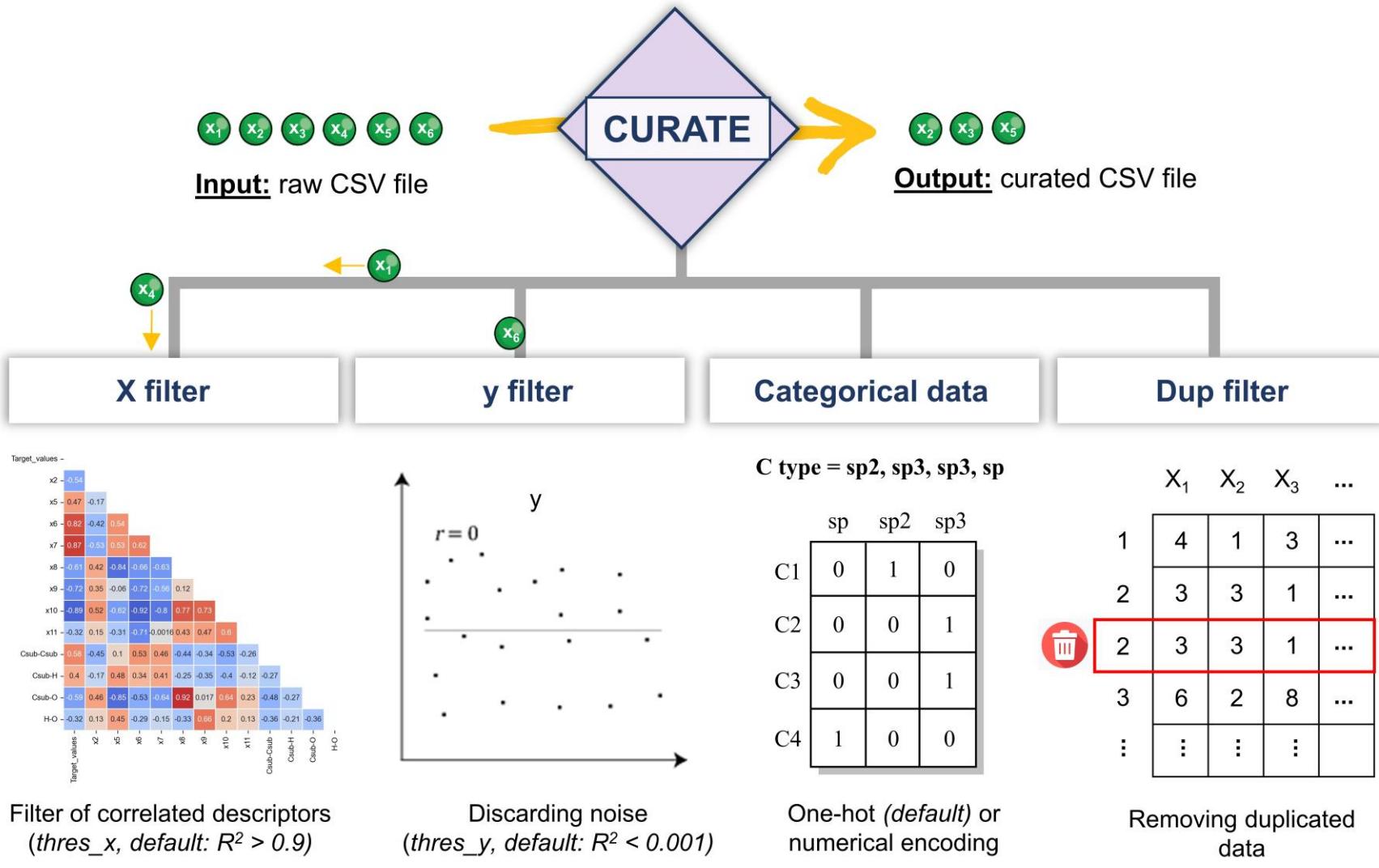
ROBERT: automation of ML protocols



Full workflow in one command line

CURATE

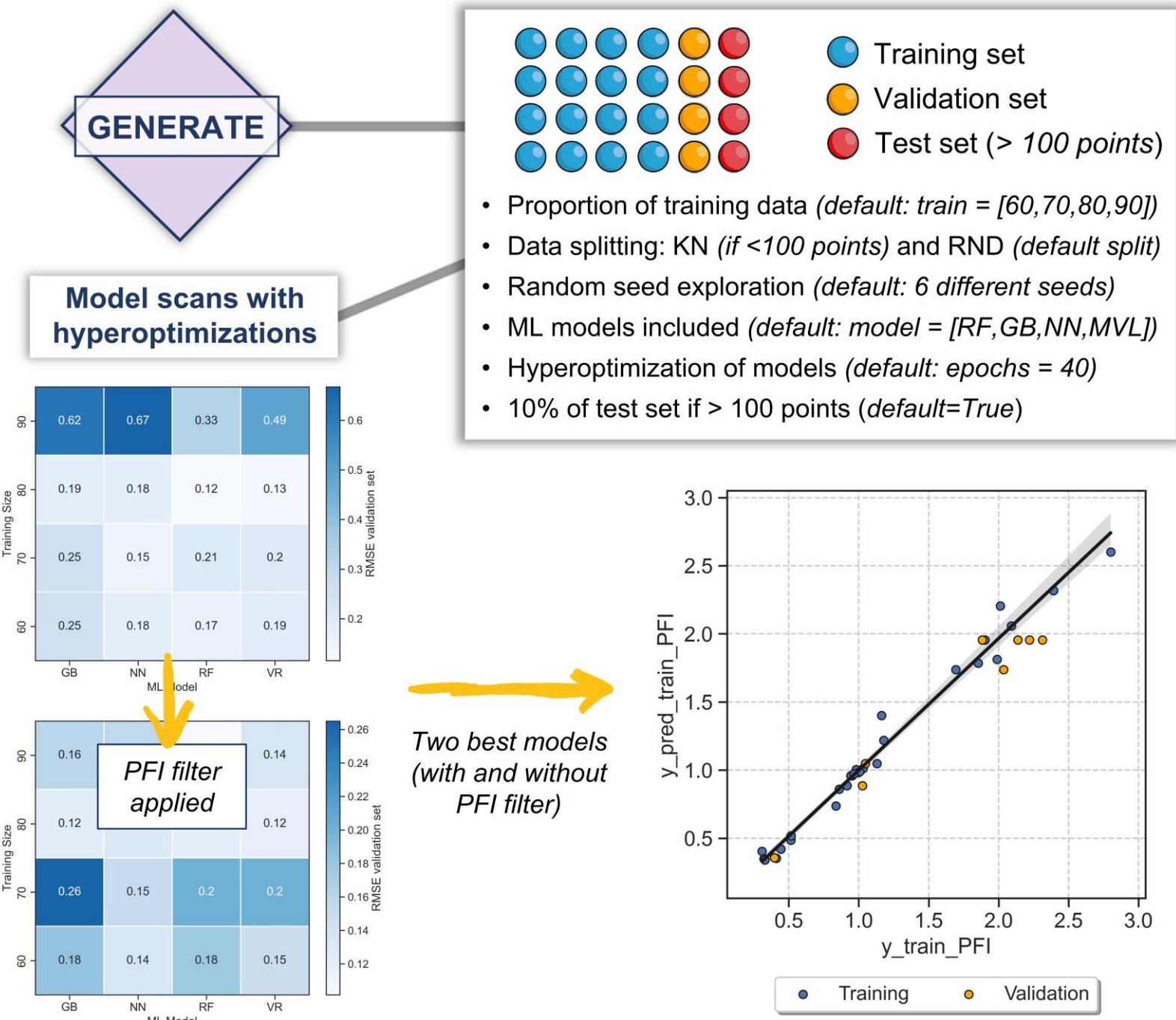
1. Filters off correlated descriptors
2. Filters off variables with very low correlation to the target values (**noise**)
3. Converts **categorical descriptors** into one-hot descriptors
4. Filters off duplicates
5. Filters off **missing values** (replaces using 0s, if >30% discard descriptor)



ROBERT: automation of ML protocols

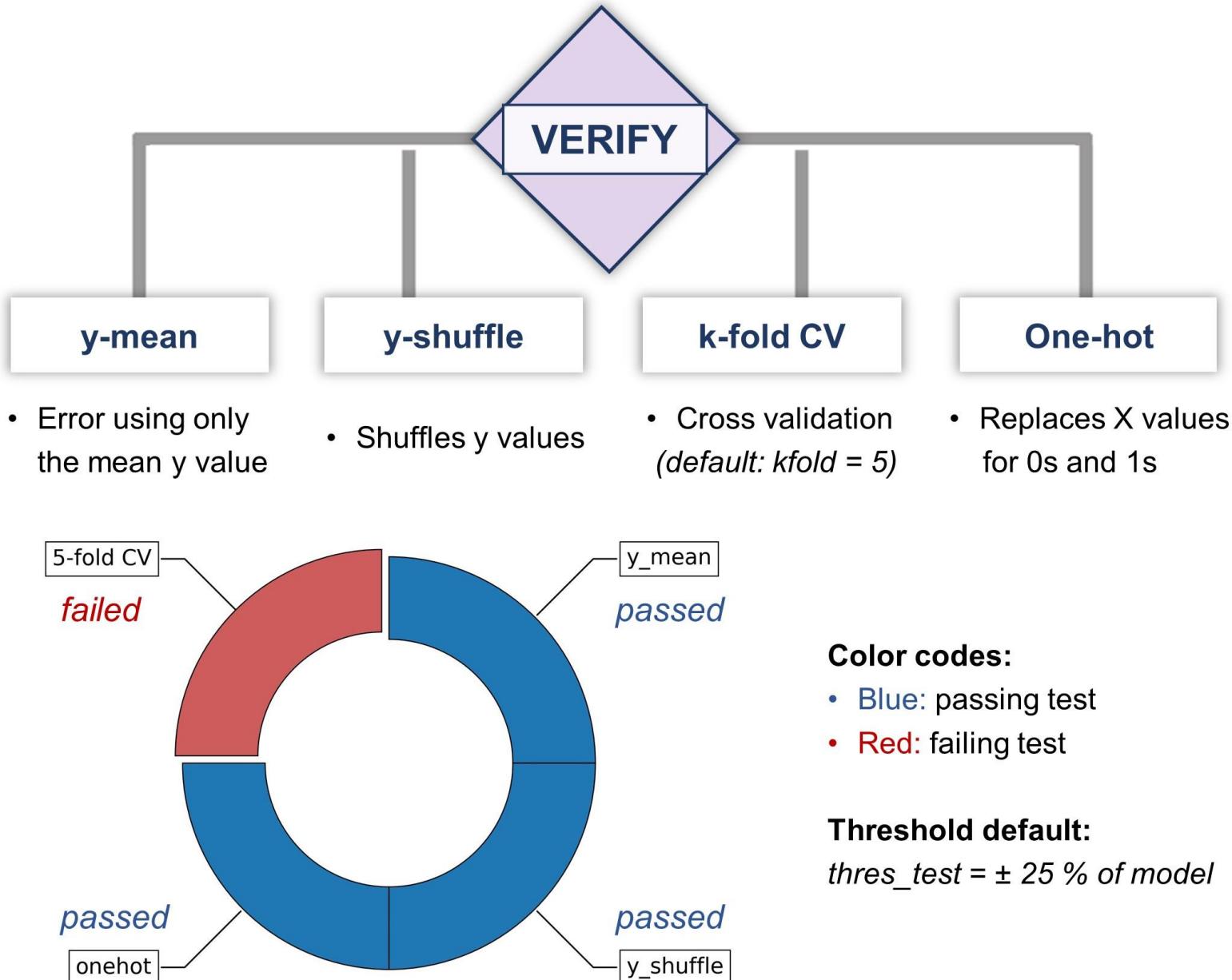
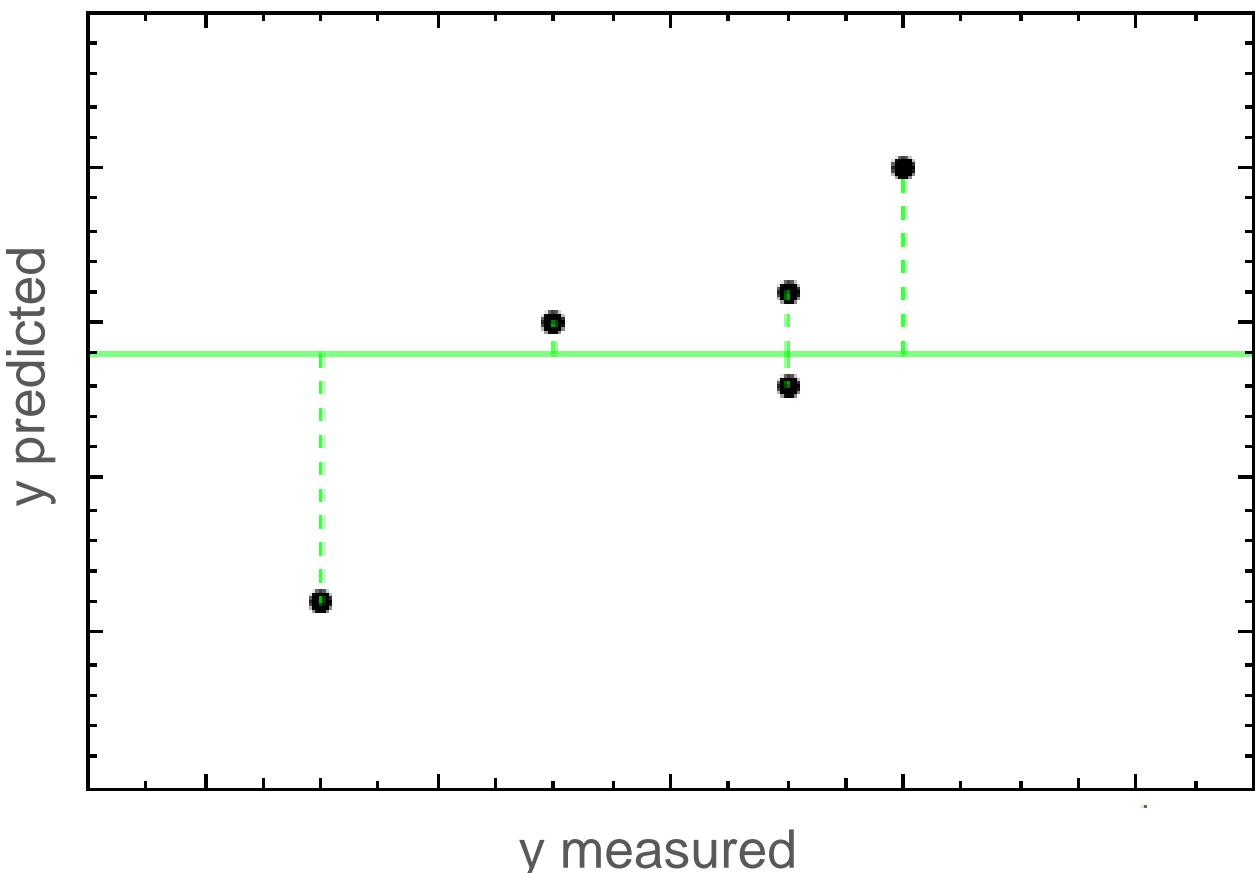
GENERATE

1. Hyperoptimizes multiple ML models
2. Filters off descriptors with low permutance feature importance (PFI).
3. Creates a heatmap plot with different model types and partition sizes.
4. Selects and stores the **best** No PFI and PFI models



VERIFY

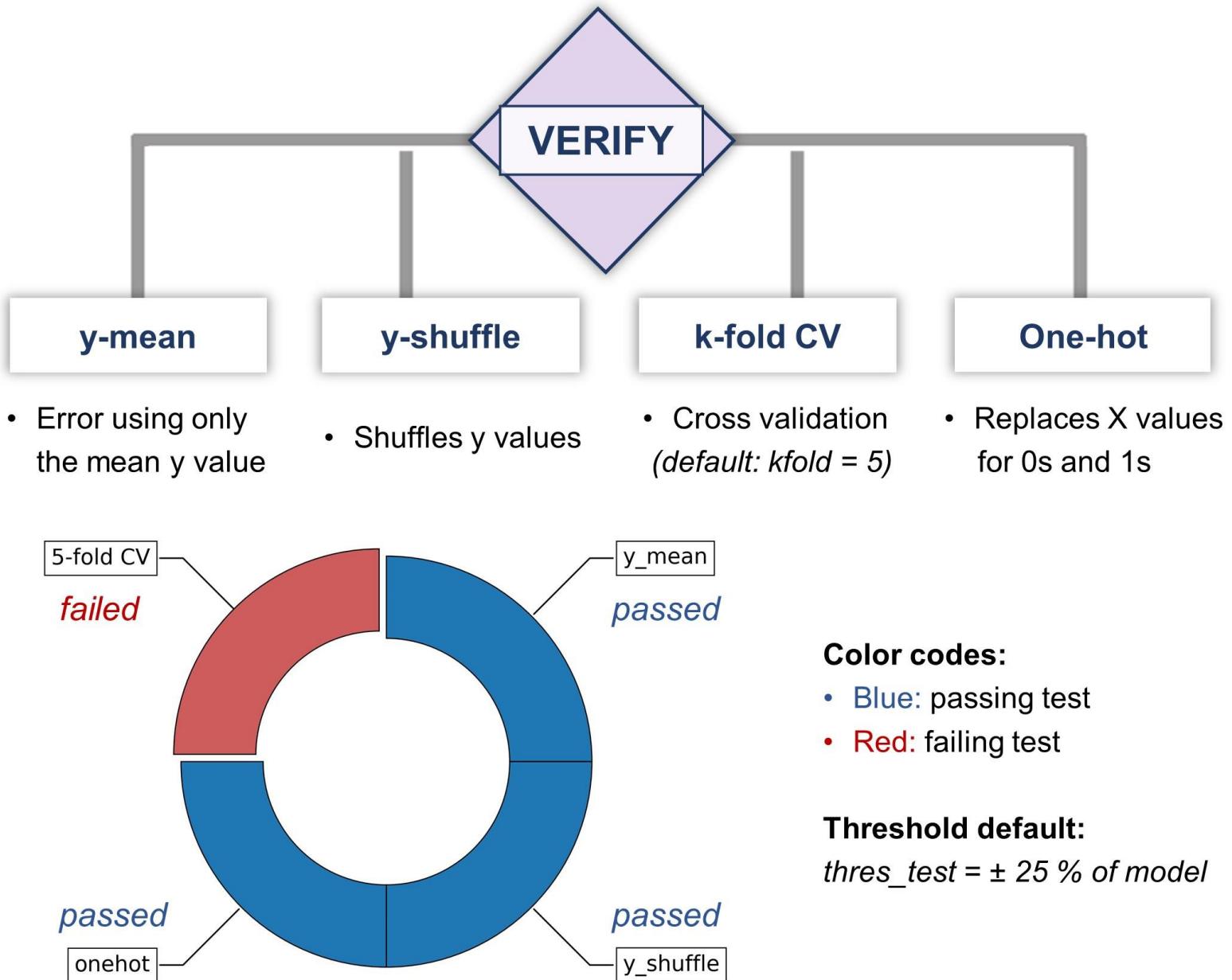
1. **y-mean test:** Calculates the accuracy of the model when all the predicted y values are fixed to the mean of the y values



VERIFY

2. **y-shuffle test:** Calculates the accuracy of the model after shuffling randomly all the measured y values

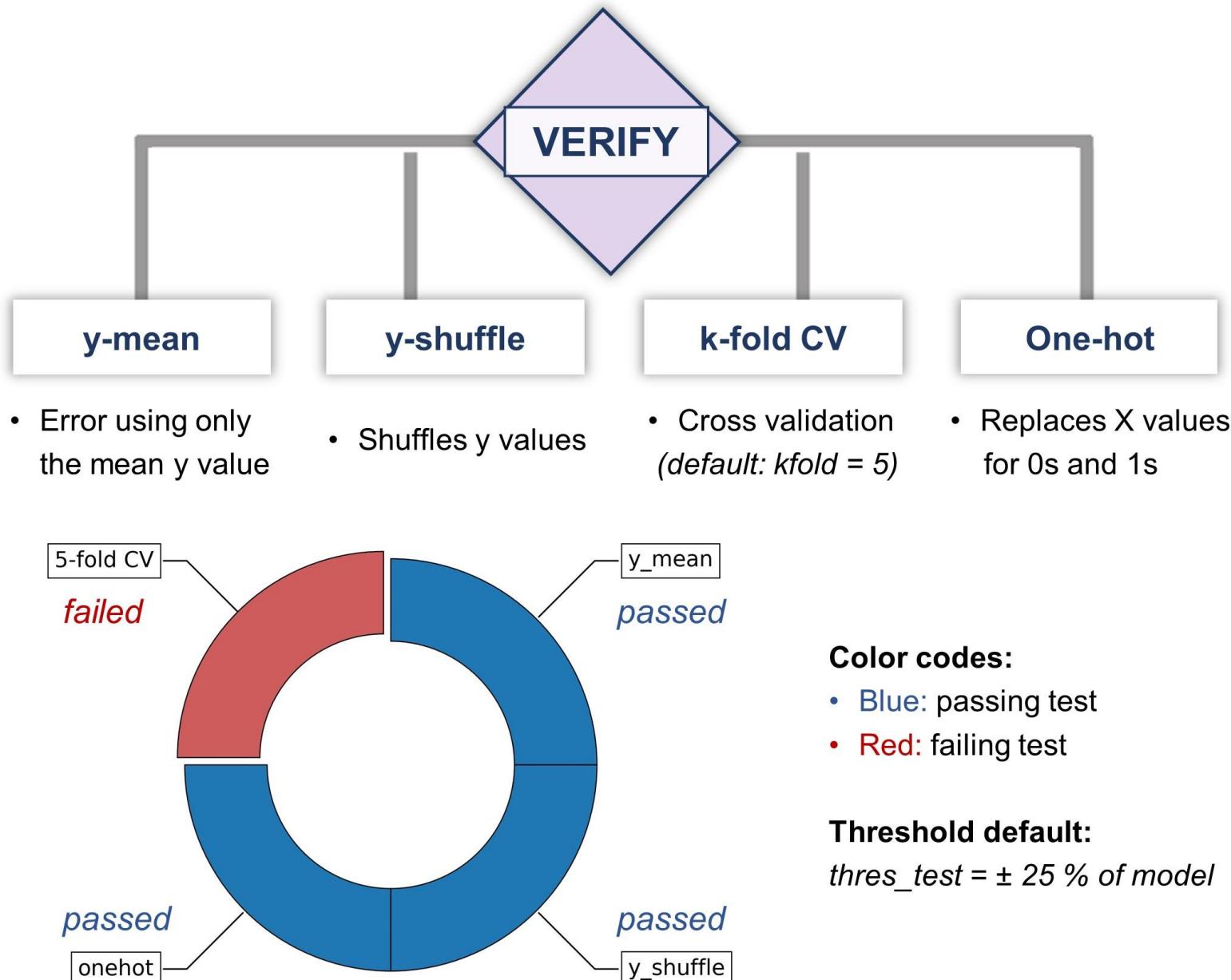
Name	X1	y	y-shuffle
A	4,1	15	22
B	5,3	10	5
C	-2,4	5	10
D	15,1	15	2
E	-7,8	22	15
F	-0,2	2	15



VERIFY

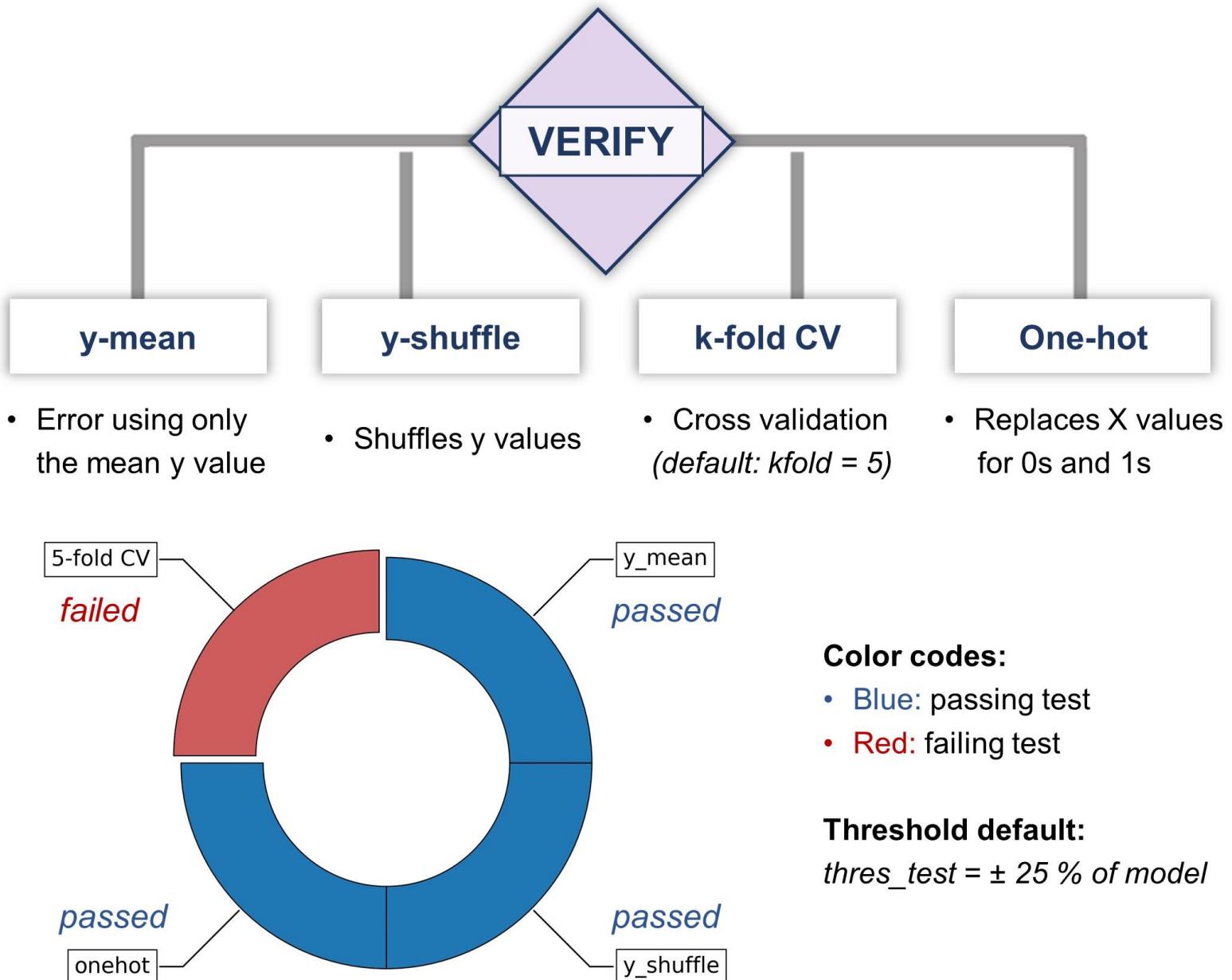
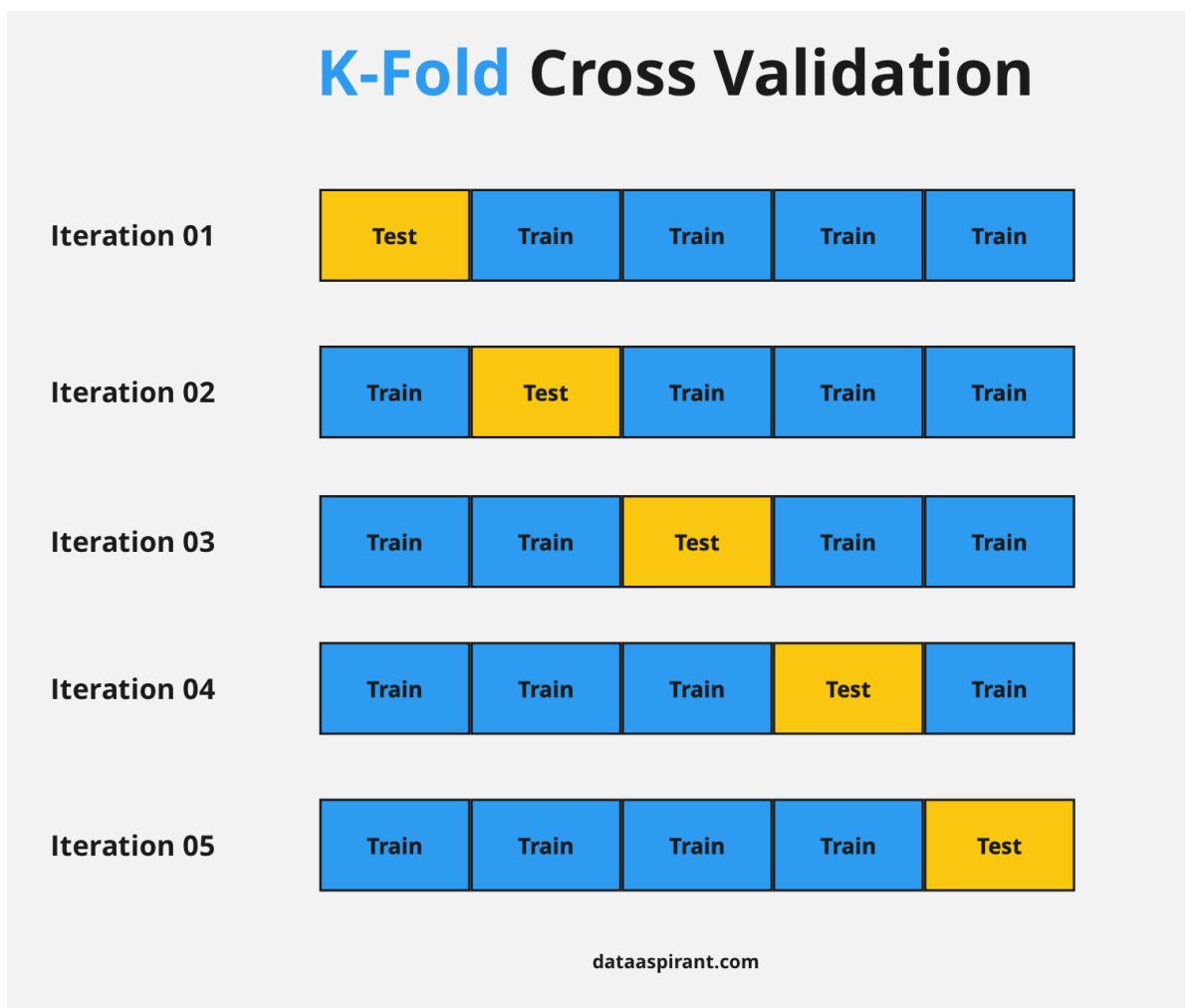
4. **One-hot test:** Calculates the accuracy of the model when replacing all descriptors for 0s and 1s

Name	X1	X1-OH	y
A	4,1	1	15
B	5,3	1	10
C	0,0	0	5
D	15,1	1	15
E	0,0	0	22
F	0,0	0	2



VERIFY

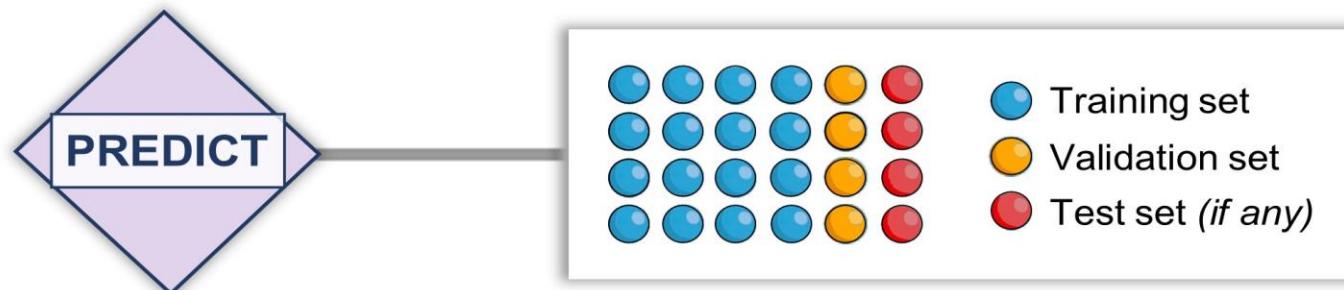
3. **k-fold CV test:** Calculates the accuracy of the model with a k-fold cross-validation



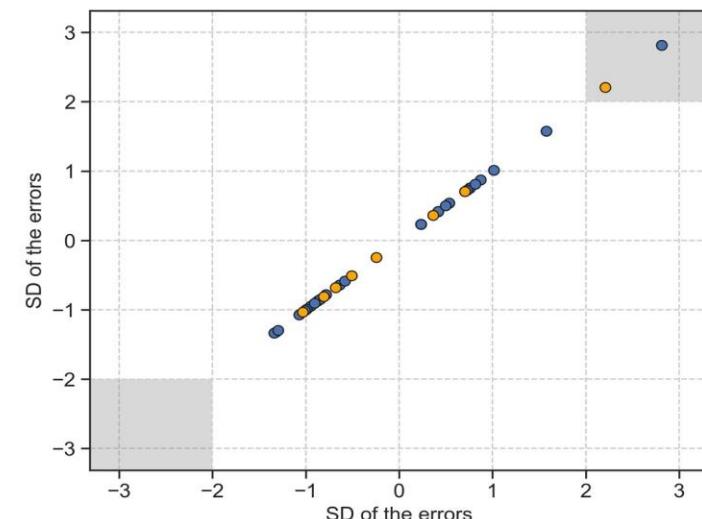
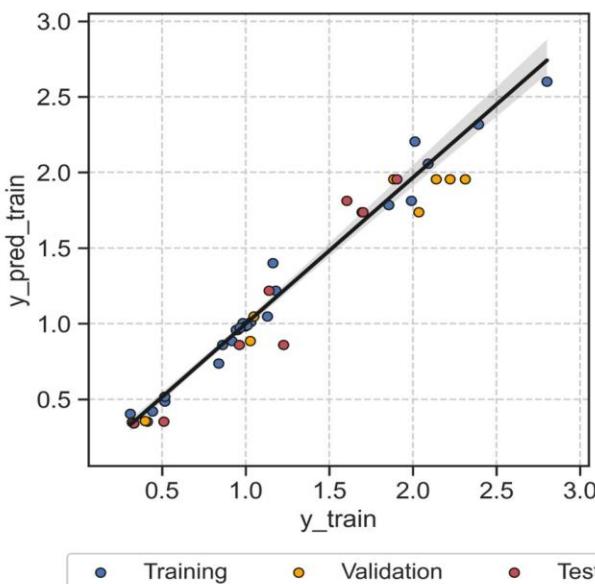
ROBERT: automation of ML protocols

PREDICT

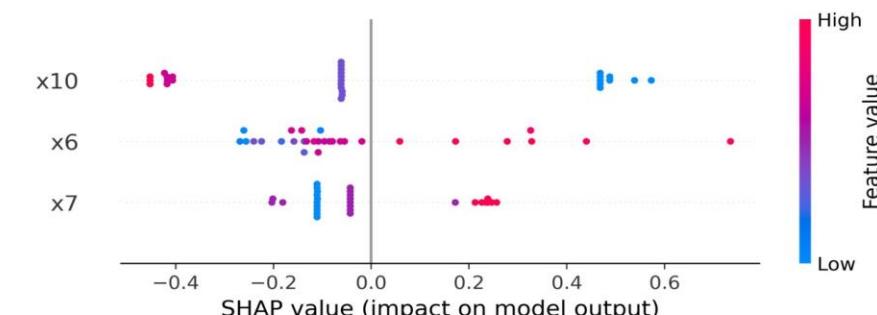
1. Calculates **R2**, **MAE** and **RMSE** (for regression) or **accuracy**, **F1 score** and **MCC** (for classification)
2. Predicts values for an **external test set (if any)**
3. Performs an **outlier analysis**
4. Performs a **SHAP feature analysis**
5. Performs a **PFI feature analysis**



- Training, validation, and test data
- Outlier analysis (default: $t_value = 2$)



- SHAP and PFI analysis



Results

- A series of folders with the names of the modules used will be created in the folder where ROBERT was executed.

 CURATE	25/07/2023 13:47	Carpeta de archivos
 GENERATE	25/07/2023 17:49	Carpeta de archivos
 PREDICT	25/07/2023 11:30	Carpeta de archivos
 VERIFY	25/07/2023 11:30	Carpeta de archivos
 a	16/06/2023 11:32	Archivo de valores... 212 KB
 ROBERT_report	25/07/2023 11:41	Documento Adob... 1.851 KB

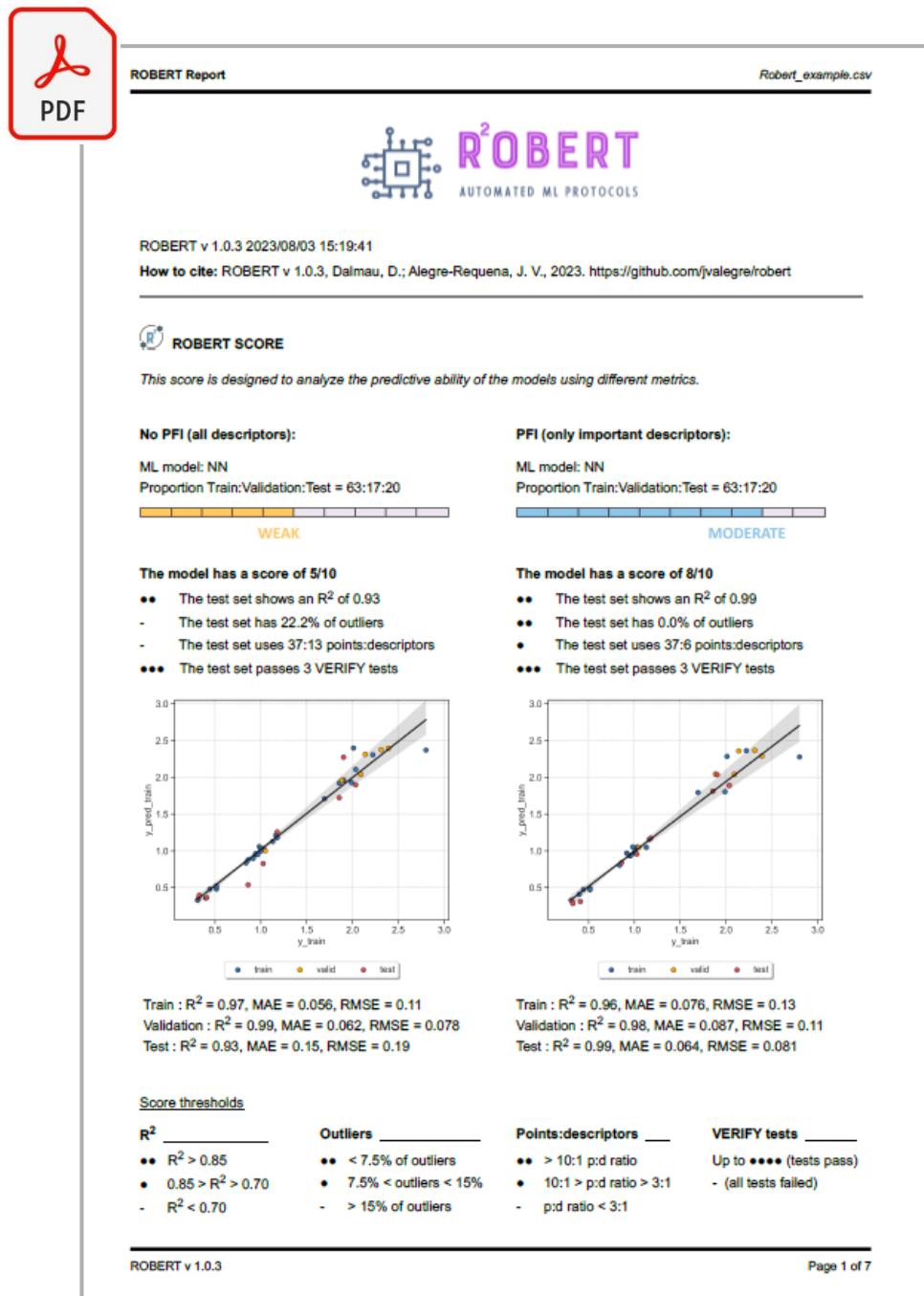
ROBERT: automation of ML protocols

Results

- A series of folders with the names of the modules used will be created in the folder where ROBERT was executed.

 CURATE	25/07/2023 13:47	Carpeta de archivos
 GENERATE	25/07/2023 17:49	Carpeta de archivos
 PREDICT	25/07/2023 11:30	Carpeta de archivos
 VERIFY	25/07/2023 11:30	Carpeta de archivos
 a	16/06/2023 11:32	Archivo de valores...
		212 KB
 ROBERT_report	25/07/2023 11:41	Documento Adob...
		1.851 KB

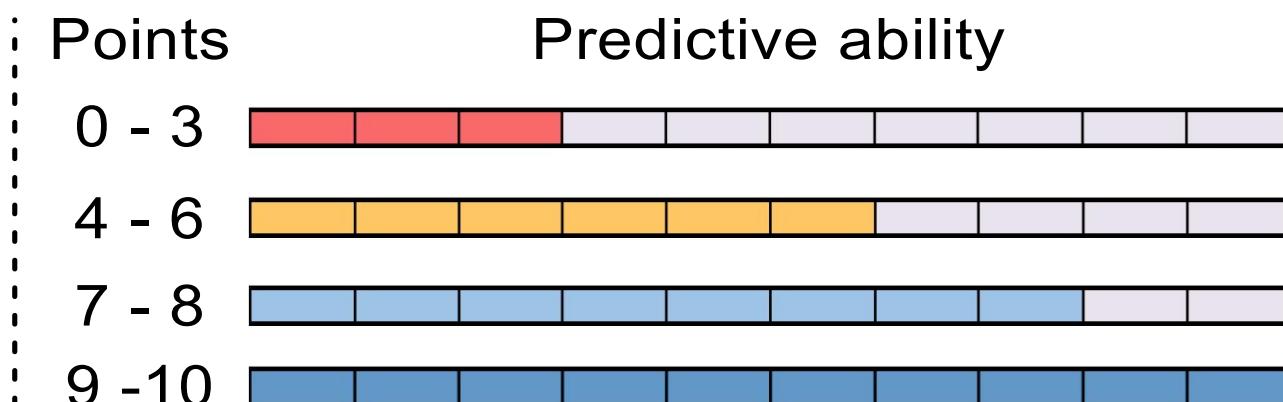
- A PDF file called **ROBERT_report.pdf** will be created with the most important information



Score criteria (points)

Points	R ²	Outliers	Desc:pts
●●	> 0.85	< 7.5%	> 1:10
●	0.70 - 0.85	7.5 - 15%	1:3 - 1:10
-	< 0.70	> 15%	< 1:3

Verify tests (up to ●●●●)	● 5-fold CV	● y-shuffle
	● y-mean	● One-hot



No PFI (all descriptors):



WEAK

Your model has a score of 5/10

- Your model shows an R² of 0.93
- Your model has 22.2% of outliers
- Your model uses 37:13 points:descriptors
- Your model passes 3 VERIFY tests

PFI (only important descriptors):



MODERATE

Your model has a score of 8/10

- Your model shows an R² of 0.99
- Your model has 0.0% of outliers
- Your model uses 37:6 points:descriptors
- Your model passes 3 VERIFY tests

ROBERT: automation of ML protocols

How is the score calculated?

Predictive ability towards an external test set (2 points):

Points	Condition
•• 2	$R^2 > 0.85$ (high correlation between predicted and measured y values)
•1	$0.85 > R^2 > 0.70$ (moderate correlation)
0	$R^2 < 0.70$ (low correlation)

No PFI (all descriptors):



WEAK

Number of outliers in the validation set (only for regression, 2 points):

Points	Condition
•• 2	Outliers < 7.5% (close to a normal distribution of errors in the valid. set)
•1	7.5% < outliers < 15% (not that far from a normal distribution of errors)
0	Outliers > 15% (far from a normal distribution of errors)

Proportion of datapoints vs descriptors (2 points):

Points	Condition
•• 2	Datapoints:descriptors ratio > 10:1 (reasonably low amount of descriptors)
•1	10:1 > ratio > 3:1 (moderate amount)
0	Ratio < 3:1 (too many descriptors)

Passing VERIFY tests (4 points):

Points	Condition
•1	Each of the VERIFY tests passed (up to •••• 4 points)

Extra points for VERIFY tests (only for classification, 2 points):

Points	Condition
•1	Each y-mean and y-shuffle tests passed (up to •• 2 points)

ROBERT: automation of ML protocols

Some tips to improve the score

- △ The model uses only 37 datapoints, adding meaningful datapoints might help to improve the model.
- △ One of your models have more than 7.5% of outliers (5% is expected for a normal distribution with the t-value of 2 that ROBERT uses), using a more homogeneous distribution of results might help. For example, avoid using many points with similar y values and only a few points with distant y values.
- △ Adding meaningful descriptors or replacing/deleting the least useful descriptors used might help. Feature importances are gathered in the SHAP and PFI sections of the /PREDICT/PREDICT_data.dat file.

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 stored in the last column of two CSV files called MODEL_SIZE_test(_No)_PFI.csv, which are stored in the PREDICT folder.

No PFI (all descriptors):



WEAK

Your model has a score of 5/10

- Your model shows an R² of 0.93
- Your model has 22.2% of outliers
- Your model uses 37:13 points:descriptors
- Your model passes 3 VERIFY tests

PFI (only important descriptors):



MODERATE

Your model has a score of 8/10

- Your model shows an R² of 0.99
- Your model has 0.0% of outliers
- Your model uses 37:6 points:descriptors
- Your model passes 3 VERIFY tests

Reproducibility and transparency

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!):

- Report with results (ROBERT_report.pdf)
- CSV database (Robert_example.csv)
- External test set (Robert_example_test.csv)

2. Install the following Python modules:

- ROBERT: conda install -c conda-forge robert=1.0.3 (or pip install robert==1.0.3)
- scikit-learn-intelex: pip install scikit-learn-intelex==2023.2.1
- To generate the ROBERT_report.pdf summary, the following libraries might be necessary:

WeasyPrint: pip install weasyprint==59.0

GLib: conda install -c conda-forge glib

Pango: conda install -c conda-forge pango

GTK3: conda install -c conda-forge gtk3

3. Run ROBERT with this command line in the folder with the CSV databases (originally run in Python 3.10.12):

```
python -m robert --csv_test "Robert_example_test.csv" --csv_name "Robert_example.csv" --y "Target_values" --names "Name"
```

4. Provide number and model of processors used to achieve:

Total execution time: 101.98 seconds

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 (all descriptors):

sklearn model: MLPRegressor
random_state: 0
batch_size: 8
hidden_layer_sizes: [16, 16]
learning_rate_init: 0.01
max_iter: 50
validation_fraction: 0.3
alpha: 0.0001
shuffle: True
tol: 0.0001
early_stopping: False
beta_1: 0.9
beta_2: 0.999
epsilon: 1e-08

PFI (only important descriptors):

sklearn model: MLPRegressor
random_state: 0
batch_size: 8
hidden_layer_sizes: [16, 16]
learning_rate_init: 0.01
max_iter: 50
validation_fraction: 0.3
alpha: 0.0001
shuffle: True
tol: 0.0001
early_stopping: False
beta_1: 0.9
beta_2: 0.999
epsilon: 1e-08

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

No PFI (all descriptors):

split: KN
type: reg
error_type: rmse

PFI (only important descriptors):

split: KN
type: reg
error_type: rmse

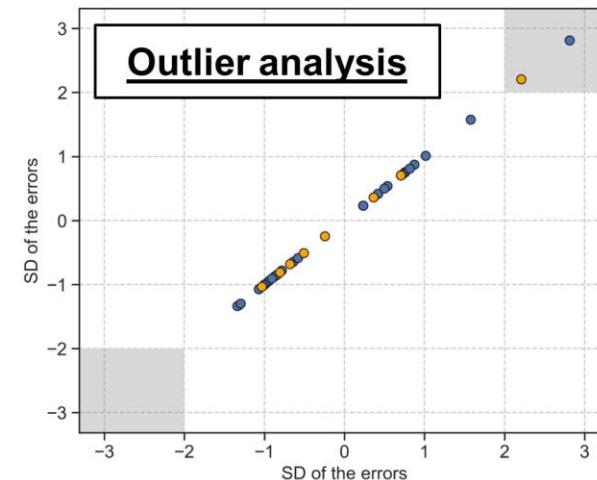
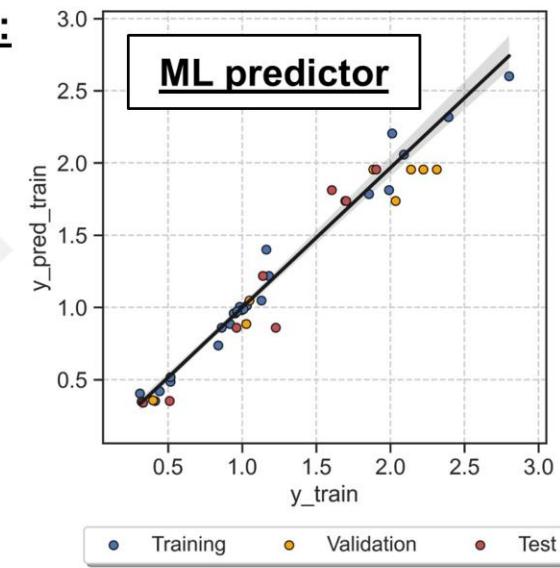
ROBERT: automation of ML protocols

Full workflow from csv

Only one command line required



Outputs:

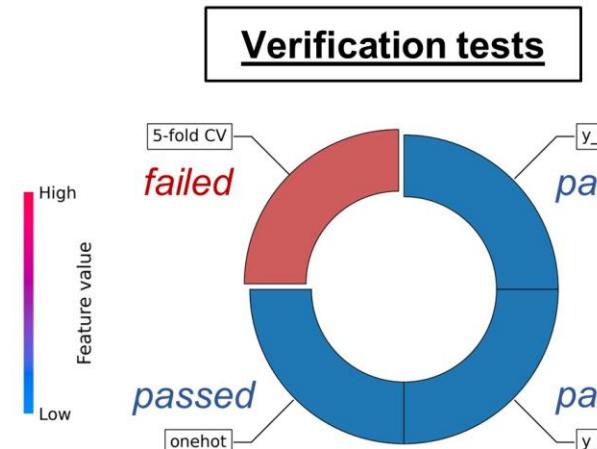
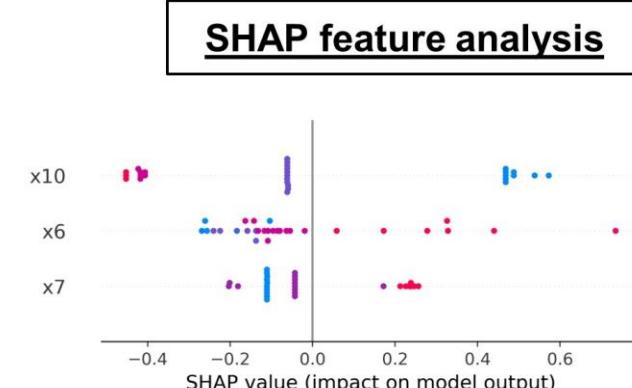


Automated tasks in ROBERT

- Conversion of categorical descriptors
- Data curation of descriptors
- Screening of hyperoptimized ML models
- Screening of partition sizes
- New models with relevant features (PFI)
- Prediction of test set (if any) & SHAP analysis
- Analysis of outliers across different sets
- Verification tests to ensure predictive ability

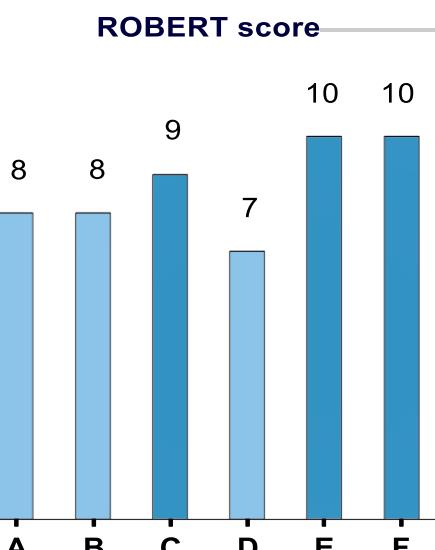
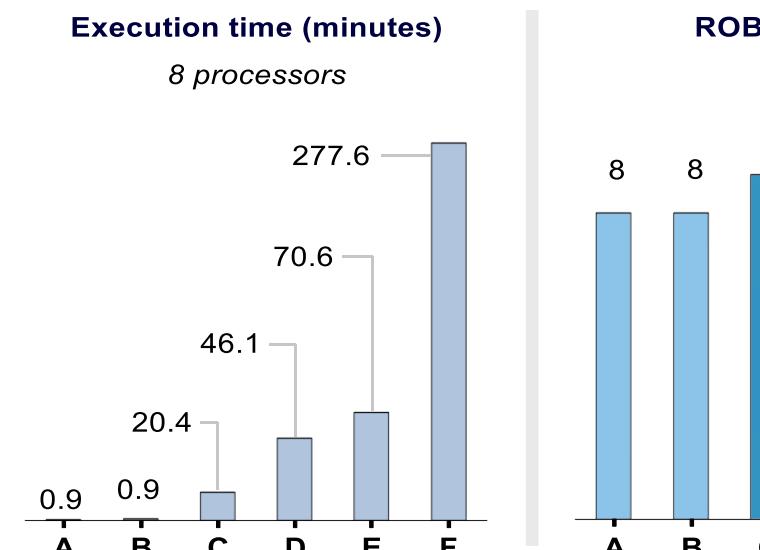
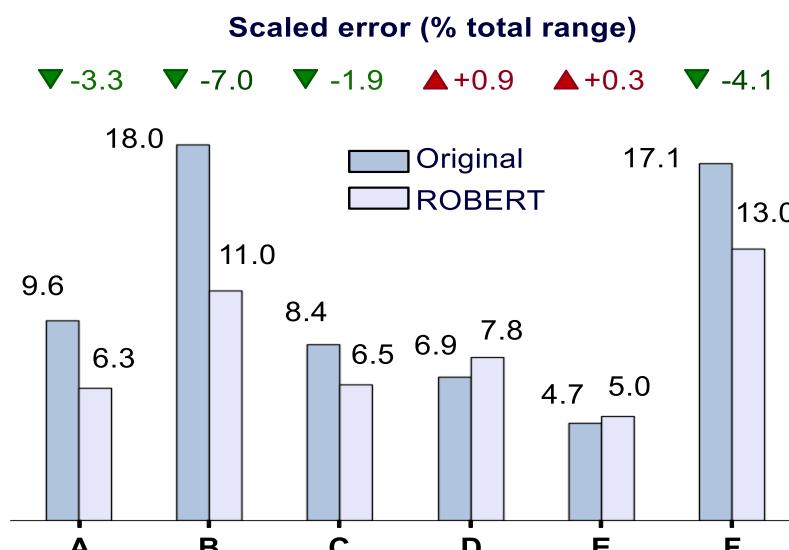
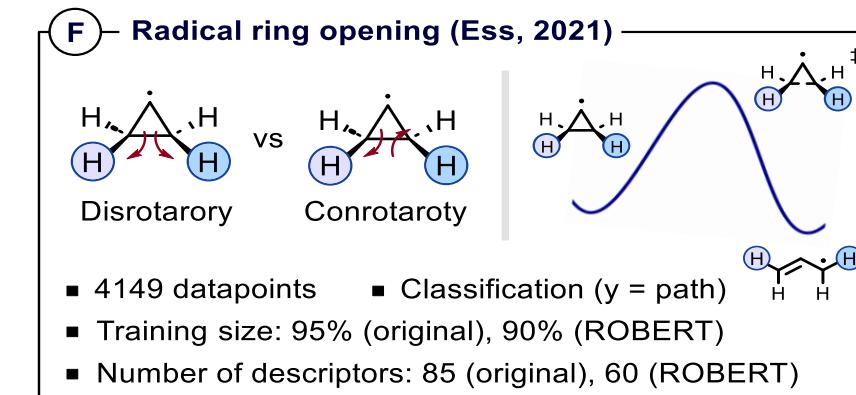
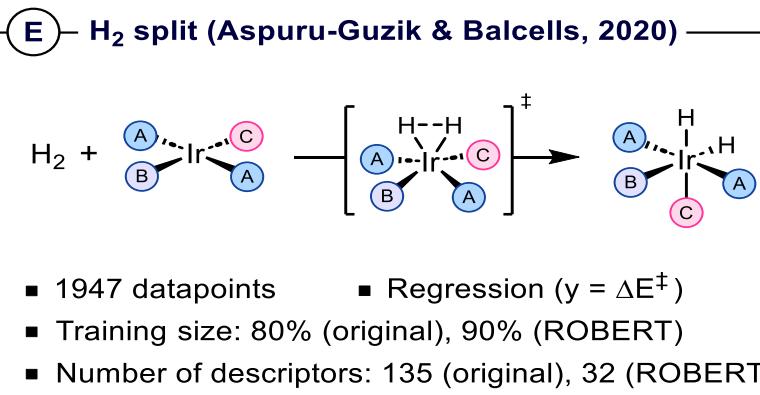
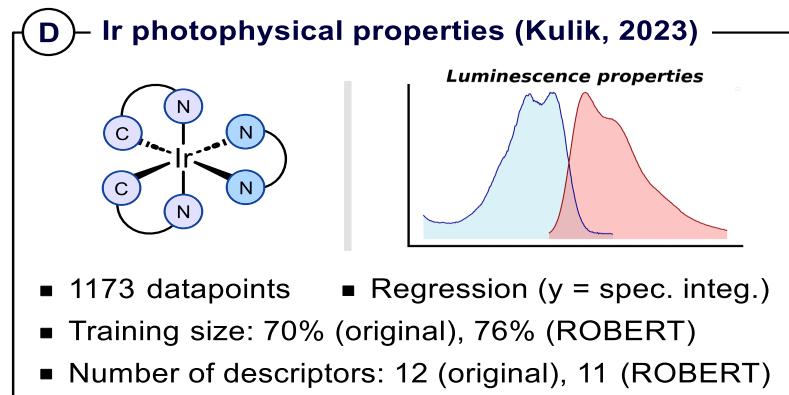
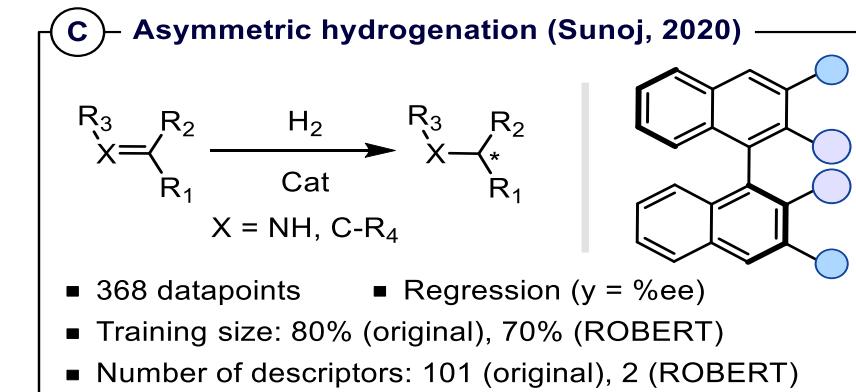
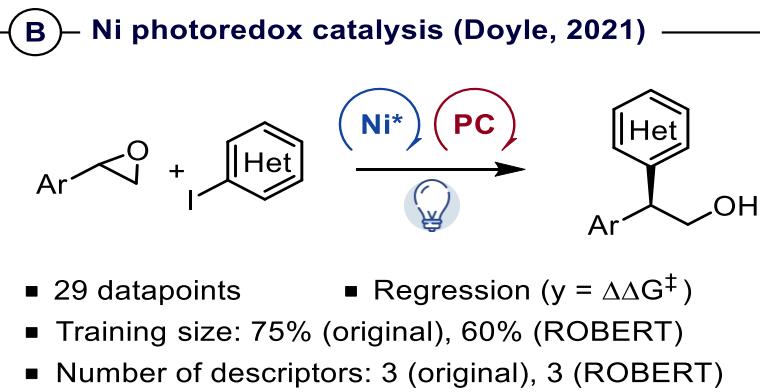
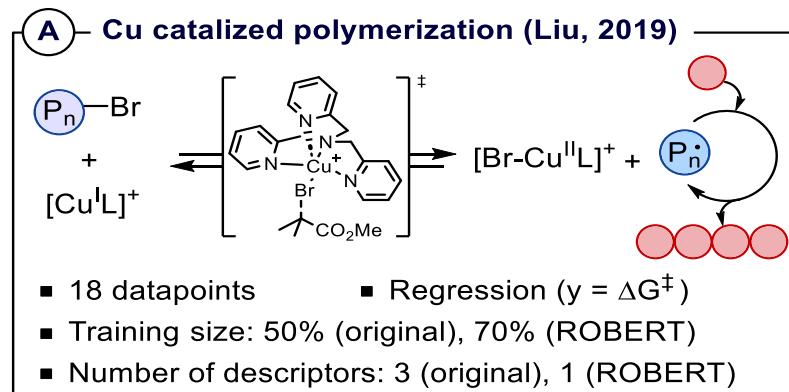
Input: CSV database

Name	y	x1	x2
Point_1	1.85	12	10.9
Point_2	2.03	11.7	10.7
Point_3	0.86	-23.05	12.1
Point_4	1.03	-16.24	12.1
Point_5	0.95	-31.94	11.9
Point_6	0.31	-68.04	13.1
Point_7	0.98	-34.71	10.8



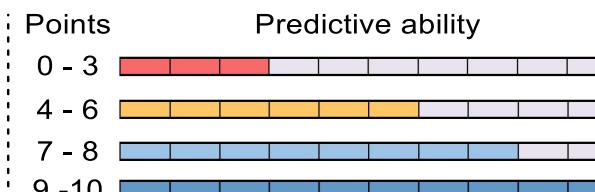
Only one command line: `python -m robert --csv_name name.csv --y target_value`

ROBERT: automation of ML protocols



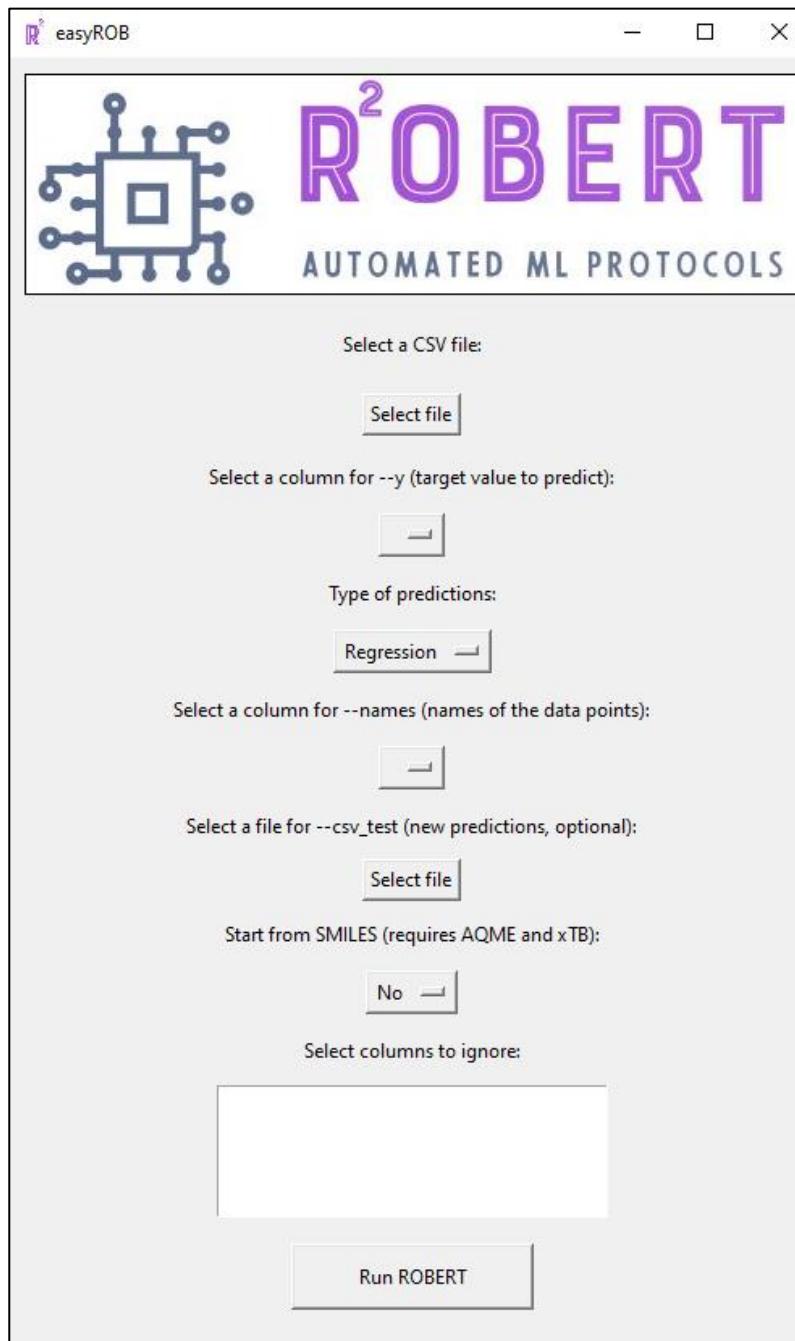
Points	R ²	Outliers	Desc:pts
●●	> 0.85	< 7.5%	< 1:3
●	0.70 - 0.85	7.5 - 15%	1:3 - 1:10
-	< 0.70	> 15%	> 1:10

Verify tests (up to ●●●●)	5-fold CV	y-shuffle
●●●●	●	●



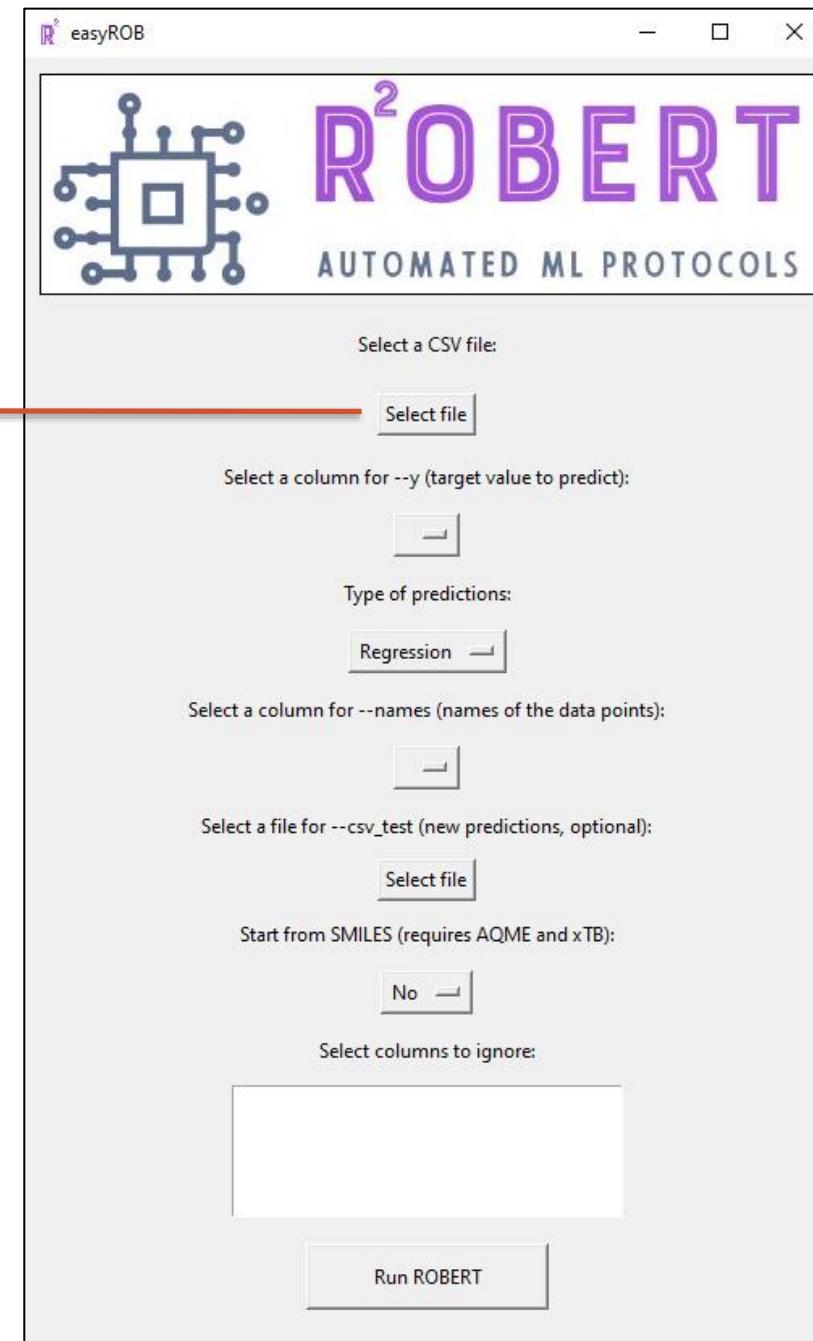
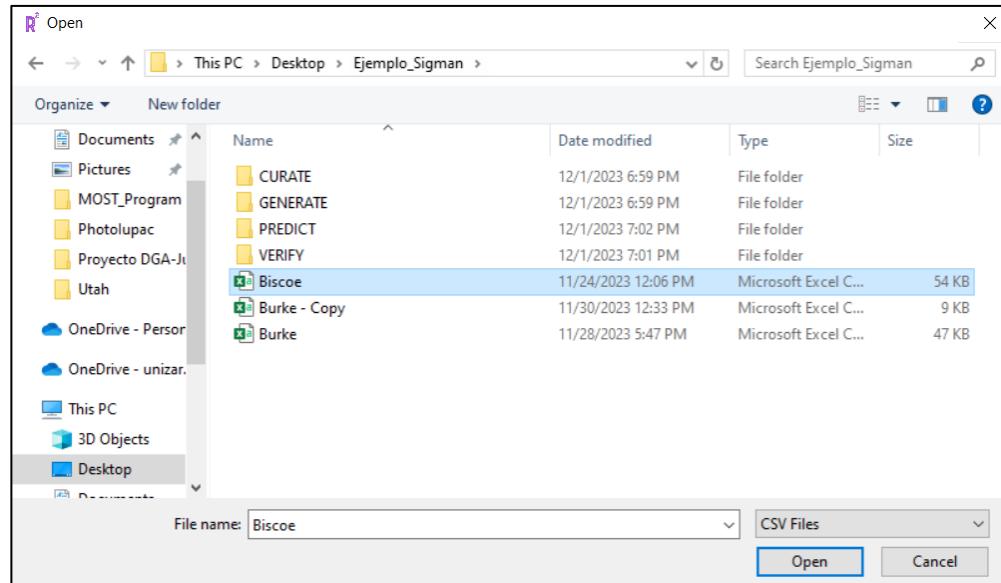
ROBERT: automation of ML protocols

Graphical User Interface: easyROB



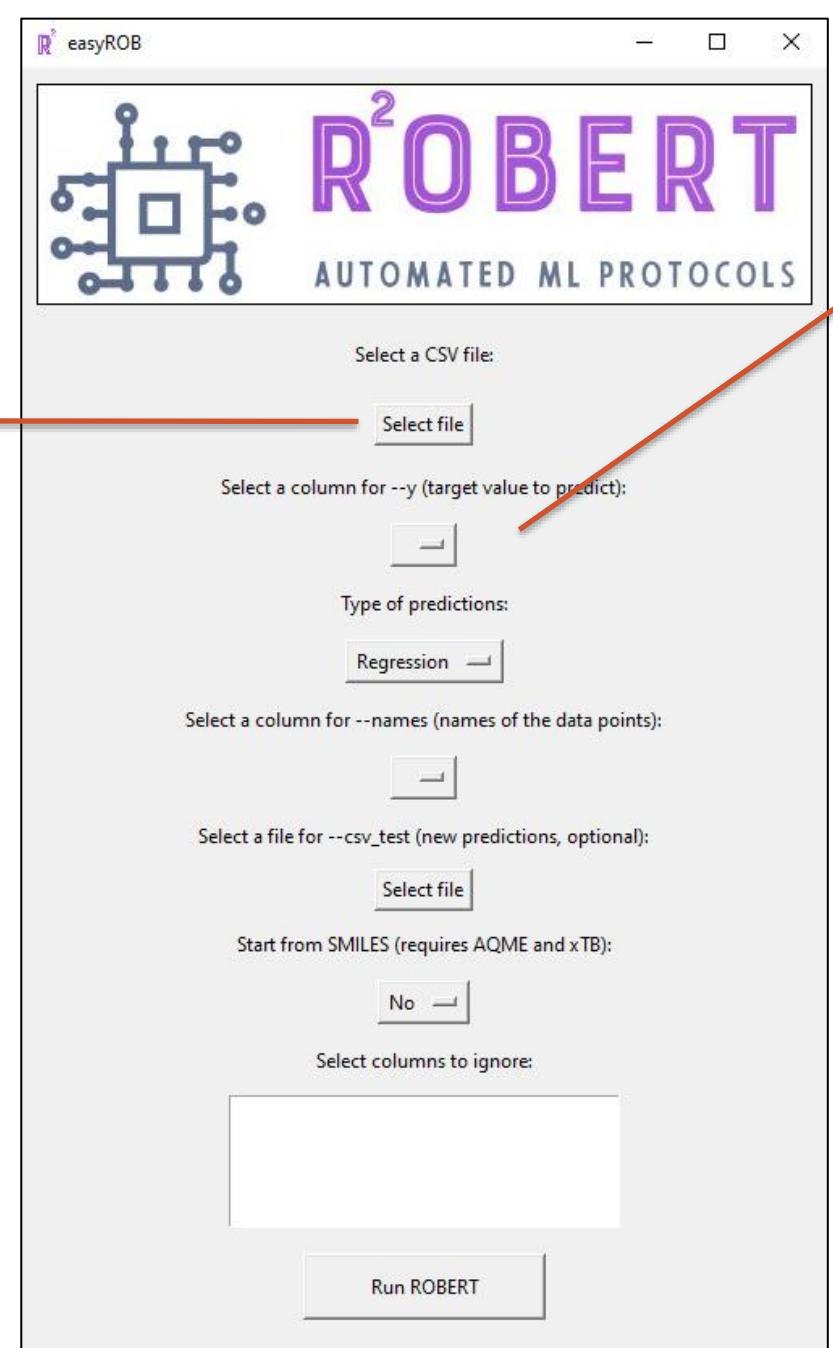
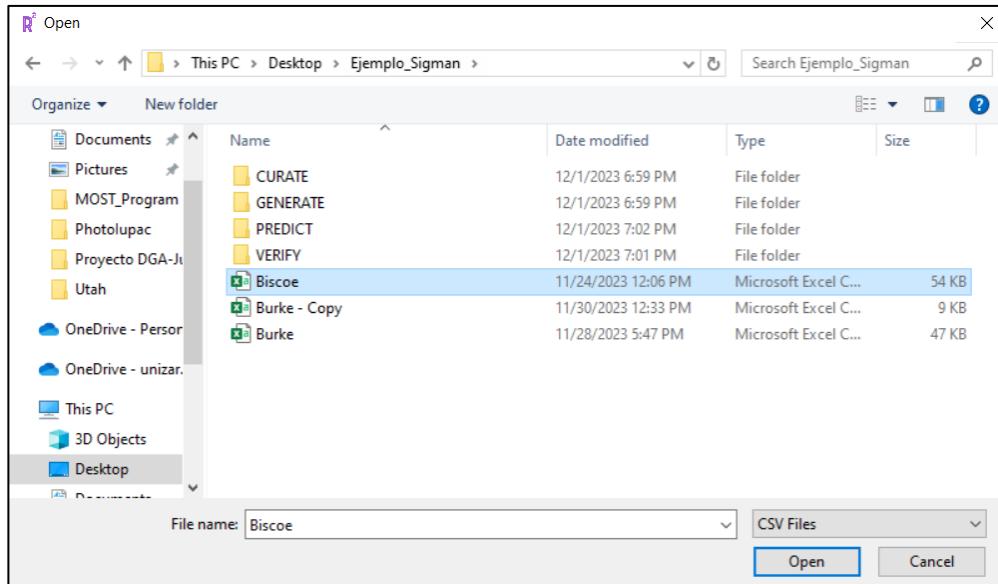
ROBERT: automation of ML protocols

Graphical user interface: easyROB



ROBERT: automation of ML protocols

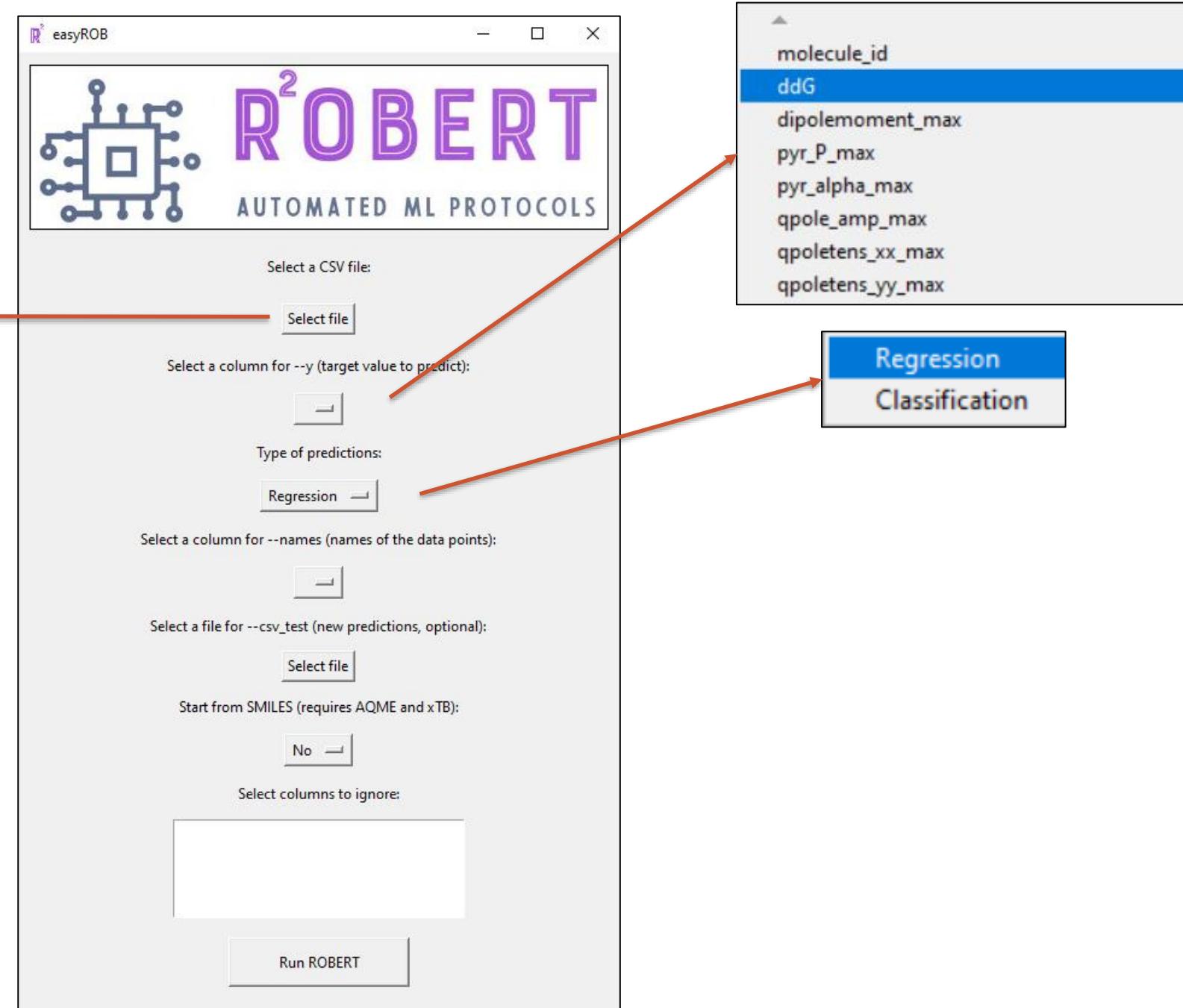
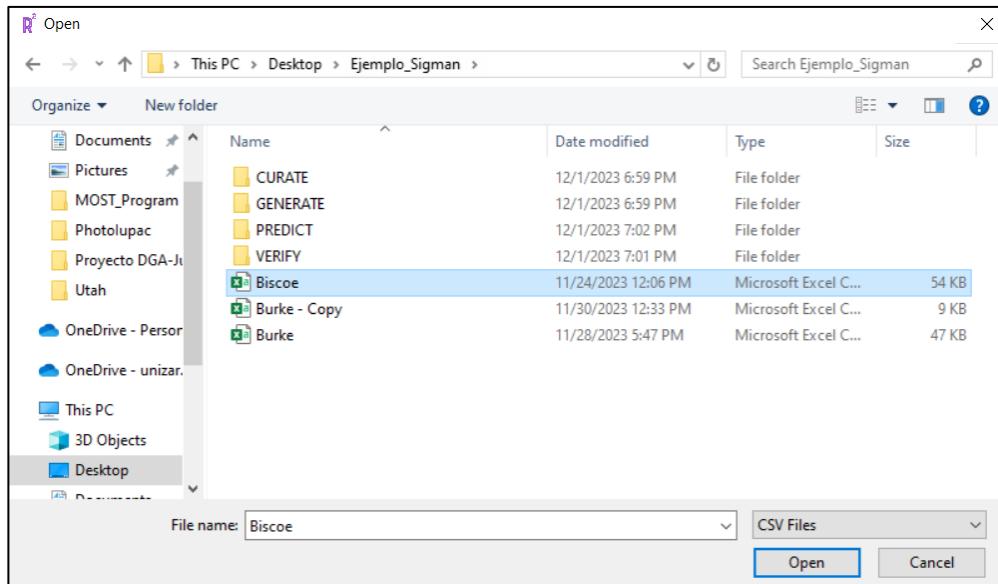
Graphical user interface: easyROB



molecule_id
ddG
dipolemoment_max
pyr_P_max
pyr_alpha_max
qpole_amp_max
qpletens_xx_max
qpletens_yy_max

ROBERT: automation of ML protocols

Graphical user interface: easyROB



The easyROB interface is a graphical application window titled 'easyROB'. It features the R²OBERT logo and the text 'AUTOMATED ML PROTOCOLS'. The window contains several configuration fields:

- 'Select a CSV file:' with a 'Select file' button pointing to the 'Biscoe' file in the File Explorer.
- 'Select a column for --yy (target value to predict):' with a dropdown menu.
- 'Type of predictions:' with a dropdown menu currently set to 'Regression' (indicated by an arrow).
- 'Select a column for --names (names of the data points):' with a dropdown menu.
- 'Select a file for --csv_test (new predictions, optional):' with a 'Select file' button.
- 'Start from SMILES (requires AQME and xTB):' with a 'No' button.
- 'Select columns to ignore:' with an input field.

Below the main window is a 'Run ROBERT' button.

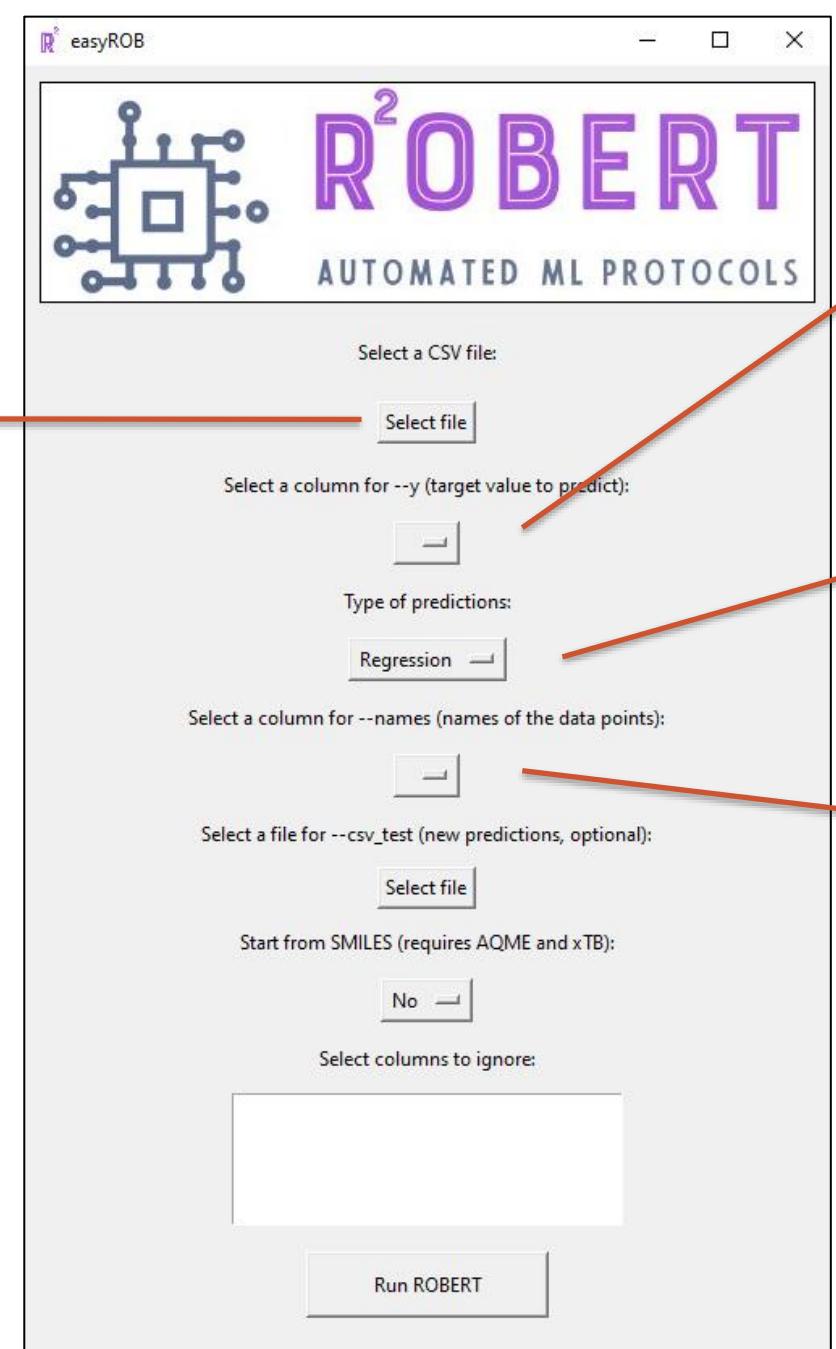
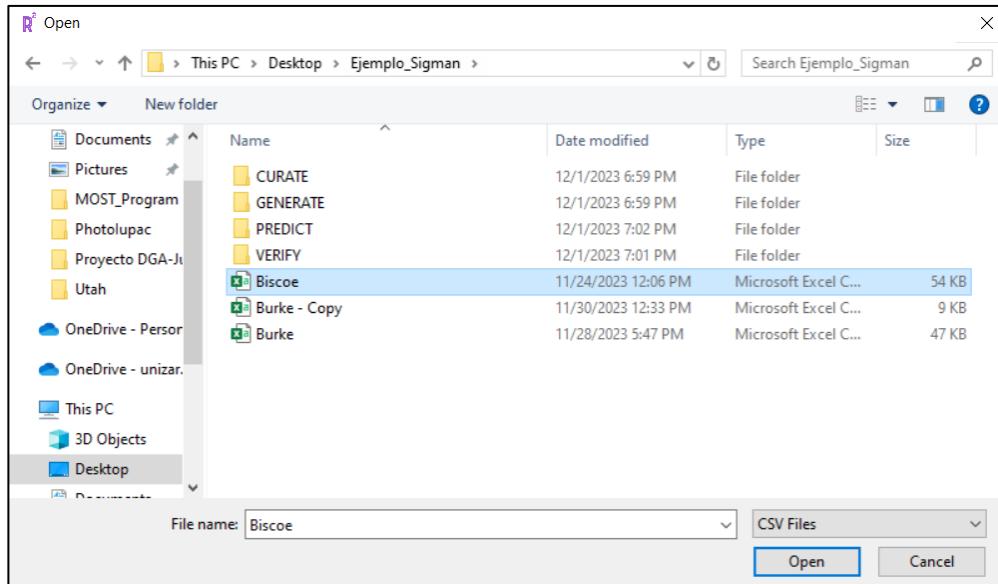
To the right of the main window, a list of molecular properties is shown:

molecule_id
ddG
dipolemoment_max
pyr_P_max
pyr_alpha_max
qpole_amp_max
qpletens_xx_max
qpletens_yy_max

Below this list is a box containing 'Regression' and 'Classification' buttons.

ROBERT: automation of ML protocols

Graphical user interface: easyROB

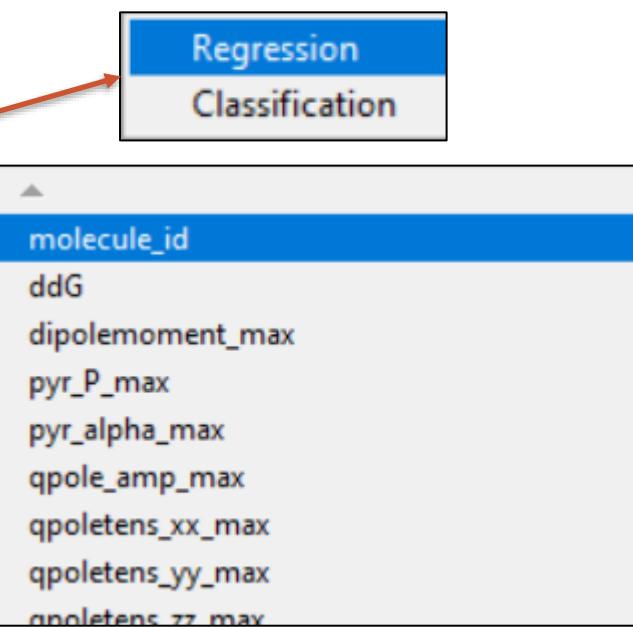
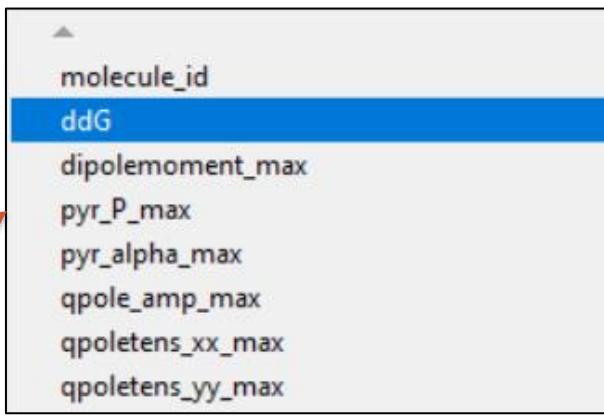
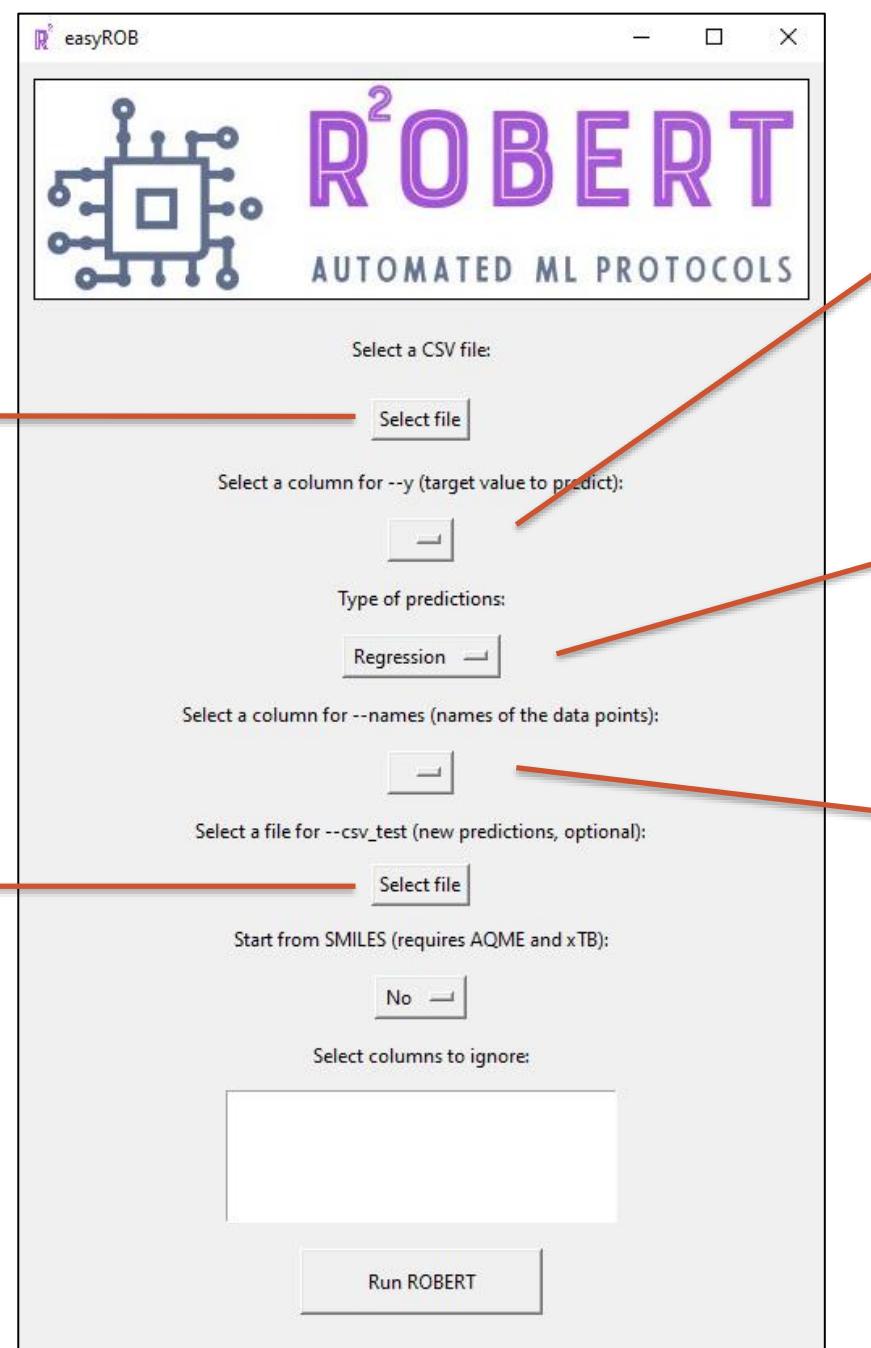
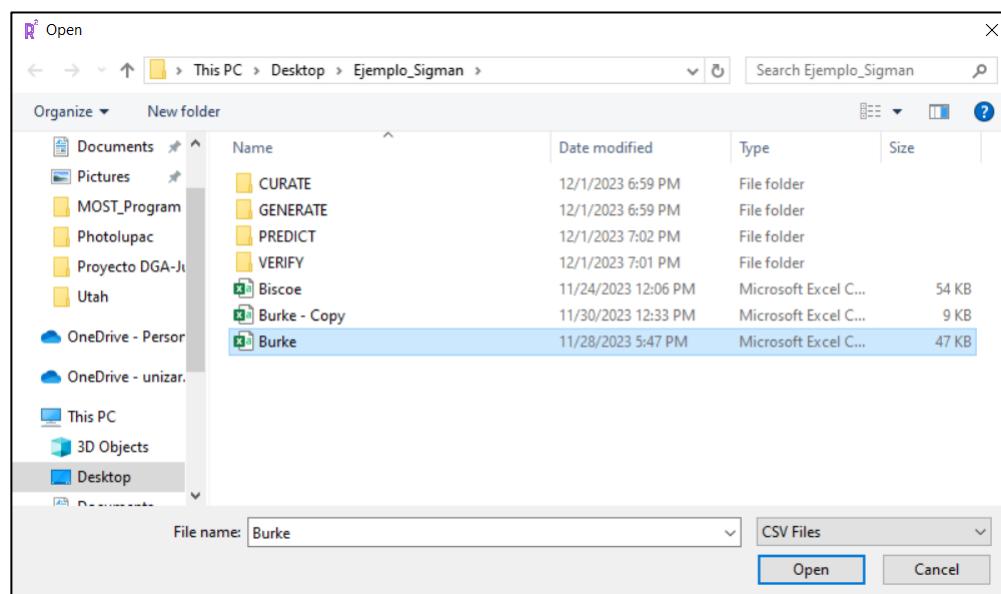
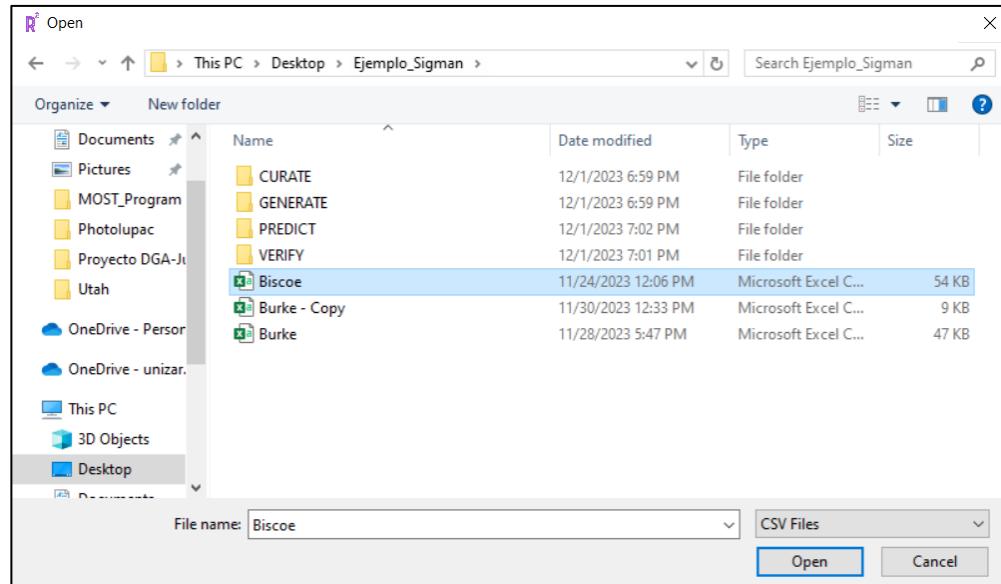


molecule_id
ddG
dipolemoment_max
pyr_P_max
pyr_alpha_max
qpole_amp_max
qpoletens_xx_max
qpoletens_yy_max

molecule_id
ddG
dipolemoment_max
pyr_P_max
pyr_alpha_max
qpole_amp_max
qpoletens_xx_max
qpoletens_yy_max
qpoletens_zz_max

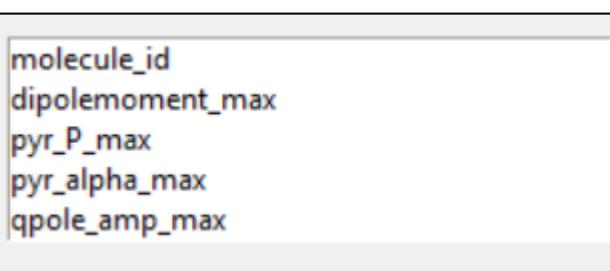
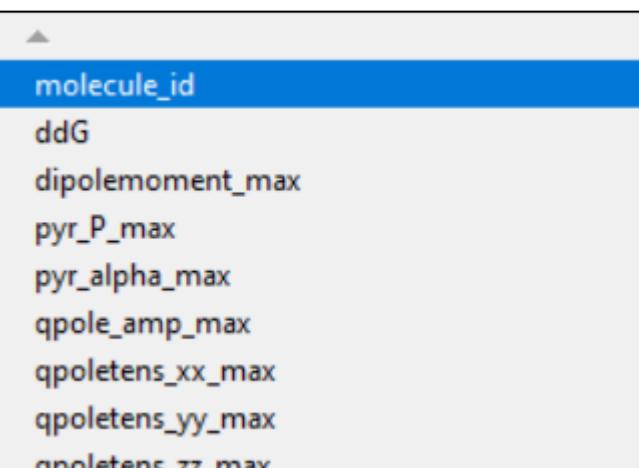
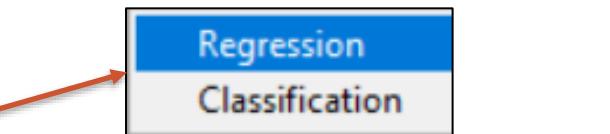
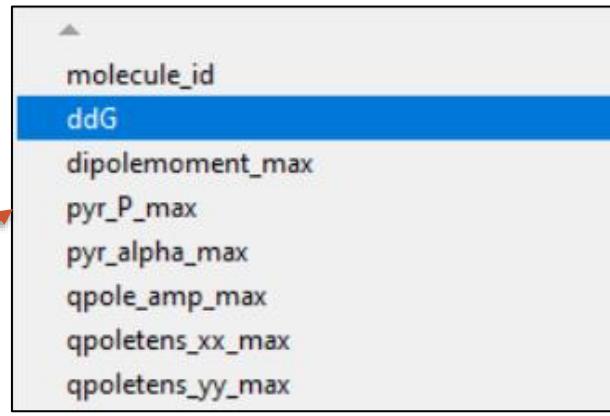
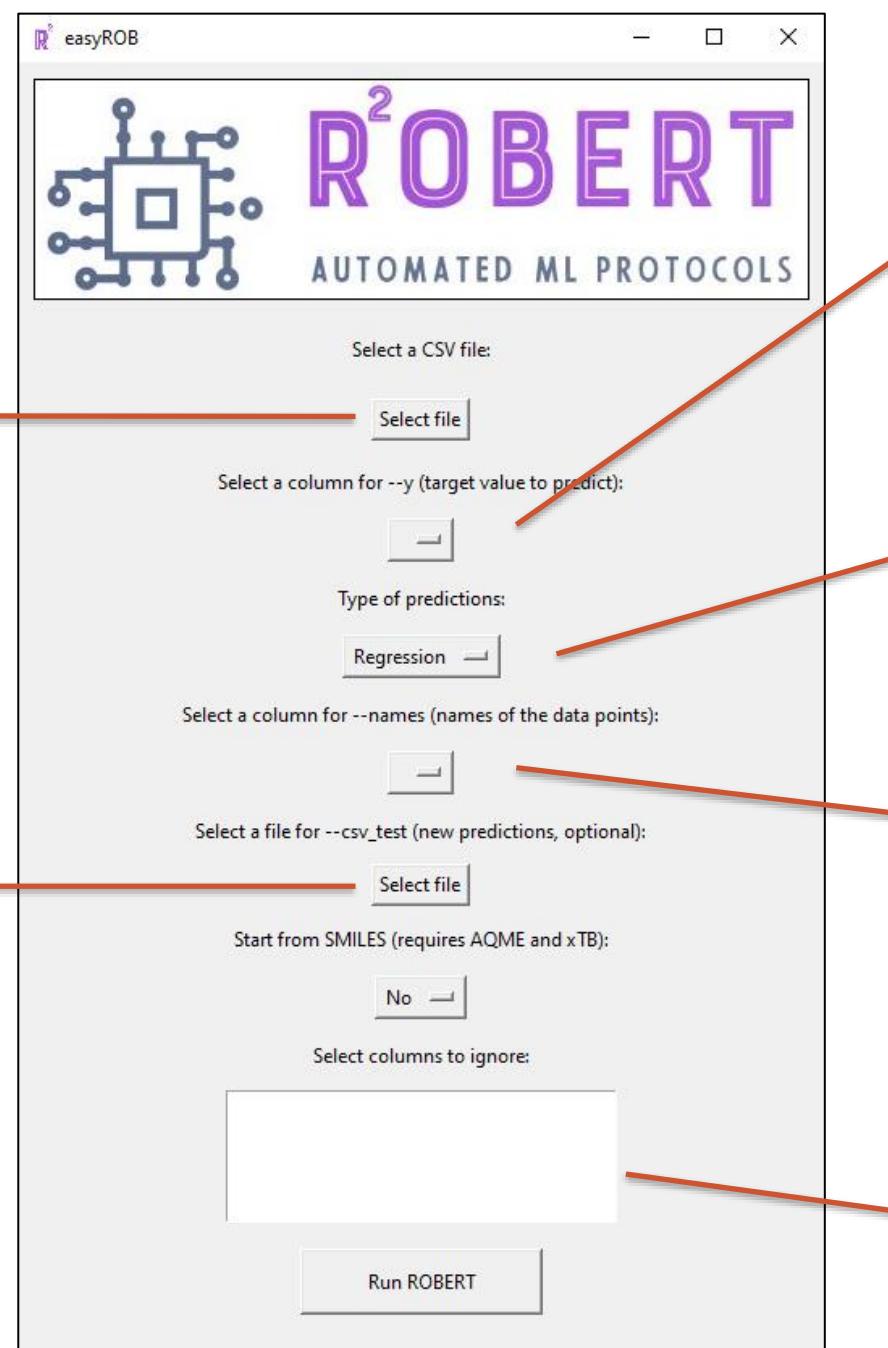
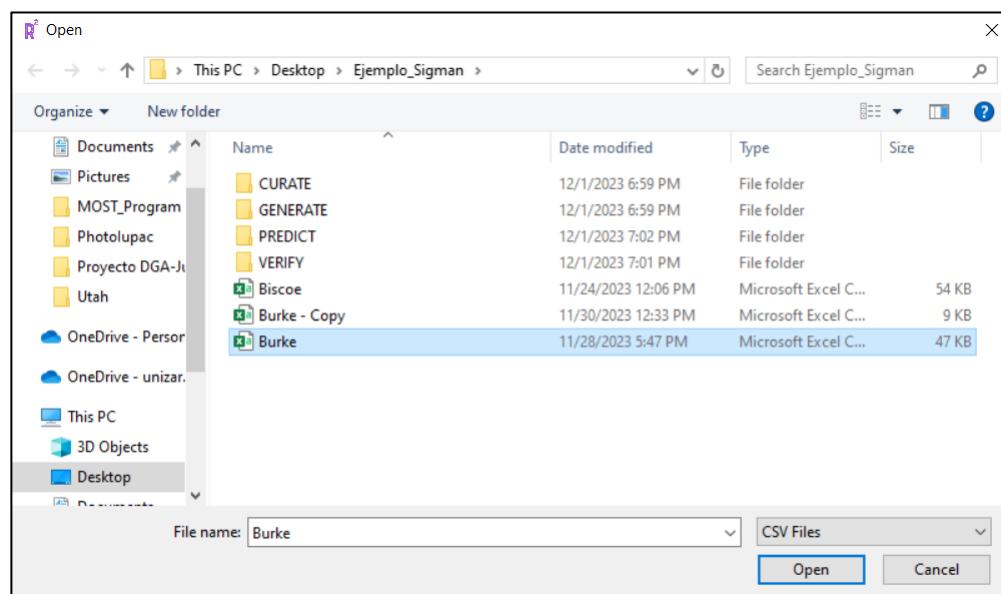
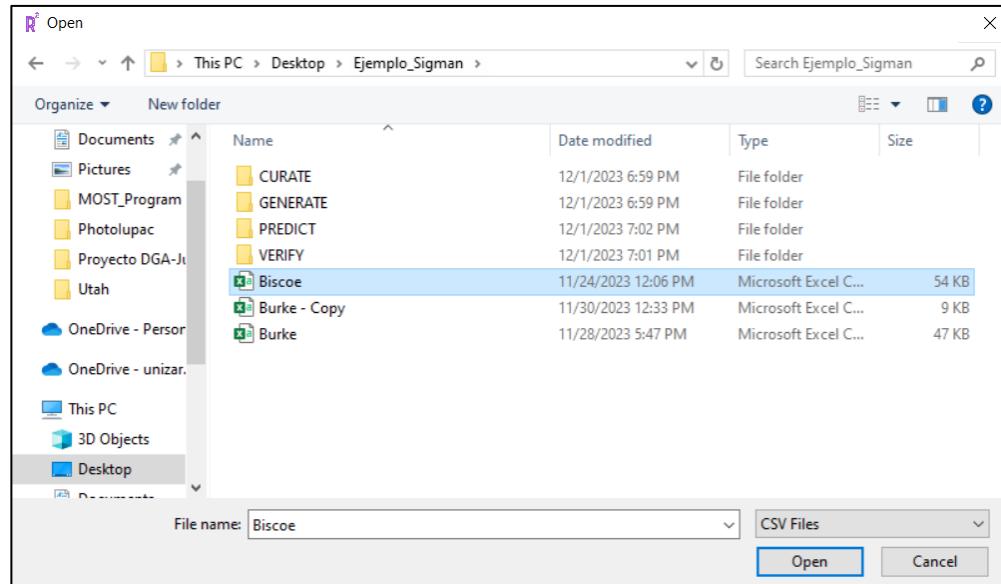
ROBERT: automation of ML protocols

Graphical user interface: easyROB



ROBERT: automation of ML protocols

Graphical user interface: easyROB



Automated generation of ML predictors



+



Automated generation of ML predictors

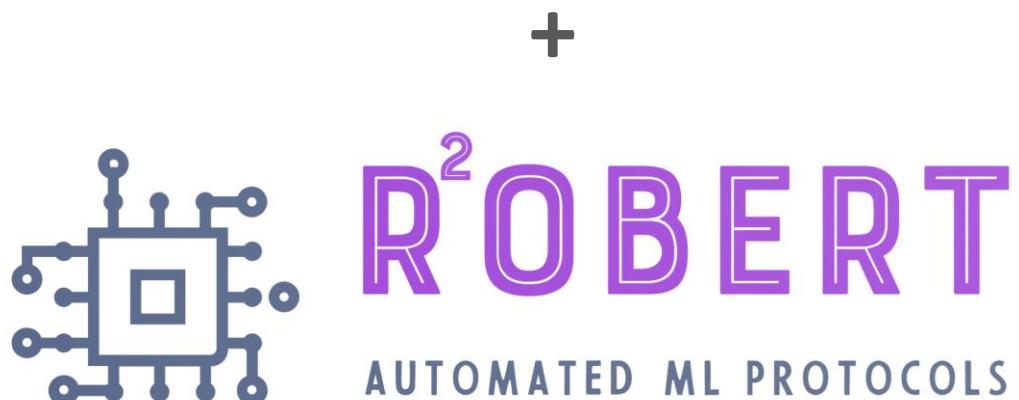


+



Only one command line: `python -m robert --csv_name name.csv --y target_value --aqme`

Automated generation of ML predictors



Only one command line: `python -m robert --csv_name name.csv --y target_value --aqme`

Initial AQME workflow

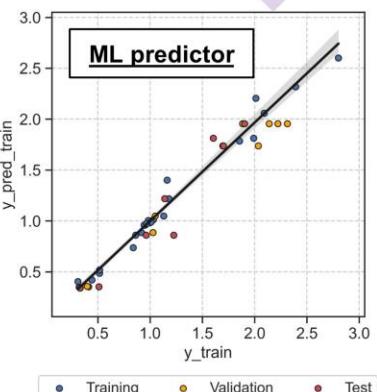
Input: CSV with SMILES

SMILES	code_name	solub.
c1ccsc1	mol_5	-1.33
CCCC=C	mol_13	-2.68
CC(C)Cl	mol_16	-1.41
C(C=C)Cl	mol_34	-1.64

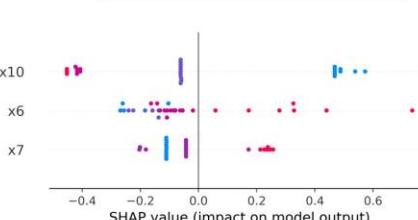
CSV database

One command line

Outputs:

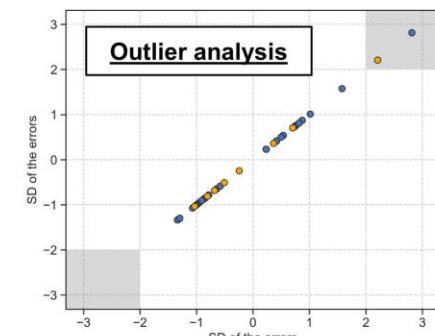


SHAP feature analysis

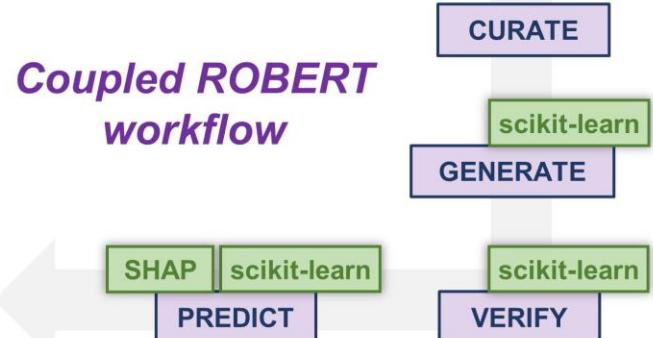


Automated tasks in AQME

- Conformational search with RDKit
- Molecular featurization using xTB & RDKit
- Boltzmann averaging of descriptors
- Creation of a CSV database for ROBERT

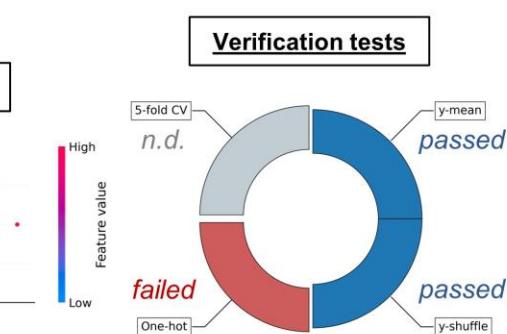


Coupled ROBERT workflow



Automated tasks in ROBERT

- Data curation of RDKit/xTB descriptors
- Screening of hyperoptimized ML models
- Screening of partition sizes
- New models with relevant features (PFI)
- Prediction of test set (if any) & SHAP analysis
- Analysis of outliers across different sets
- Verification tests to ensure predictive ability



CSV

Automated generation of ML predictors

ESOL: Estimating Aqueous Solubility Directly from Molecular Structure

John S. Delaney

[View Author Information](#) ▾ **Cite this:** *J. Chem. Inf. Comput. Sci.* 2004, 44, 3, 1000–1005

Publication Date: March 13, 2004 ▾

<https://doi.org/10.1021/ci034243x>

Copyright © 2004 American Chemical Society

[RIGHTS & PERMISSIONS](#)

Article Views

9422

Altmetric

38

Citations

395[LEARN ABOUT THESE METRICS](#)[Share](#) [Add to](#) [Export](#)[Read Online](#) PDF (72 KB) Supporting Info (1) »**SUBJECTS:** Molecular modeling, Molecules, ▾

Automated generation of ML predictors

1 command line

ESOL: Estimating Aqueous Solubility Directly from Molecular Structure

John S. Delaney

[View Author Information](#) ▾

 [Cite this:](#) *J. Chem. Inf. Comput. Sci.* 2004, 44, 3, 1000–1005

Publication Date: March 13, 2004 ▾

<https://doi.org/10.1021/ci034243x>

Copyright © 2004 American Chemical Society

[RIGHTS & PERMISSIONS](#)

Article Views

9422

Altmetric

38

Citations

395

[LEARN ABOUT THESE METRICS](#)

Share Add to Export



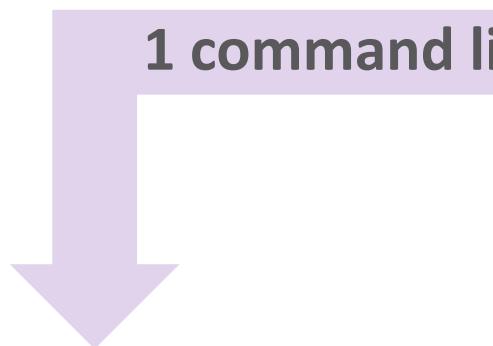
[Read Online](#)

 PDF (72 KB)

 Supporting Info (1) »

SUBJECTS: Molecular modeling, Molecules, ▾

Automated generation of ML predictors



ESOL: Estimating Aqueous Solubility Directly from Molecular Structure

John S. Delaney

[View Author Information](#) ▾

[Cite this: J. Chem. Inf. Comput. Sci. 2004, 44, 3, 1000–1005](#)
 Publication Date: March 13, 2004
<https://doi.org/10.1021/ci034243x>
 Copyright © 2004 American Chemical Society
[RIGHTS & PERMISSIONS](#)

Article Views: 9422 | Altmetric: 38 | Citations: 395 | LEARN ABOUT THESE METRICS

[Share](#) [Add to](#) [Export](#)   

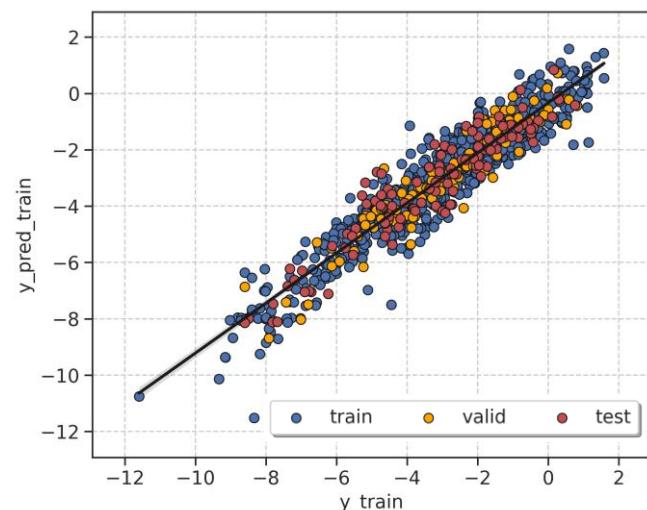
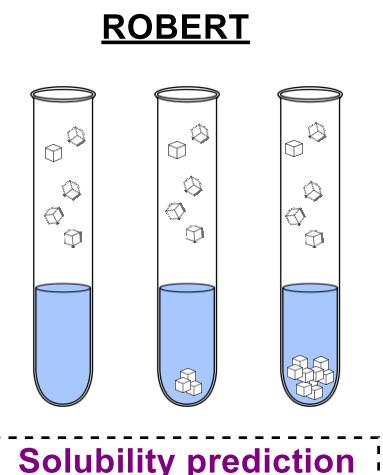
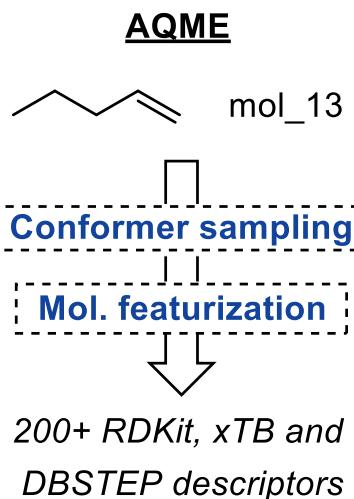
[Read Online](#)  [PDF \(72 KB\)](#)  [Supporting Info \(1\) »](#) **SUBJECTS:** Molecular modeling, Molecules, ▾

A Estimating aqueous solubility

Input: CSV with SMILES

ESOL database

SMILES	code_name	solub.
c1ccsc1	mol_5	-1.33
CCCC=C	mol_13	-2.68
CC(C)Cl	mol_16	-1.41
CIC(=C)Cl	mol_34	-1.64



ROBERT SCORE

ML model: MVL
 Proportion Train:Validation:Test = 81:9:10



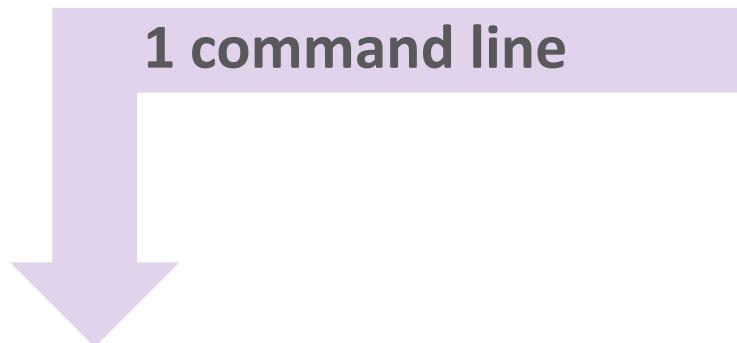
The model has a score of 10/10

- The test set shows an R² of 0.89
- The valid. set has 4.9% of outliers
- Using 1012:47 points(train+valid.):descriptors
- The valid. set passes 4 VERIFY tests

Only one command line: `python -m robert --csv_name solubility.csv --y solubility --aqme --qdescp_keywords "--qdescp_solvent h2o"`

Automated generation of ML predictors

1 command line



ESOL: Estimating Aqueous Solubility Directly from Molecular Structure

John S. Delaney

[View Author Information](#) ▾

Cite this: *J. Chem. Inf. Comput. Sci.* 2004, 44, 3, 1000–1005

Publication Date: March 13, 2004

<https://doi.org/10.1021/ci034243x>

Copyright © 2004 American Chemical Society

[RIGHTS & PERMISSIONS](#)

Article Views

9422

Altmetric

38

Citations

395

[LEARN ABOUT THESE METRICS](#)

Share Add to Export



[Read Online](#)

[PDF \(72 KB\)](#)

[SI Supporting Info \(1\) »](#)

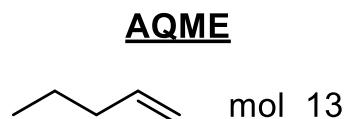
SUBJECTS: Molecular modeling, Molecules, ▾

A Estimating aqueous solubility

Input: CSV with SMILES

ESOL database

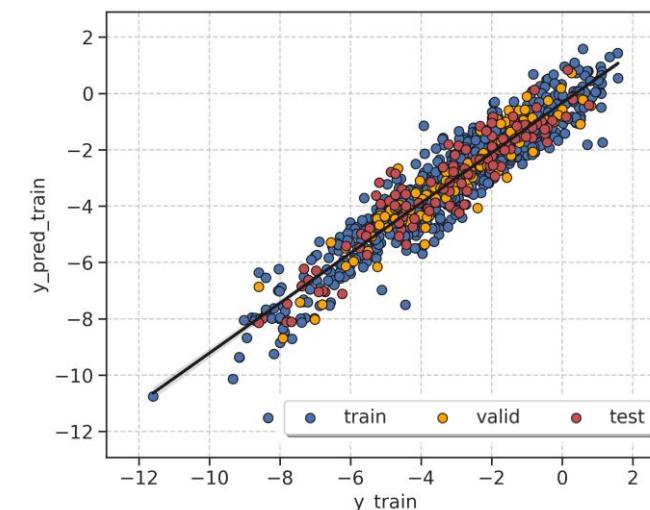
SMILES	code_name	solub.
c1ccsc1	mol_5	-1.33
CCCC=C	mol_13	-2.68
CC(C)Cl	mol_16	-1.41
CIC(=C)Cl	mol_34	-1.64



200+ RDKit, xTB and DBSTEP descriptors

ROBERT

Solubility prediction



ROBERT SCORE

ML model: MVL

Proportion Train:Validation:Test = 81:9:10



Train : $R^2 = 0.89$, MAE = 0.54, RMSE = 0.7

Validation : $R^2 = 0.89$, MAE = 0.53, RMSE = 0.68

Test : $R^2 = 0.89$, MAE = 0.59, RMSE = 0.74

Only one command line: `python -m robert --csv_name solubility.csv --y solubility --aqme --qdescp_keywords "--qdescp_solvent h2o"`

Automatic generation of predictors

Input: CSV with SMILES

SMILES	code_name	ΔE^\ddagger
[Ir]([P+])(CC)...	ir_tbp_1_dft-pet3...	8.9
[Ir]([P+](C)...	ir_tbp_1_dft-pme3...	6.5
[Ir]([As+](C)...	ir_tbp_1_dft-asme3...	17.9
[Ir]([n+]1ccn...	ir_tbp_1_dft-pyz...	22

Machine learning dihydrogen activation in the chemical space surrounding Vaska's complex†‡

Pascal Friederich,  ^{abc} Gabriel dos Passos Gomes,  ^{ac} Riccardo De Bin, ^d Alán Aspuru-Guzik ^{*acef}
 and David Balcells  ^{*g}

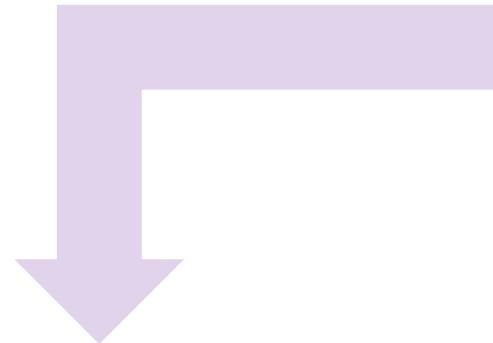
 Check for updates

Automatic generation of predictors

Input: CSV with SMILES

SMILES	code_name	ΔE^\ddagger
[Ir]([P+])(CC)...	ir_tbp_1_dft-pet3...	8.9
[Ir]([P+](C)...	ir_tbp_1_dft-pme3...	6.5
[Ir]([As+](C)...	ir_tbp_1_dft-asme3...	17.9
[Ir]([n+]1ccn...	ir_tbp_1_dft-pyz...	22

1 command line



Machine learning dihydrogen activation in the chemical space surrounding Vaska's complex†‡

Pascal Friederich,  ^{abc} Gabriel dos Passos Gomes,  ^{ac} Riccardo De Bin, ^d Alán Aspuru-Guzik ^{*acef}
and David Balcells  ^{*g}

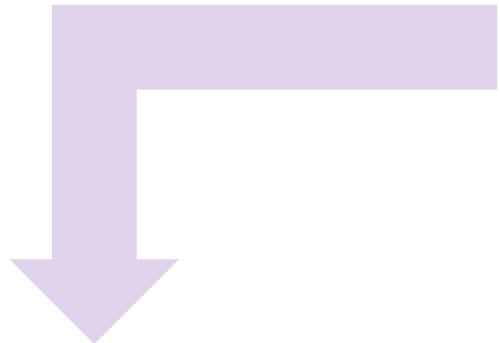


Automatic generation of predictors

Input: CSV with SMILES

SMILES	code_name	ΔE^\ddagger
[Ir]([P+])(CC)...	ir_tbp_1_dft-pet3...	8.9
[Ir]([P+](C)...	ir_tbp_1_dft-pme3...	6.5
[Ir]([As+](C)...	ir_tbp_1_dft-asme3...	17.9
[Ir]([n+]1ccn...	ir_tbp_1_dft-pyz...	22

1 command line



Machine learning dihydrogen activation in the chemical space surrounding Vaska's complex†‡



Pascal Friederich,  ^{abc} Gabriel dos Passos Gomes,  ^{ac} Riccardo De Bin, ^d Alán Aspuru-Guzik ^{*acef} and David Balcells  ^{*g}

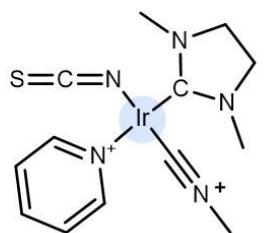
B – Application to catalysis: Estimating activation energies

Input: CSV with SMILES

Vaska's complex database

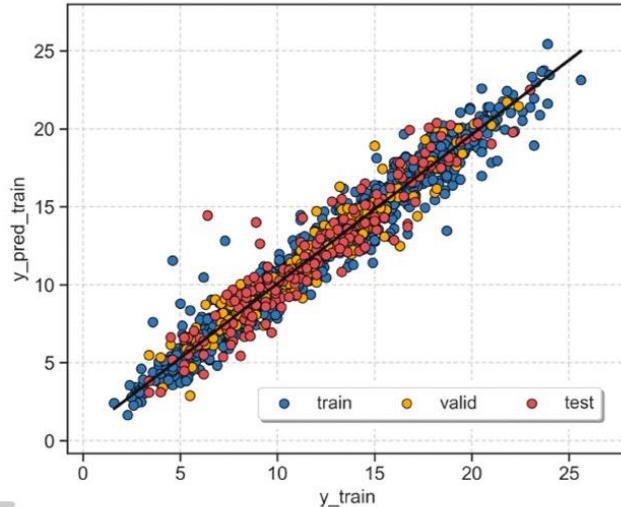
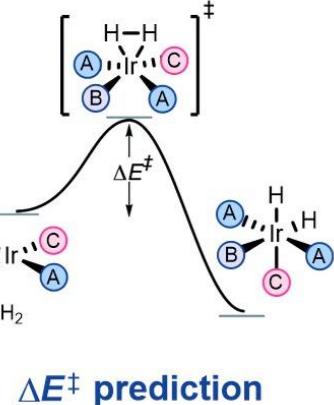
SMILES	code_name	ΔE^\ddagger
[Ir]([P+])(CC)...	ir_tbp_1_dft-pet3...	8.9
[Ir]([P+](C)...	ir_tbp_1_dft-pme3...	6.5
[Ir]([As+](C)...	ir_tbp_1_dft-asme3...	17.9
[Ir]([n+]1ccn...	ir_tbp_1_dft-pyz...	22

Step 1: AQME



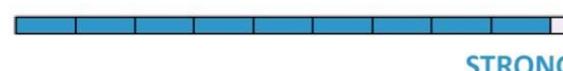
200+ electronic & steric atomic/molecular descs.
(RDKit, xTB & DBSTEP)

Step 2: ROBERT



 ROBERT SCORE

ML model: NN
Proportion Train:Validation:Test = 81:9:10

 STRONG

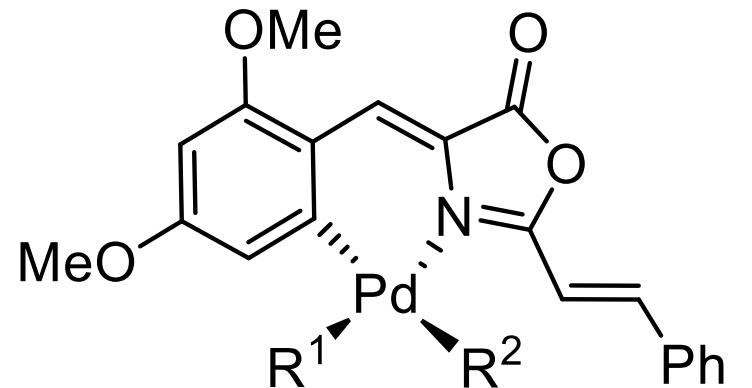
The model has a score of 9/10

- The test set shows an R^2 of 0.89
- The valid. set has 10.5% of outliers
- Using 1711:21 points(train+valid.):descriptors
- The valid. set passes 4 VERIFY tests

Only one command line: `python -m robert --csv_name vaskas.csv --y barrier --aqme --qdescp_keywords "--qdescp_atoms [Ir]"`

Real example of our lab

- Discovery of new Pd-oxazolone luminiscent compounds



- 1a:** R¹ = Pyr, R² = TFA (18% QY)
1b: R¹ = Pyr, R² = Cl (12%)
1c: R¹ = Pyr, R² = Pyr (28%)
1d: R¹ = NHC, R² = TFA (15%)
1e: 8-Aminoquinoline (12%)

Eur. J. Org. Chem., **2018**, 6158-6166.

Molecules **2021**, *26*, 1238.

Org. Biomol. Chem., **2021**, *19*, 2773-2783

Inorg. Chem. **2023**, *62*, 9792–9806.

From all the compounds reported across four articles, only five examples with QY > 10%!

Real example of our lab

- Discovery of new Pd-oxazolone luminiscent compounds

Eur. J. Org. Chem., **2018**: 6158-6166.

Molecules **2021**, *26*, 1238.

Org. Biomol. Chem., **2021**, *19*, 2773-2783

Inorg. Chem. **2023**, *62*, 9792–9806.

From all the compounds reported across four articles, only five examples with QY > 10%!



We could go to the lab and try new possible combinations



Experim.

Real example of our lab

- Discovery of new Pd-oxazolone luminiscent compounds

Eur. J. Org. Chem., 2018: 6158-6166.

Molecules 2021, 26, 1238.

Org. Biomol. Chem., 2021, 19, 2773-2783

Inorg. Chem. 2023, 62, 9792–9806.

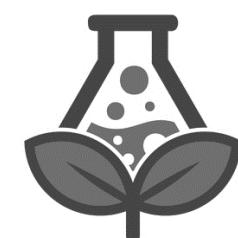
From all the compounds reported across four articles, only five examples with QY > 10%!



~~We could go to the lab and try each of the combinations~~



Experim.



Real example of our lab

- Discovery of new Pd-oxazolone luminiscent compounds

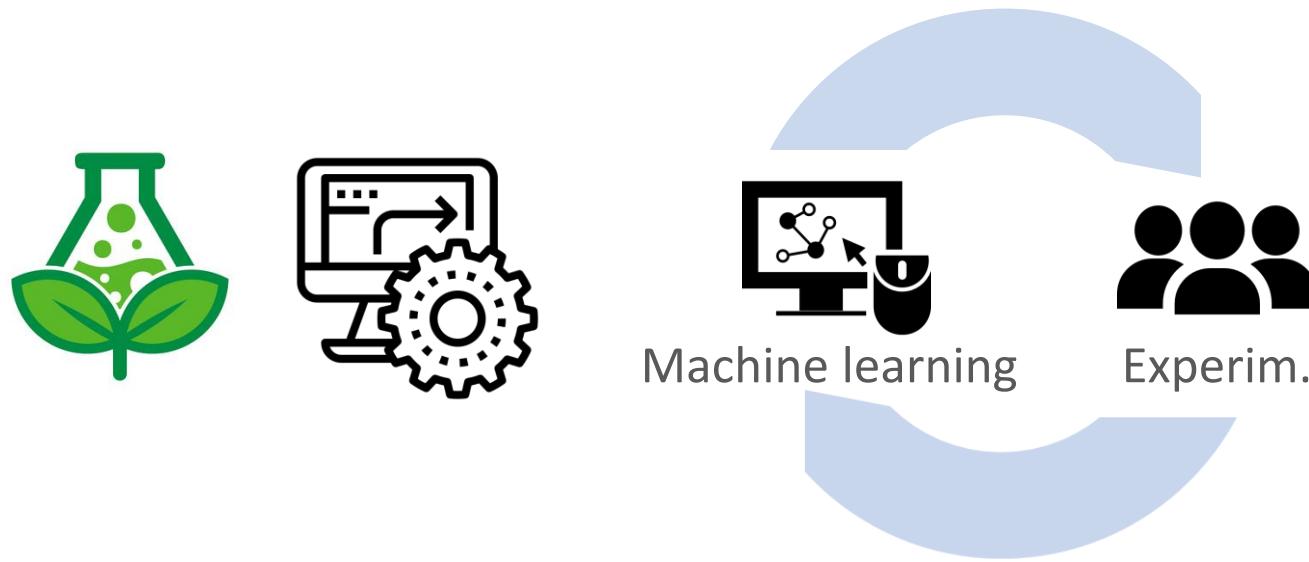
Eur. J. Org. Chem., **2018**: 6158-6166.

Molecules **2021**, *26*, 1238.

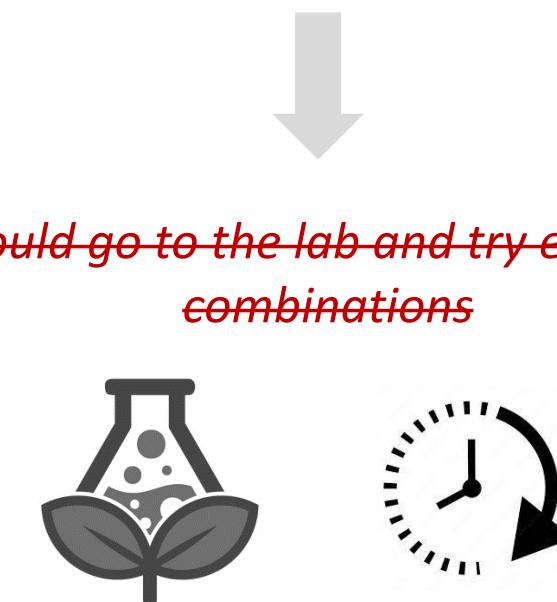
Org. Biomol. Chem., **2021**, *19*, 2773-2783

Inorg. Chem. **2023**, *62*, 9792–9806.

From all the compounds reported across four articles, only five examples with QY > 10%!

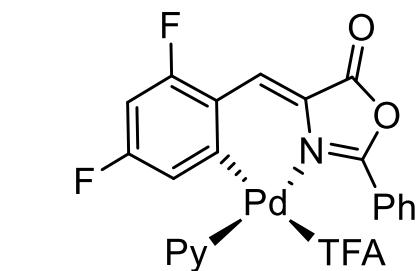
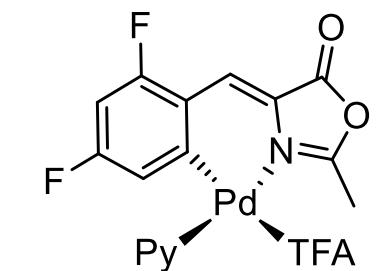
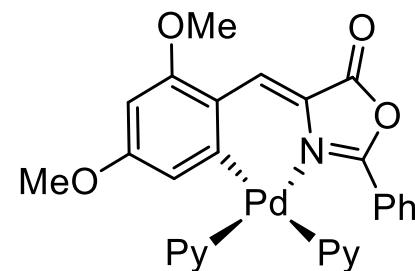
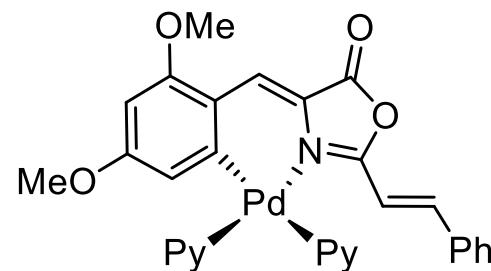
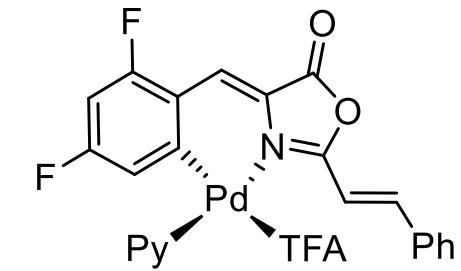
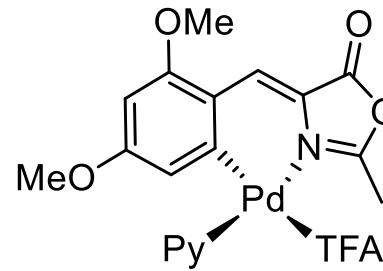
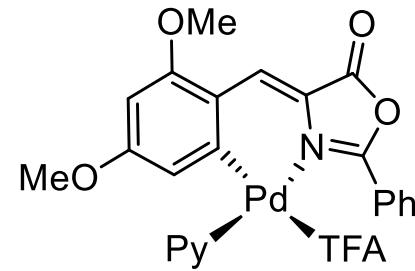
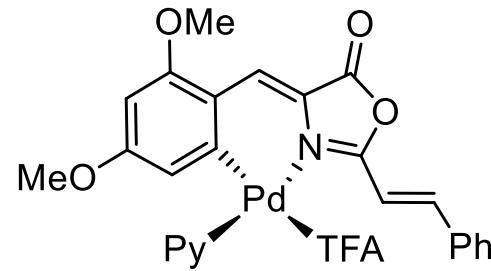


~~We could go to the lab and try each of the combinations~~



Real example of our lab

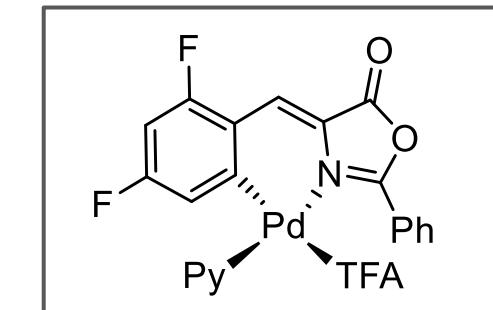
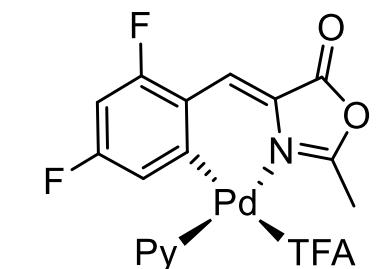
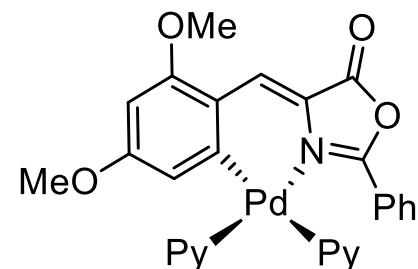
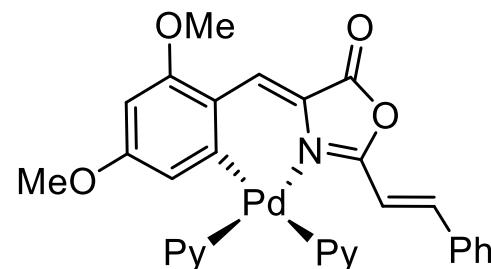
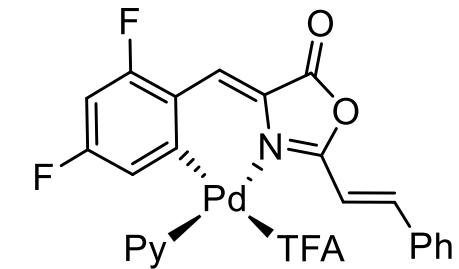
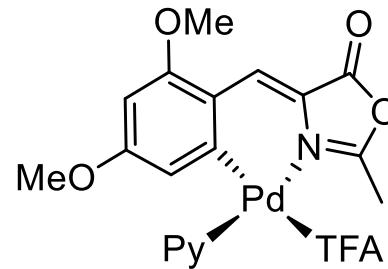
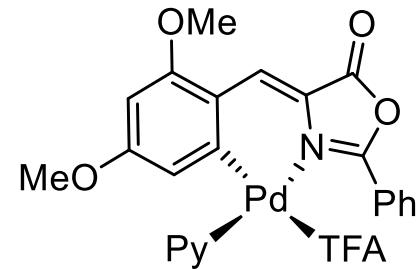
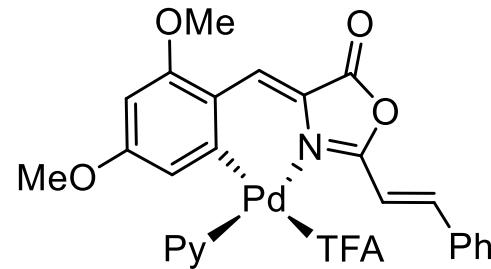
1. Draw in ChemDraw the 23 compounds that already exist



...

Real example of our lab

1. Draw in ChemDraw the 23 compounds that already exist



ccc(F)cc(F)C=C...

2. Convert the structures to SMILES strings (Ctrl + Alt + C)

Real example of our lab

3. Create the Excel database with names, SMILES and QY

code_name	SMILES	QY(%)
complex_1	O=C1OC=N/C1=C\C2...	2
complex_2	[Pd]C1=CC(OC)=CC(OC)...	5
complex_3	FC1=C2C([Pd][N]C=C2)...	12
...		

Real example of our lab

3. Create the Excel database with names, SMILES and QY

code_name	SMILES	QY(%)
complex_1	O=C1OC=N/C1=C\C2...	2
complex_2	[Pd]C1=CC(OC)=CC(OC)...	5
complex_3	FC1=C2C([Pd][N]C=C2)...	12
...		

4. Run ROBERT

```
> python -m robert --y QY(%) --names code_name --aqme --csv Pd.csv
```

Real example of our lab

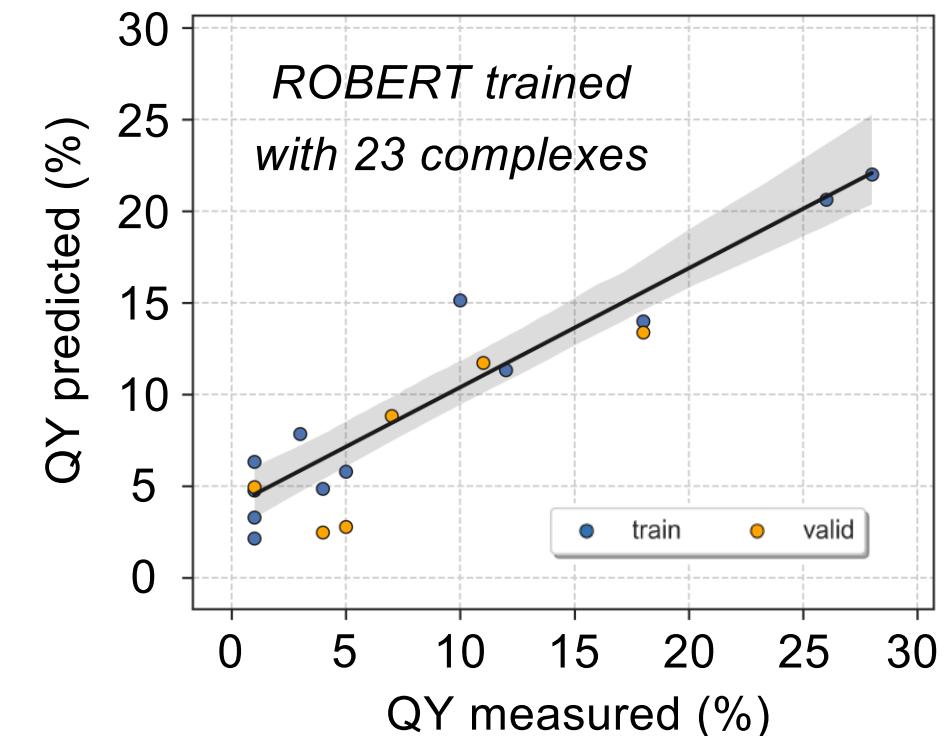
3. Create the Excel database with names, SMILES and QY

code_name	SMILES	QY(%)
complex_1	O=C1OC=N/C1=C\C2...	2
complex_2	[Pd]C1=CC(OC)=CC(OC)...	5
complex_3	FC1=C2C([Pd][N]C=C2)...	12
...		

4. Run ROBERT

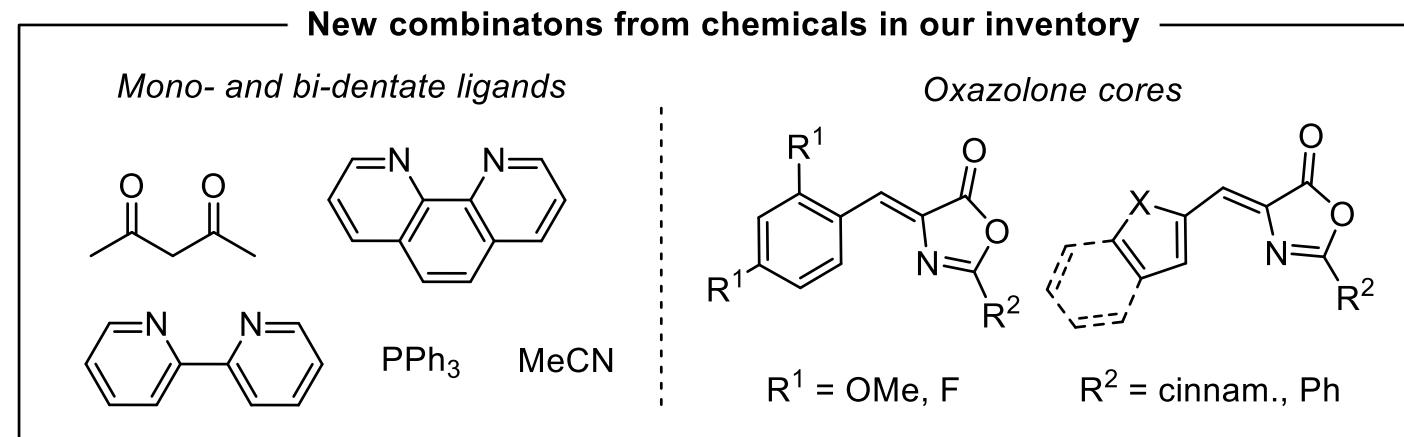
```
> python -m robert --y QY(%) --names code_name --aqme --csv Pd.csv
```

Aprox. 30-45 min



Real example of our lab

- ROBERT for predicting new candidates

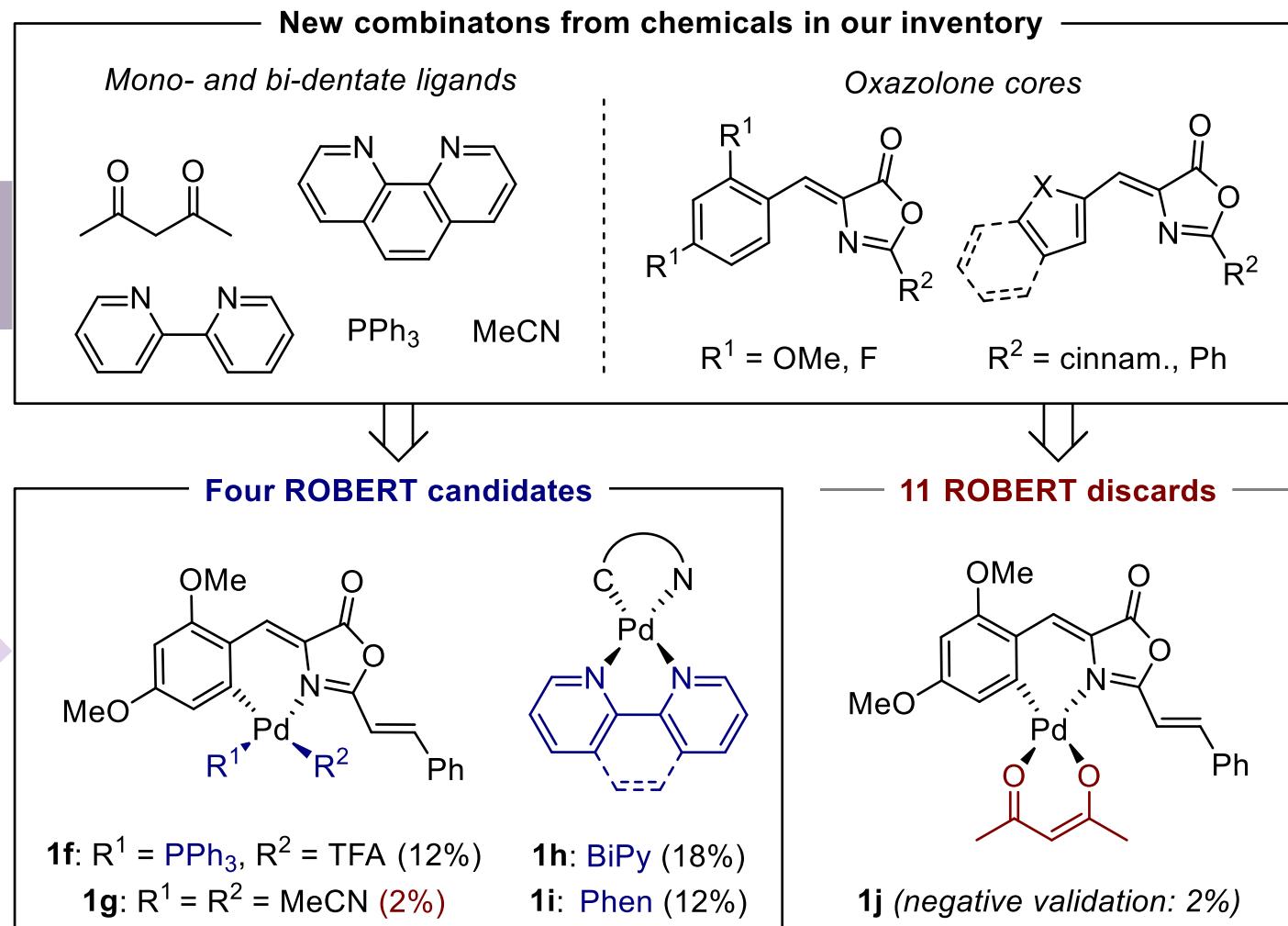
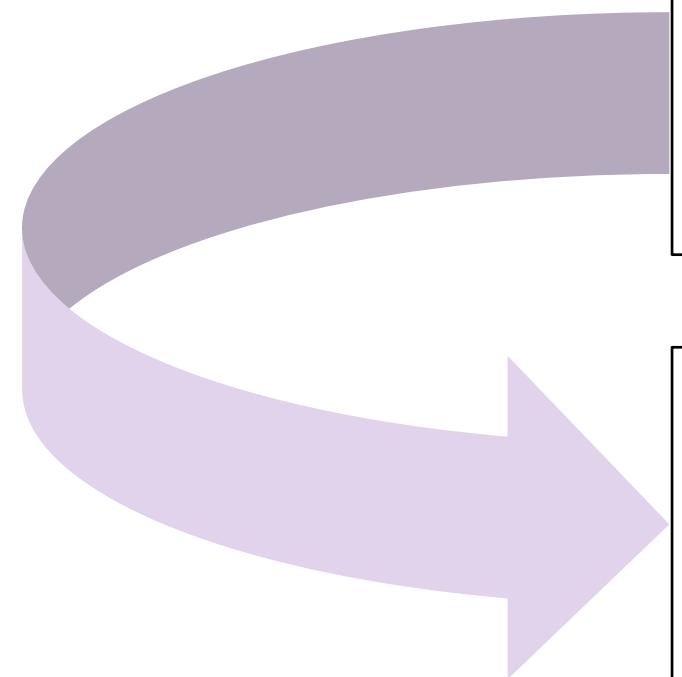


Real example of our lab

- ROBERT for predicting new candidates



Machine learning



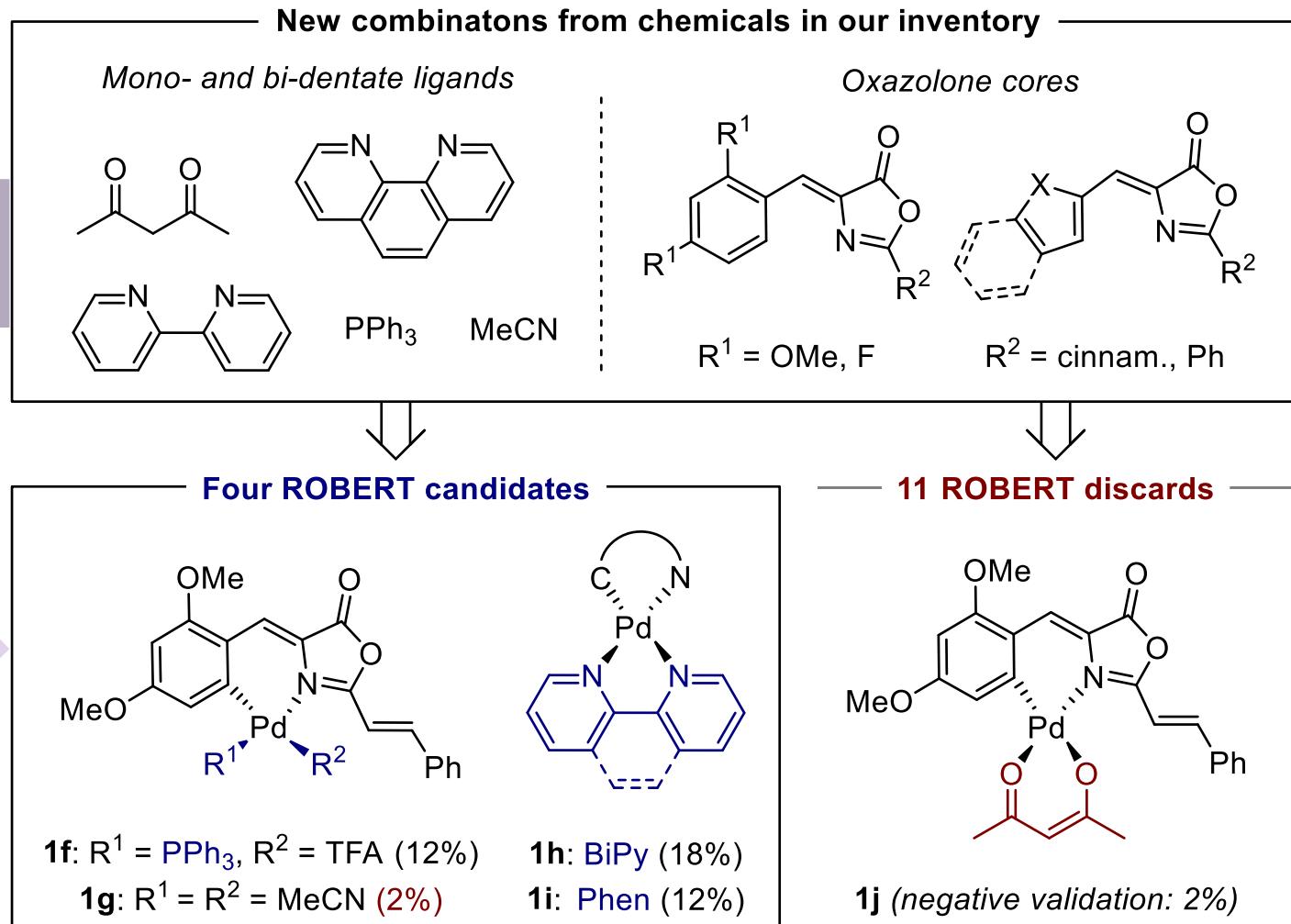
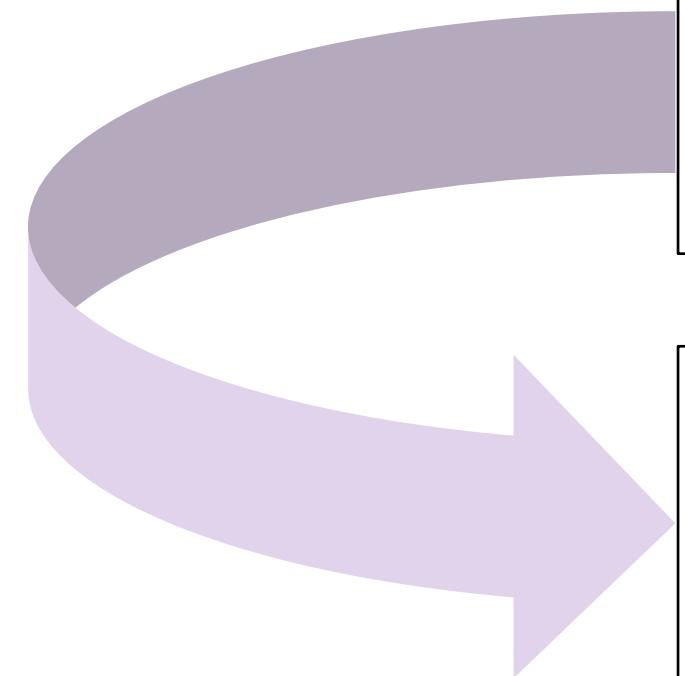
Real example of our lab

- ROBERT for predicting new candidates



Machine learning

- Lab-tested: 80% Accuracy



ROBERT: Bringing Machine Learning Closer to Chemists

David Dalmau Ginesta

06/19/2024

CAMLC24

