What is MAST-ML?

MAST-ML is an open-source Python package designed to broaden and accelerate the use of machine learning in materials science research

MAST-ML:

- Leverages canonical machine learning packages (e.g. scikit-learn) to enable the easy construction and execution of general machine learning analysis pipelines

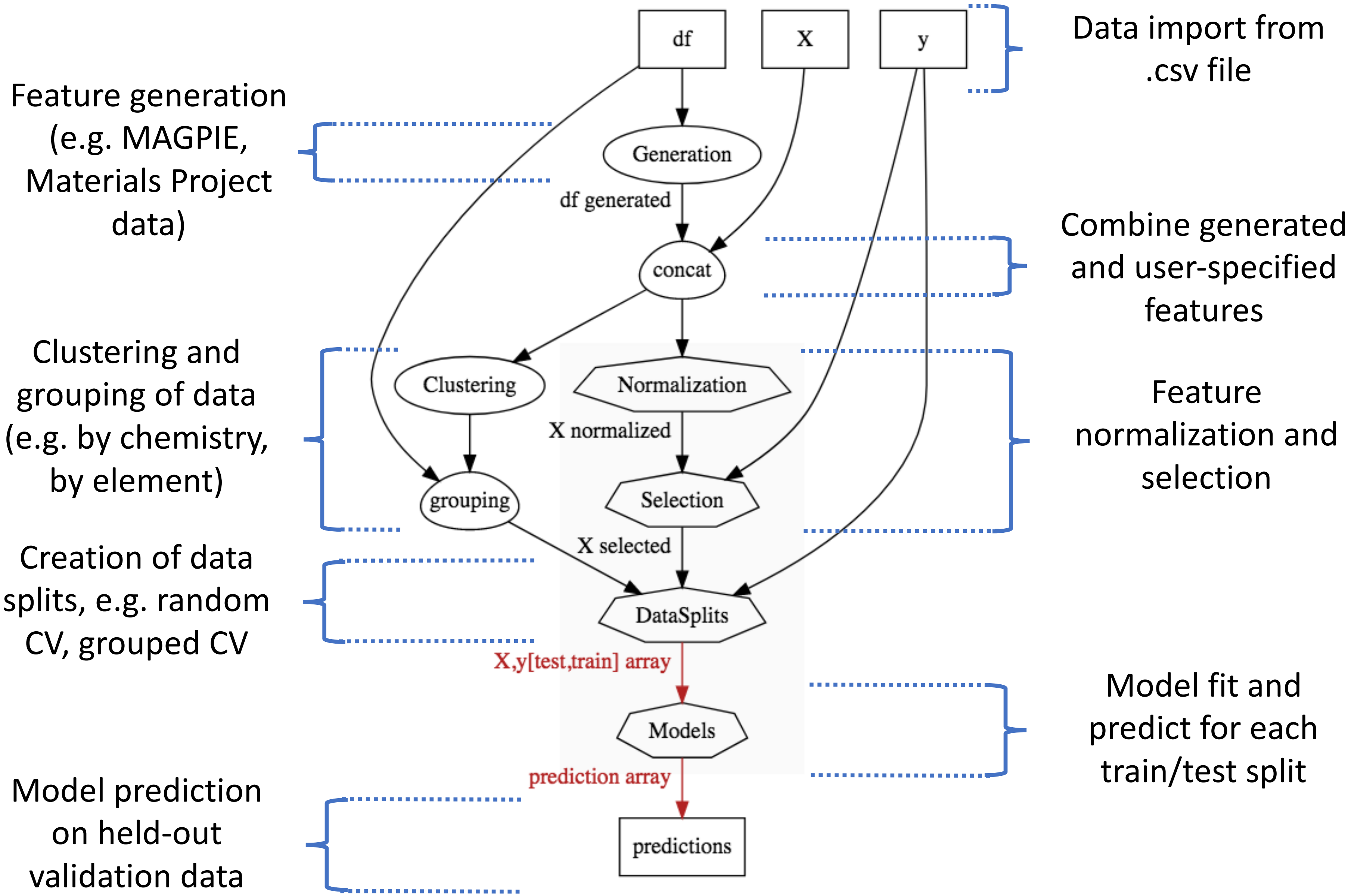
- Codifies best practices of in-depth statistical analysis on user-defined model assessment tests (e.g. leave out group CV)

- Enables data-driven materials research on a faster scale by automating execution and assessment of analysis pipelines, particularly for non-experts

MAST-ML scope and capabilities

- The focus of MAST-ML is currently on supervised learning problems, with emphasis on its application to materials research problems
- MAST-ML supports the full library of scikit-learn modules, and is currently being extended to support tensorflow with Keras
- MAST-ML allows for the simultaneous execution of an arbitrary combination of data preprocessing, feature generation/selection, model types and model evaluation metrics
- MAST-ML is publicly available on GitHub (https://github.com/uw-cmg/MAST-ML) (pull/download master branch)

MAST-ML workflow



MAST-ML sample input

```
[GeneralSetup]
    input_features = Auto
    target_feature = Reduced barrier (eV)
    randomizer = False
    metrics = Auto
    not_input_features = Host element, Solute element, predict_Pt
    validation_column = predict_Pt

[FeatureNormalization]
    [[StandardScaler]]

[DataSplits]
    [[NoSplit]]
    [[RepeatedKFold]]
        n_splits = 5
        n_repeats = 5
    [[LeaveOneGroupOut_host]]
        grouping_column = Host element

[Models]
    [[LinearRegression]]
    [[KernelRidge_5fold]]
        alpha = 0.009
        gamma = 0.027
        kernel = rbf
    [[RandomForestRegressor]]
        criterion = mse
        max_depth = 10
        max_leaf_nodes = 200
        min_samples_leaf = 1
        min_samples_split = 2
        n_estimators = 10
    [[MLPRegressor]]
        #hidden_layer_sizes = 50, 4
        hidden_layer_sizes = 296, 26
        activation = relu
        solver = adam
        alpha = 0.001
        batch_size = 20
        learning_rate = constant

[PlotSettings]
    feature_learning_curve = False
    data_learning_curve = False
    target_histogram = True
    train_test_plots = True
    predicted_vs_true = True
    predicted_vs_true_bars = True
    best_worst_per_point = True
    feature_vs_target = True
```

General setup: names of input and target features, which feature to predict on, etc.

Method to normalize features

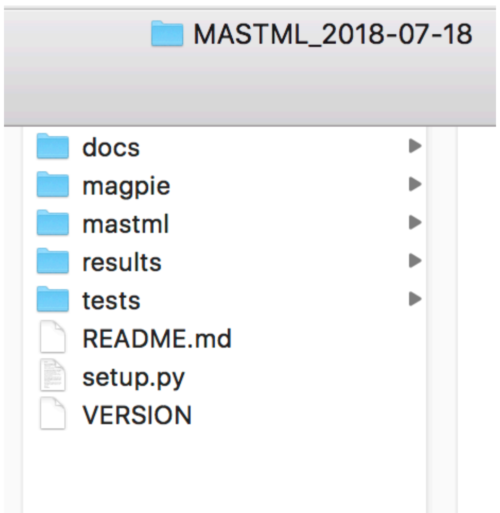
How to split up data for testing, e.g. full fit (“NoSplit”), random CV, leave out group

Which models to test on and their associated parameters. Note that all model and parameter names are the same as in scikit-learn!

Plotting controls: decide what is output

Running MAST-ML

(1) Navigate to your main MAST-ML directory:



(2) In your terminal or IDE, run the command (one line):

python3 -m mastml.mastml_driver ← Call module
tests/conf/example_input.conf ← Path to input
tests/csv/example_data.csv ← Path to data
-o results/example_results ← Path to results

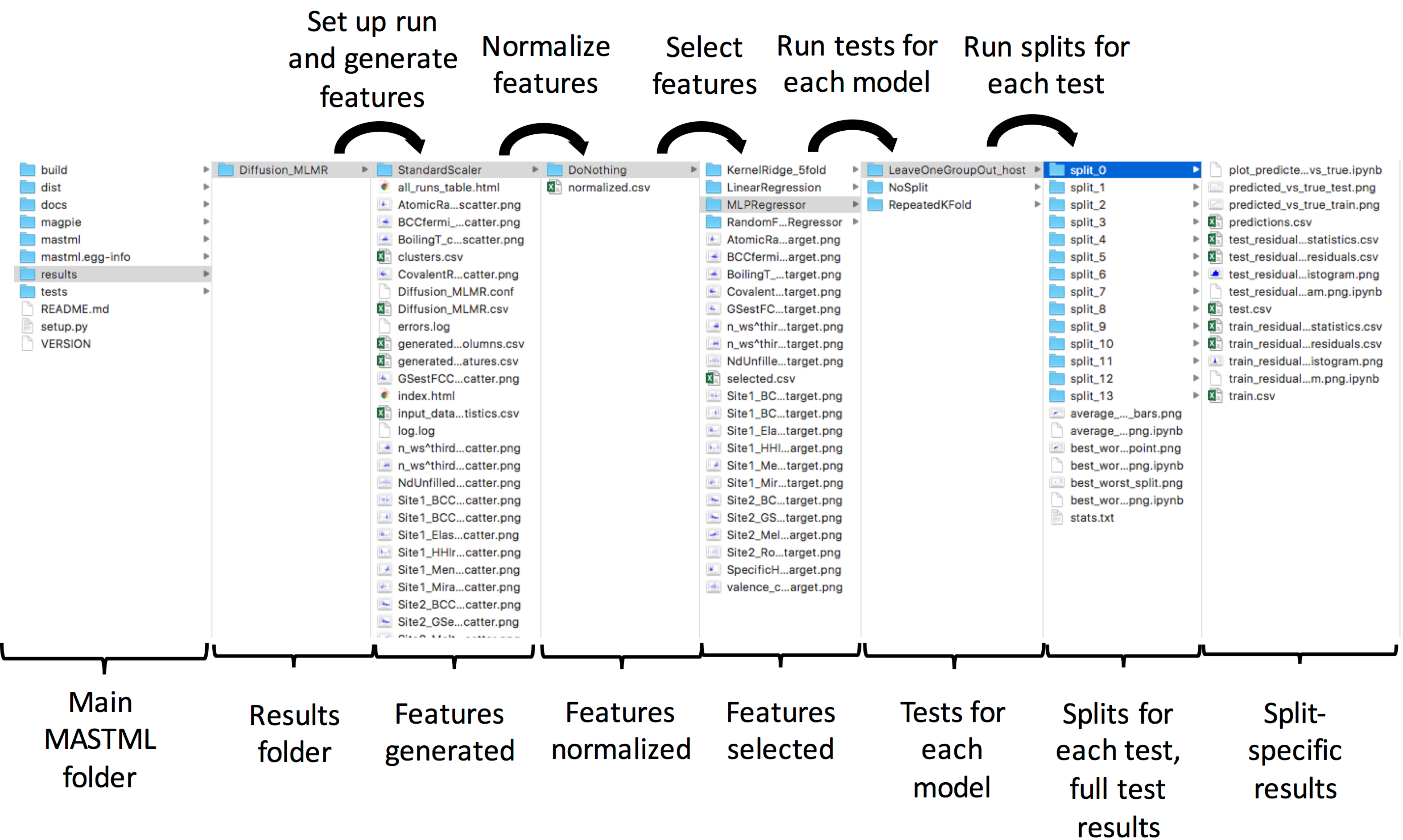
(3) If it's working, you'll start seeing output on your screen:

```
[INFO] 2018-07-26 11:07:55,438 :

MAST-ML run on 2018-07-26 16:07:55 using
conf file: Diffusion_MLMR.conf
csv file:  Diffusion_MLMR.csv
saving to: Diffusion_MLMR_07_26_11_07_55


[INFO] 2018-07-26 11:07:55,438 : Copying input files to output directory...
[INFO] 2018-07-26 11:07:55,461 : blacklisted features, either from "not_input_features" or a "grouping_column":['Host element',
'Solute element', 'predict_Pt']
[DEBUG] 2018-07-26 11:07:56,434 : splitter_to_group_names:
{'LeaveOneGroupOut_host': 'Host element'}
```

MAST-ML high-level output



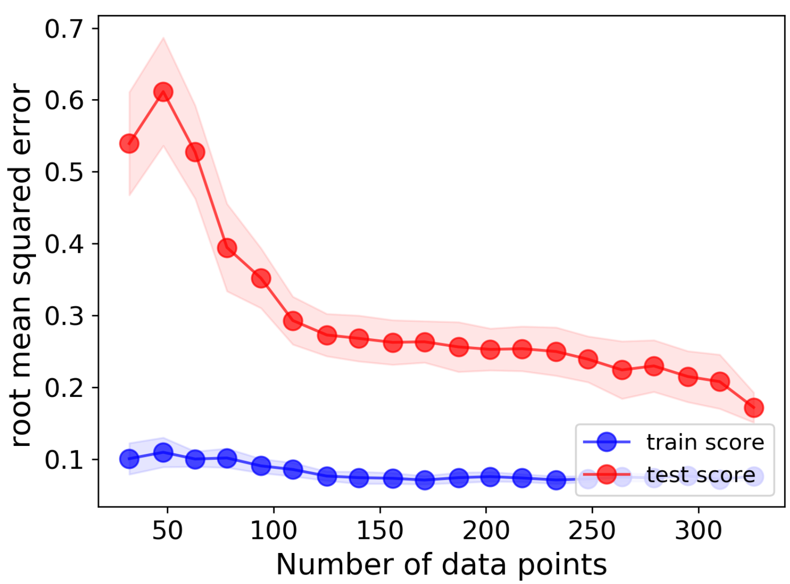
MAST-ML feature generation and selection

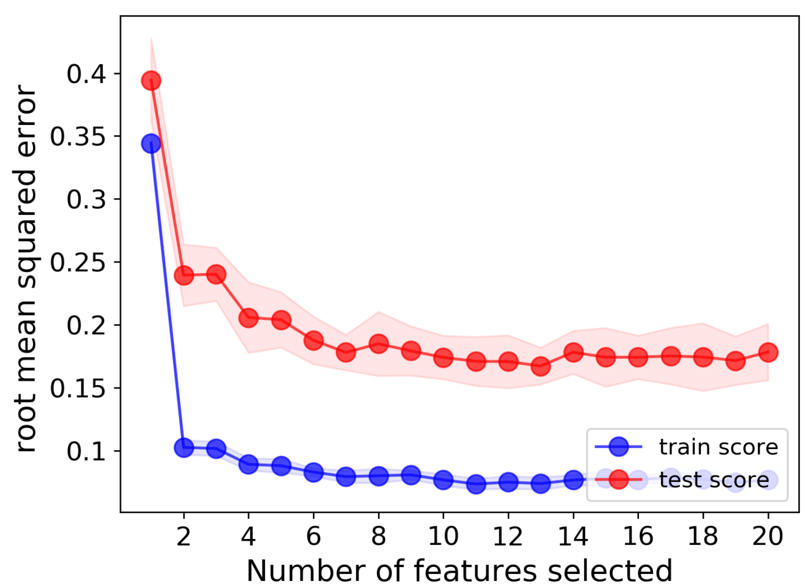
Generation (MAGPIE, Materials Project, Citrination)

100s or 1000s of features…

Host elemen	Reduced barrier (eV)	SecondIonizationEnergy_	ShearModulus_	SpaceGroupNumber_	SpecificHeatCapacity_	ThermalConductivity_	ThermalExpansionCoefficient_	ThirdIonizationEnergy_min_value
Ag	0	21.49	30	225	0.235	429	18.9	34.83
Ag	-0.090141676	21.49	30	225	0.235	429	18.9	34.83
Ag	0.259138544	21.49	30	225	0.235	429	18.9	34.83
Ag	-0.022200405	21.49	30	225	0.235	429	18.9	34.83
Ag	0.317672341	21.49	30	225	0.235	429	18.9	34.83
Ag	0.202185741	21.49	30	225	0.235	429	18.9	34.83
Ag	0.250571478	21.49	30	225	0.235	429	18.9	34.83
Ag	-0.001431337	21.49	30	225	0.235	429	18.9	34.83
Ag	0.164968058	21.49	30	225	0.235	429	18.9	34.83
Ag	0.248163228	21.49	30	225	0.235	429	18.9	34.83
Ag	-0.146976233	21.49	30	225	0.235	429	18.9	34.83
Al	0	18.828	26	225	0.9	237	23.1	28.447
Al	-0.12503	18.828	26	225	0.9	237	23.1	28.447
Al	-0.14243	18.828	26	225	0.9	237	23.1	28.447

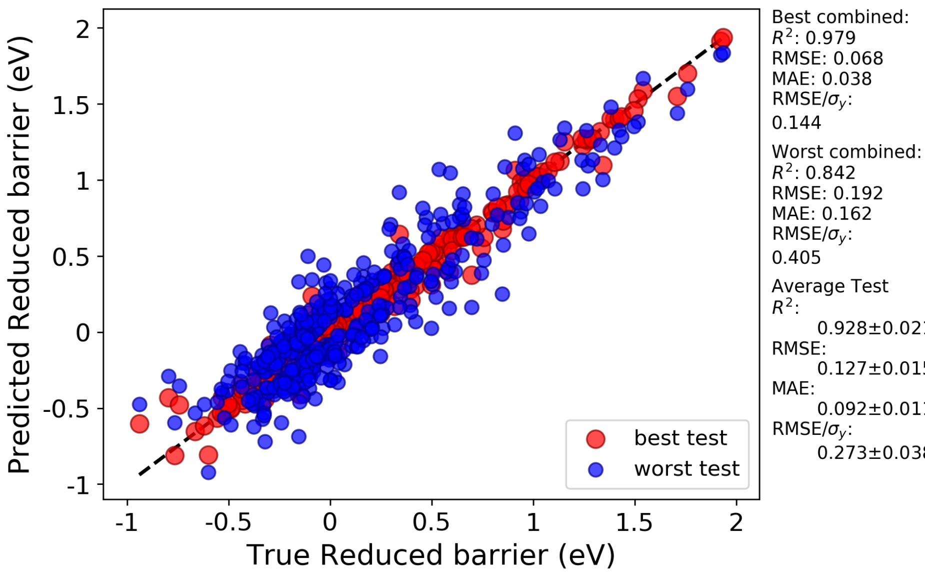
Selection and learning curves (Random Forest on Diffusion data)

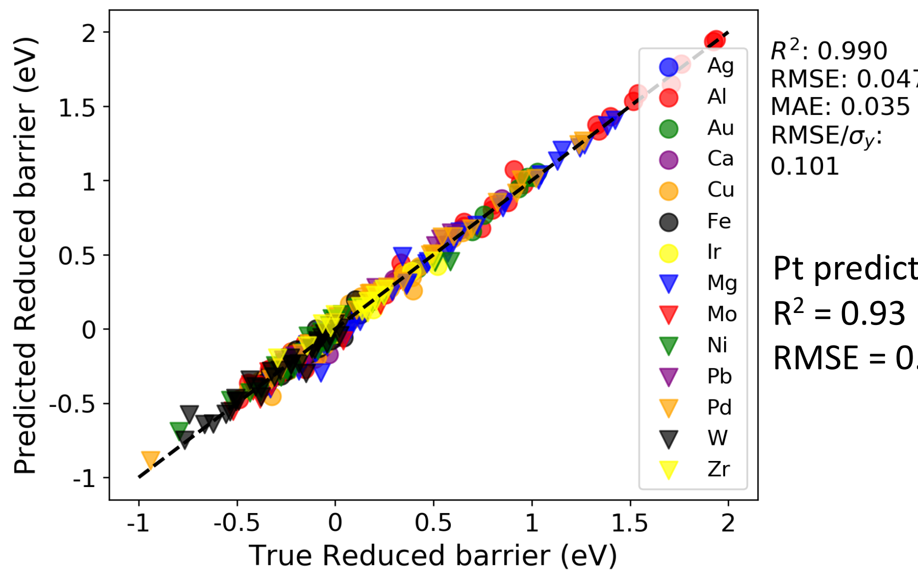




MAST-ML model assessment

- A blizzard of statistics:
  - Output of every train/test split and prediction
  - Averages over every split and error bars for each point
  - Best/worst on per-split and per-point basis
  - Per-group and per-cluster train/test visualization
  - Output as:
    - Spreadsheets
    - Histograms
    - Parity/scatter plots
    - HTML summary file





Pt predictions:

$R^2 = 0.93$

RMSE = 0.188 eV

MAST-ML hyperparameter optimization

- MAST-ML currently supports hyperparameter optimization using grid search and a genetic algorithm (GA).

- Example heat maps of running grid search to optimize the α and γ parameters in a KernelRidge model on the diffusion data set from the work of Wu, *et al.* Comp. Mat. Sci. (2017)