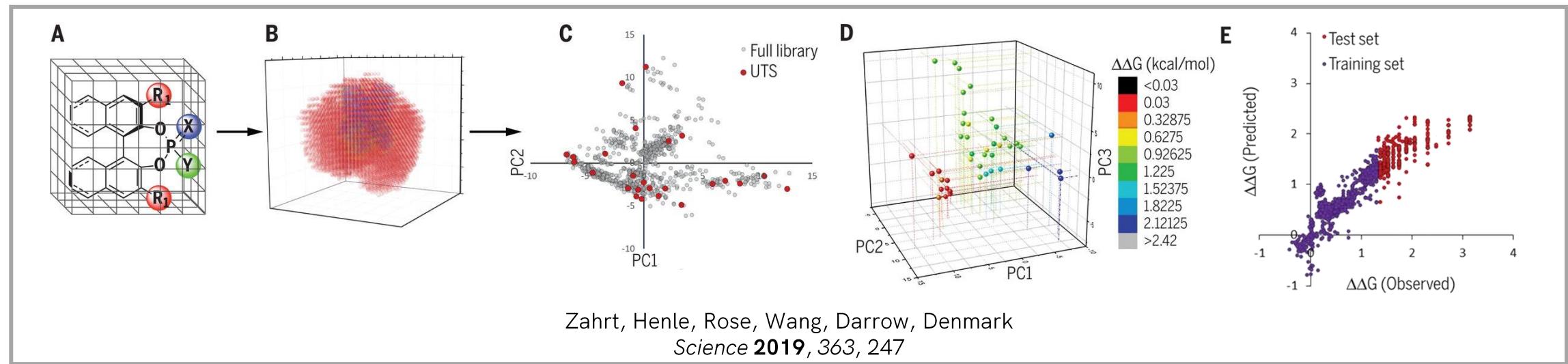


Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning



and
an introduction to ML

Nov 2022
Literature Meeting

What do we need to know about machine learning?

used to train the model. To ensure that model efficacy and training set-test set partitioning were unbiased, this process was repeated 10 times. Models were then developed with support vector machines by using a grid-based optimization of hyperparameters with fivefold cross validation (see supplementary materials for details).

We will not cover:

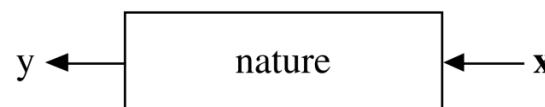
1. A comprehensive survey of ML models
2. How to implement ML models
3. Math underlying ML models

What is machine learning?

The study of algorithms and statistical models to develop computer programs that improve with experience

AI combines machine learning components into a larger system that includes a decision-making component, and therefore *exhibits a behavior*.

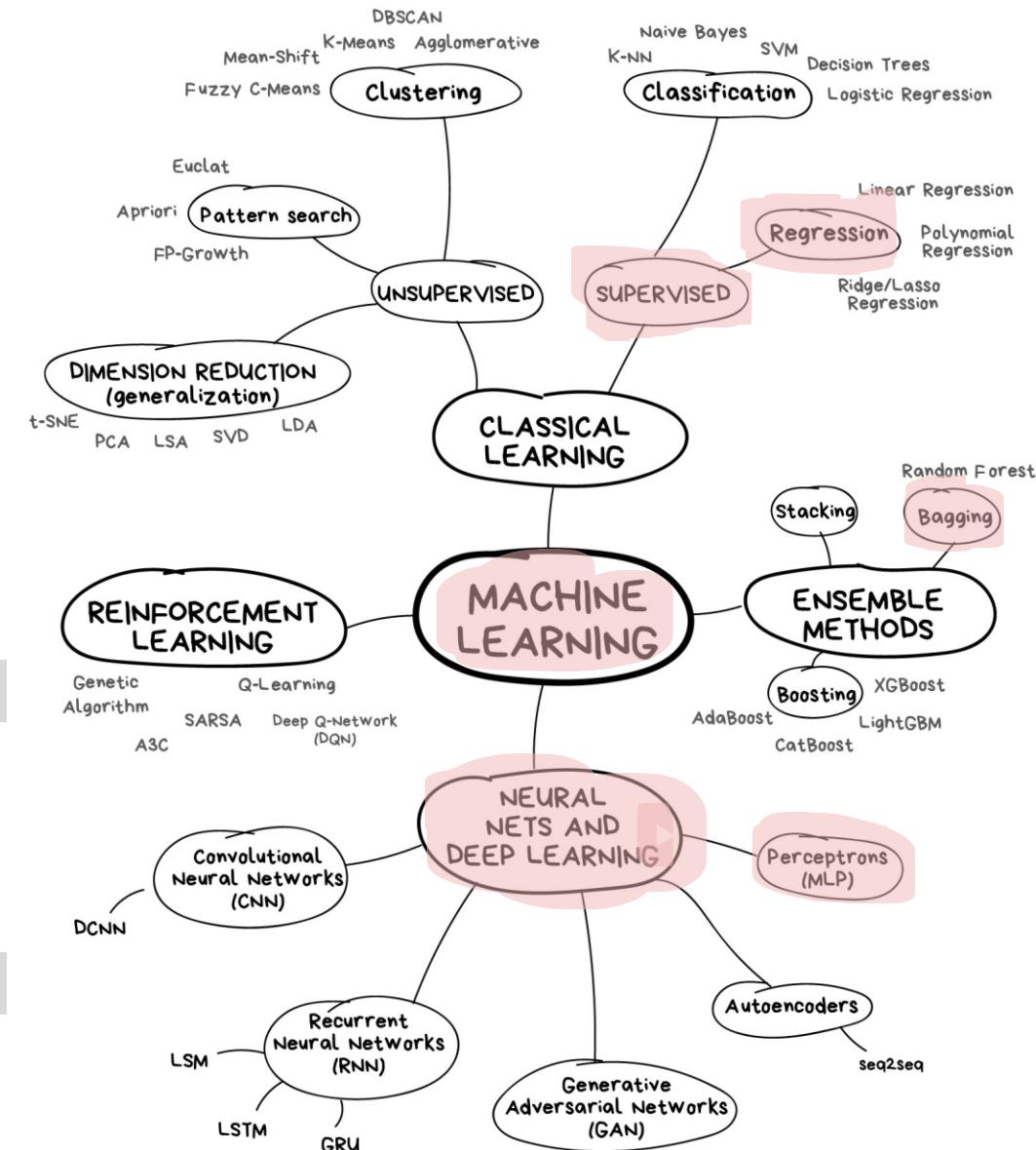
Given some data:



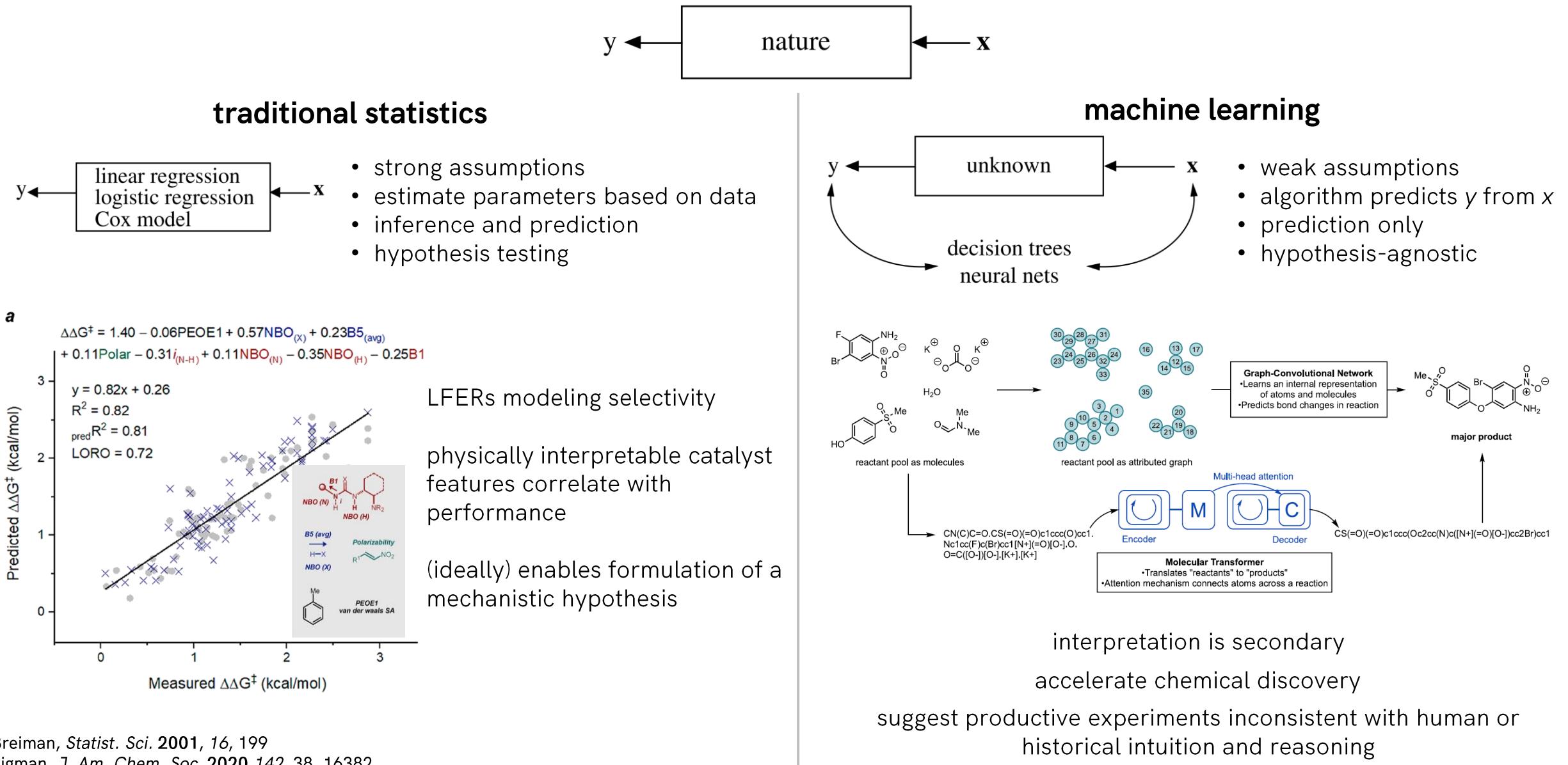
we want to 1. *predict* future y from future x ; 2. *infer* how nature associates y to x

Some terms:

supervised	unsupervised
given a set of (x,y) , learn to predict y using x	given a set of x , learn underlying structure of x
e.g. map reaction conditions to selectivity	e.g. identify common motifs in drugs
regression	classification
model continuous y based on x	model category of y based on x
e.g. map reaction conditions to selectivity	e.g. predicting high ee vs low ee



Breiman's "Two Cultures" of modeling



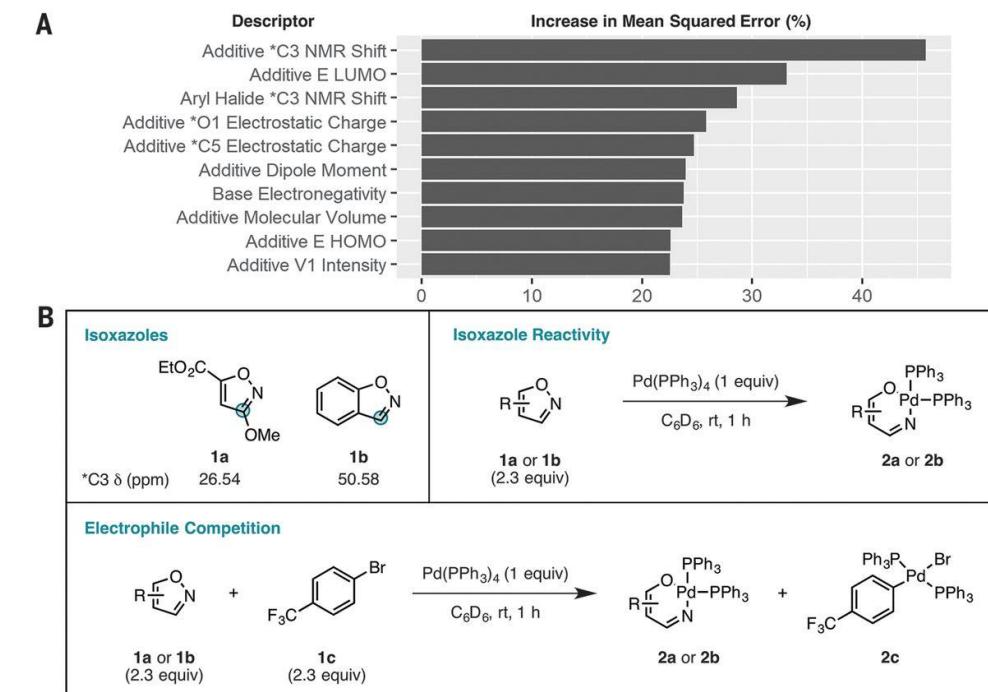
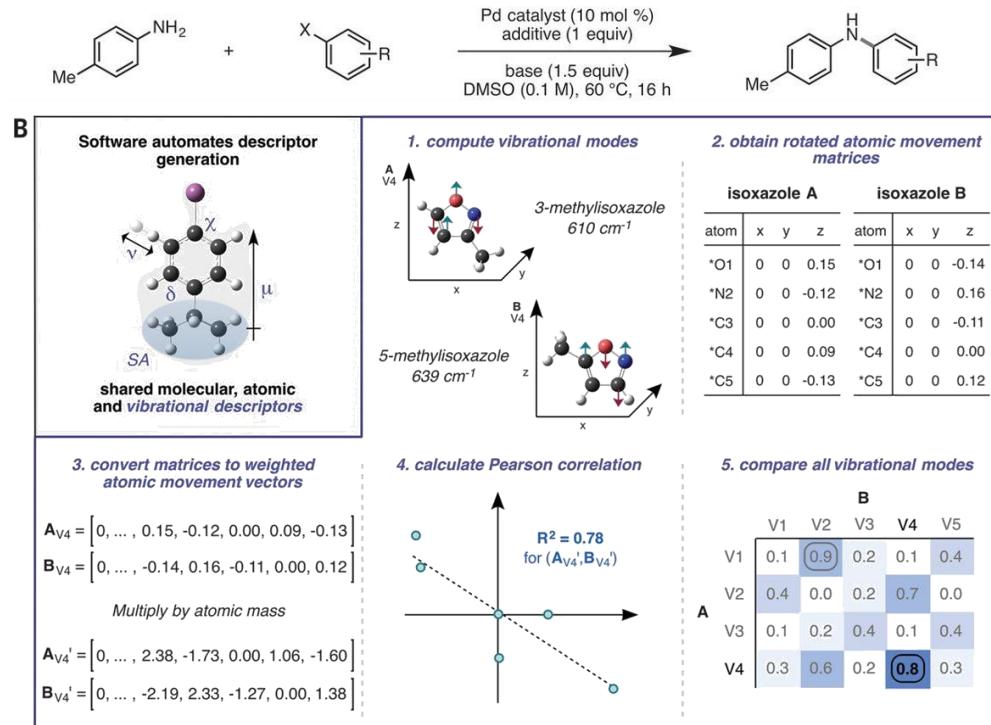
How is ML relevant to us?

Why should we care?

1. High-throughput experimentation and data sharing make large, systematic data sets more available
2. Growing compute power enables i) faster ML model training, and ii) faster computation of chemical properties
3. Modeling methods (e.g. DFT) have advanced significantly, enabling generation of higher-quality datasets
4. ML models are becoming more sophisticated yet more accessible
5. ML is ubiquitous in our lives!

1. Revealing patterns where few or none are known

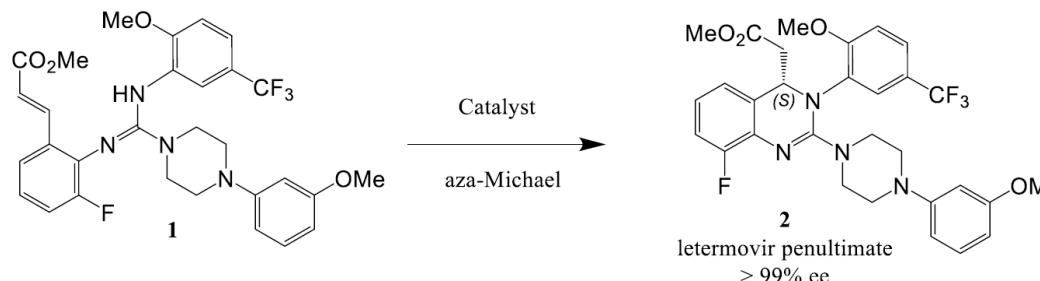
Random forest models predict yields of B-H reactions and inform mechanistic hypotheses for a side reaction



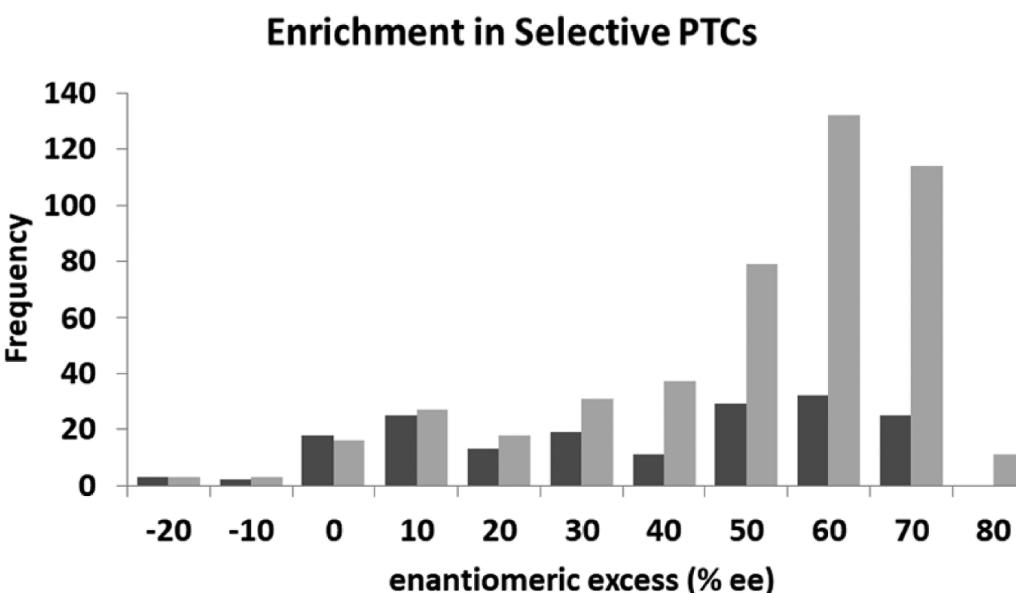
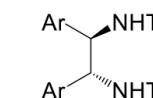
How is ML relevant to us?

2. Overcoming the limits of simple models and human experience

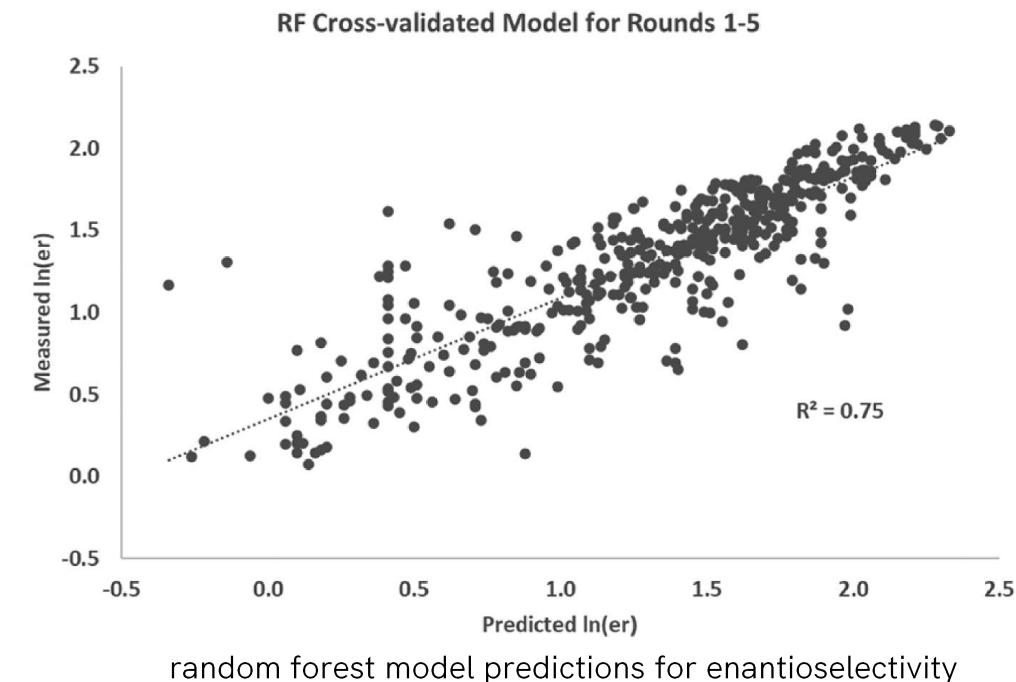
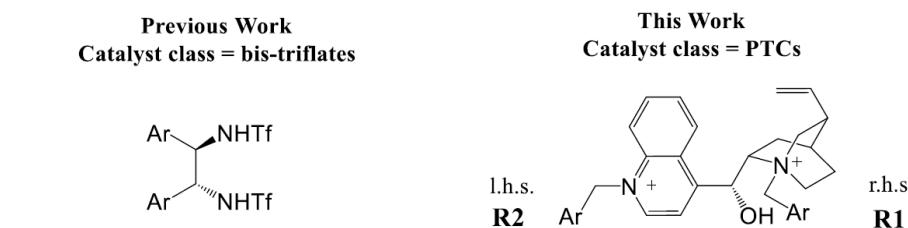
Iterative ML/reaction optimization improves selectivity in phase-transfer-catalyst-promoted conjugate addition from 76% to 89% ee



Previous Work
Catalyst class = bis-triflates



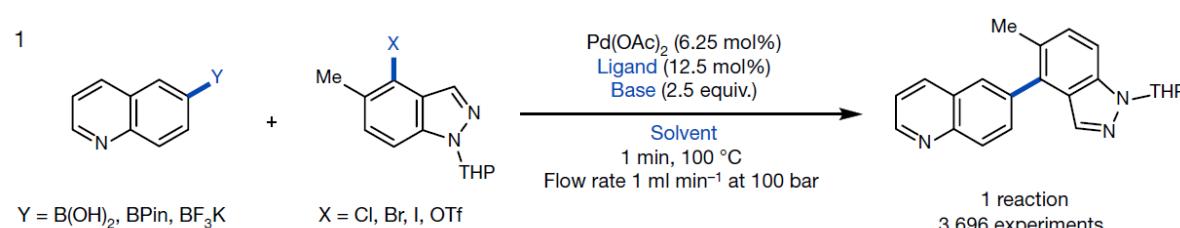
% ee distribution in optimization round 1 (dark) and after rounds 1–5 (light)



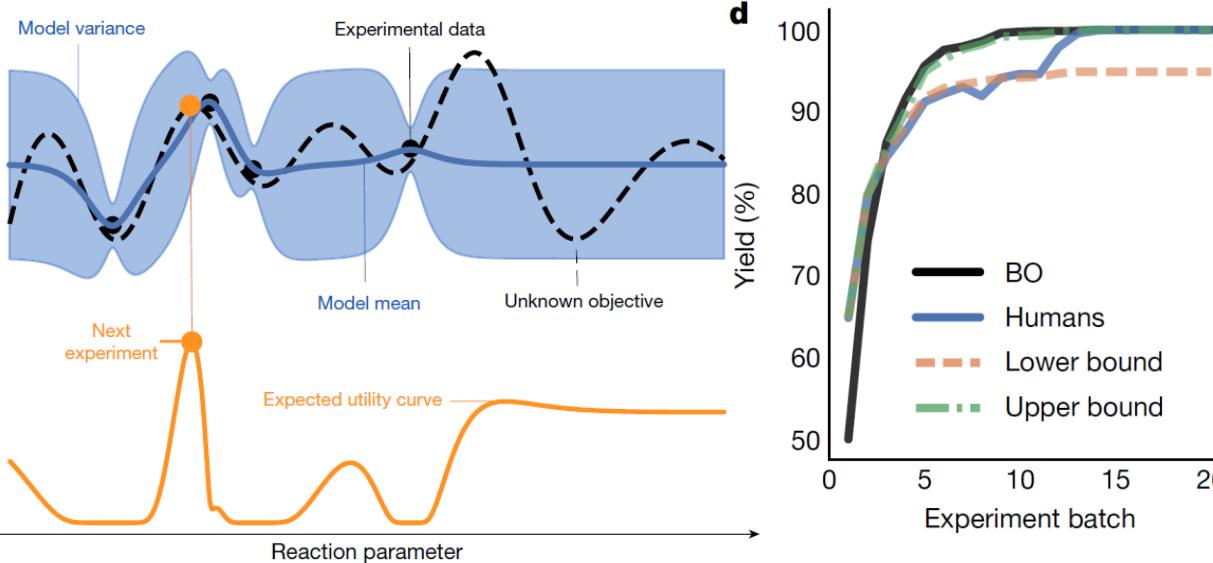
How is ML relevant to us?

3. Accelerating computations and analysis to enable rapid discovery

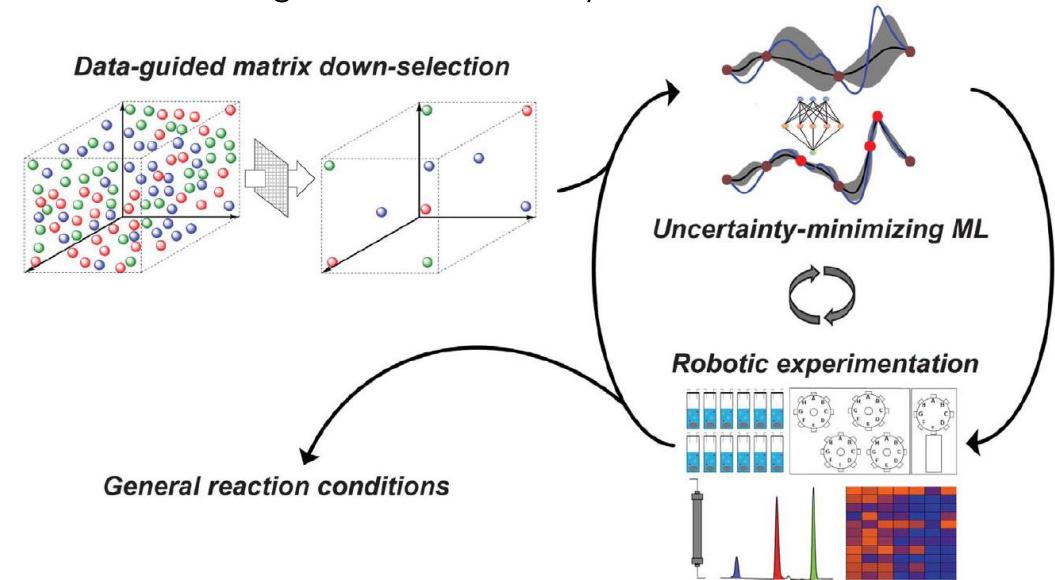
Bayesian optimization (BO) helps select the next experiments that reduce model uncertainty the most



c Bayesian optimization: one-dimensional visualization



Uncertainty-minimizing ML with robotic synthesis enables discovery of general Suzuki-Miyaura XC conditions

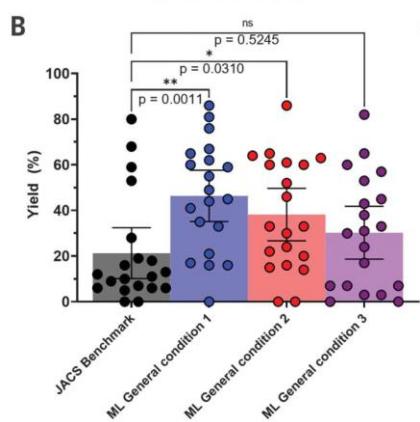


Denoting the set of possible reaction conditions as $C = \{c\}$, a set of substrate pairs as $S = \{s\}$, and reaction yield as $y(s, c)$, our aim is to maximize the objective function given by

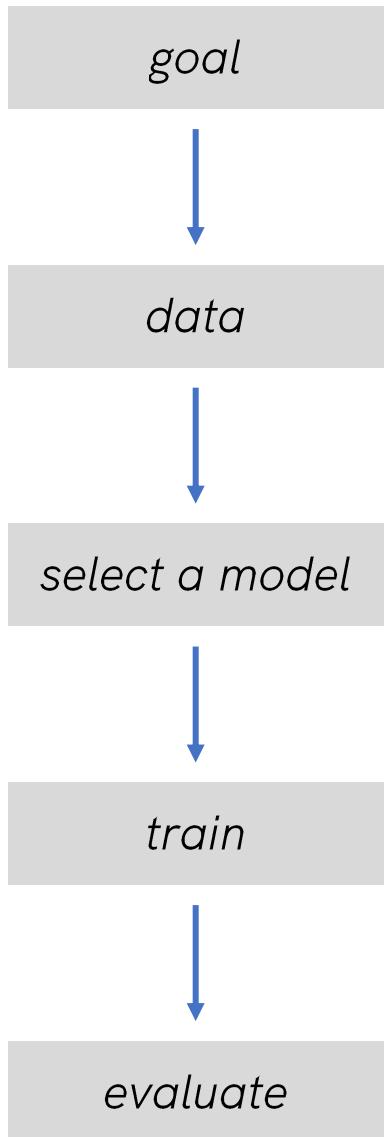
$$f(c) = \frac{1}{|S|} \sum_{s \in S} y(s, c) \quad (1)$$

Then, the general conditions c_{general} are given as

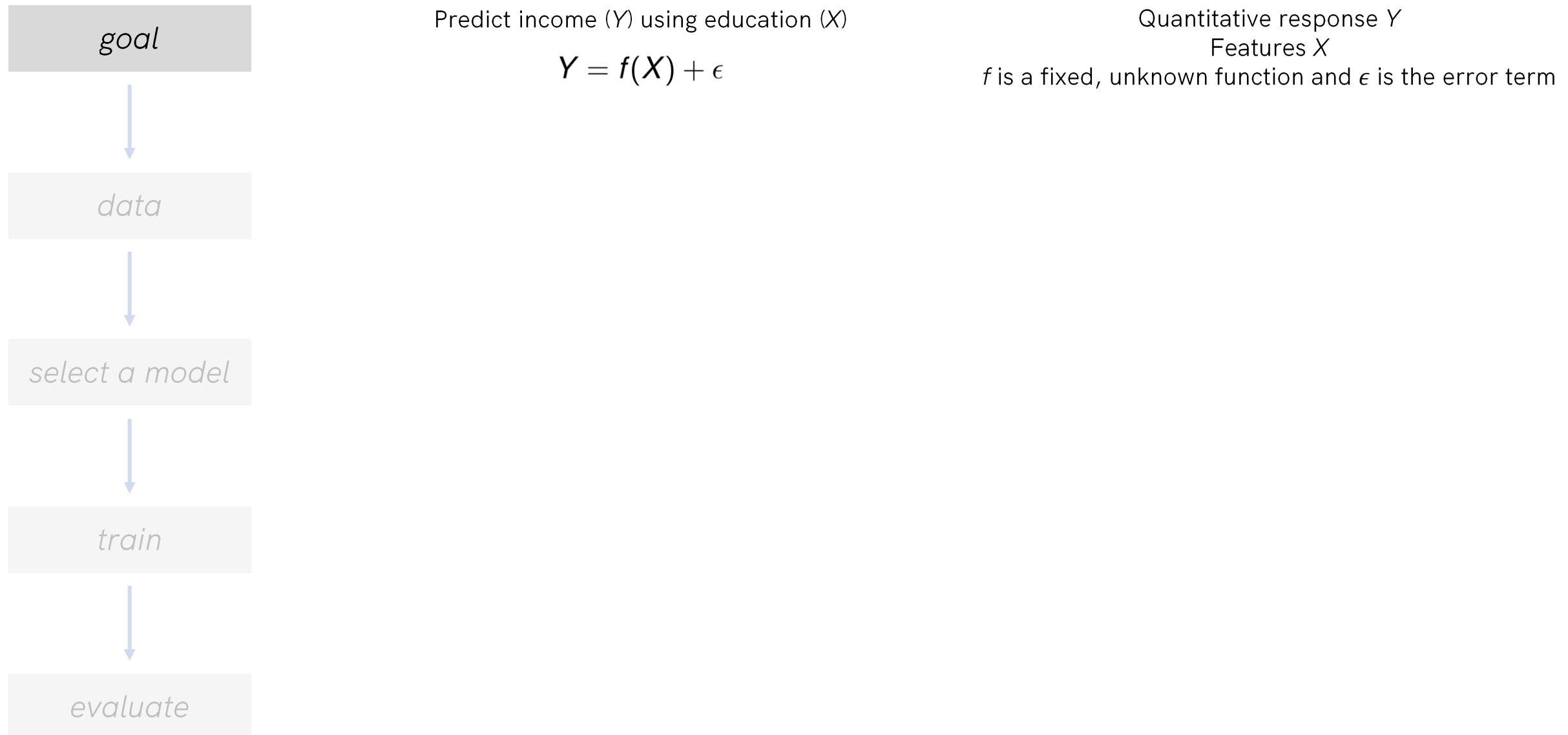
$$c_{\text{general}} = \arg \max_{c \in C} f(c) \quad (2)$$



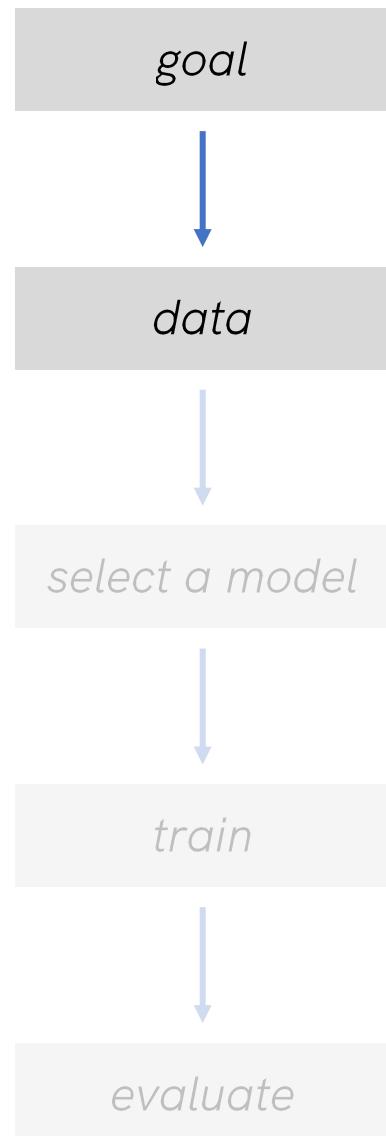
Back to basics: developing a simple regression model



Back to basics: developing a simple regression model

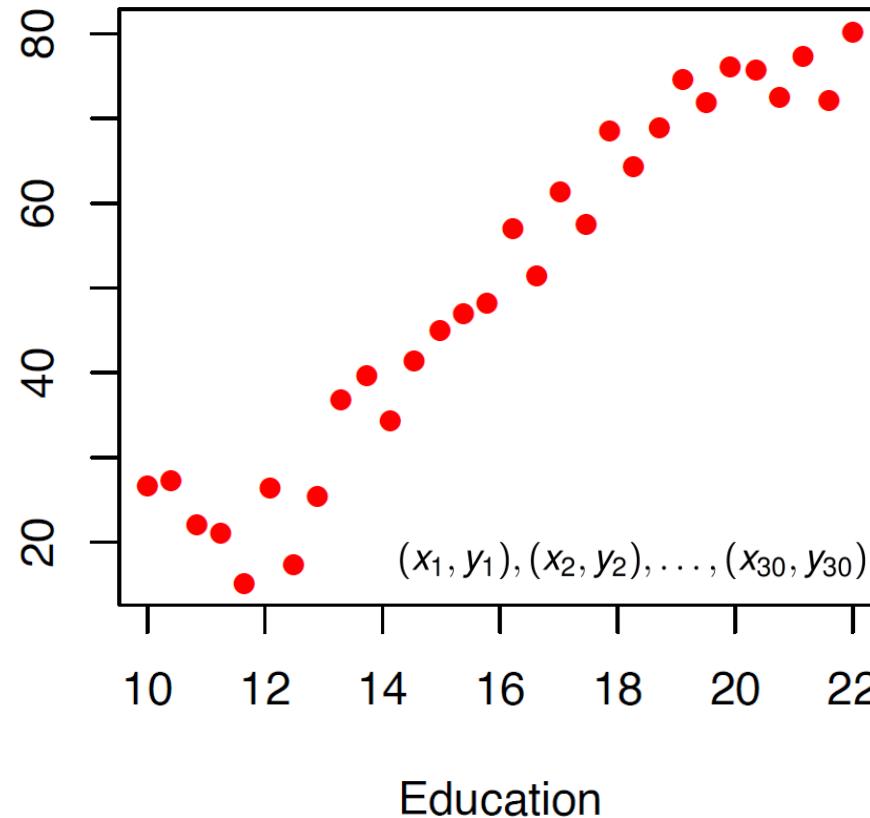


Back to basics: developing a simple regression model

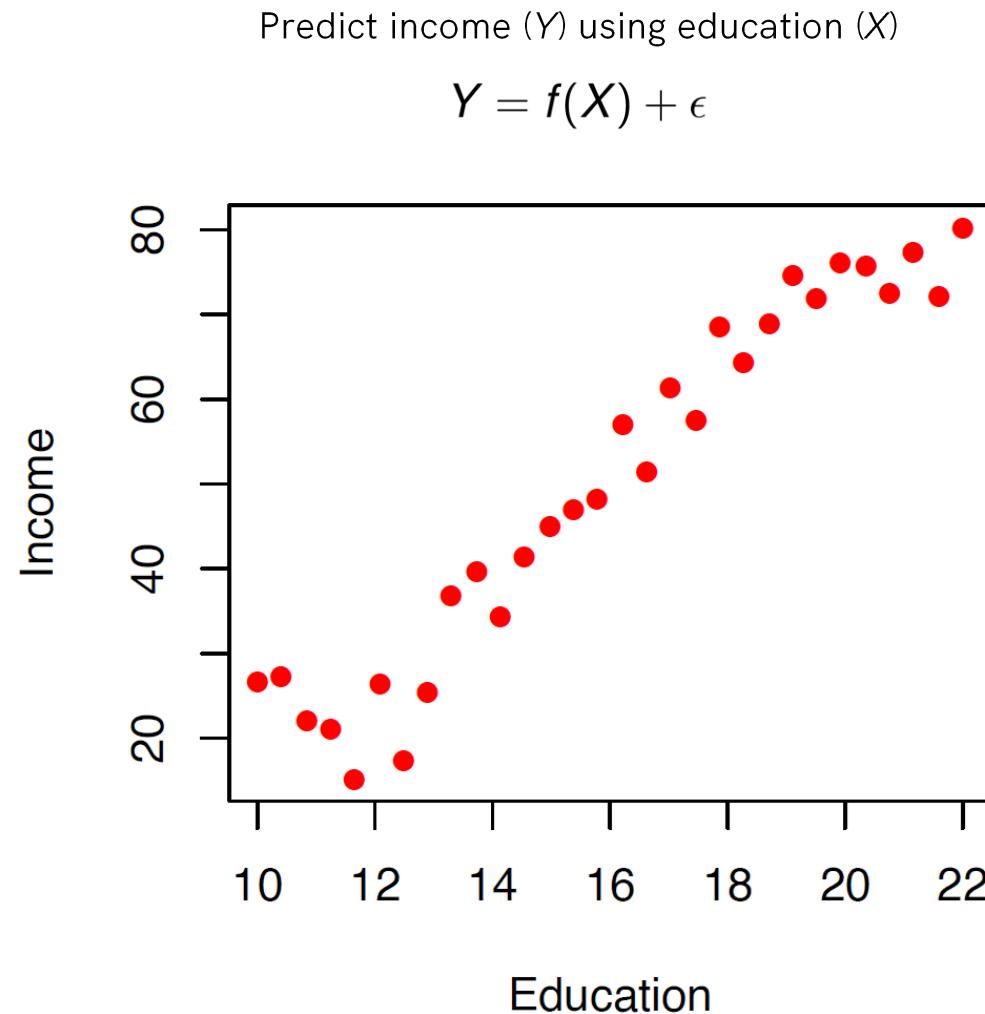
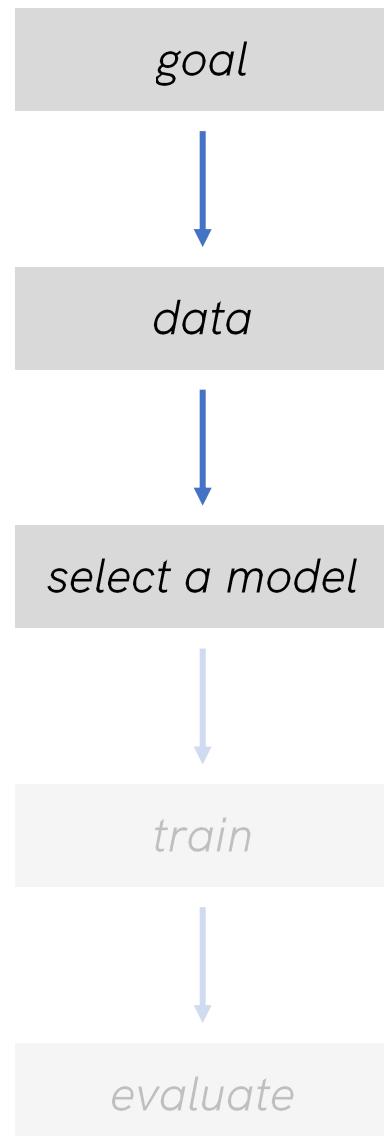


Predict income (Y) using education (X)

$$Y = f(X) + \epsilon$$



Back to basics: developing a simple regression model



Linear model

$$f(X) = \beta_0 + \beta_1 X \quad \epsilon \sim N(0, \sigma^2)$$

find **parameters** $\hat{\beta}_0$ and $\hat{\beta}_1$ such that $\hat{Y} = \hat{f}(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ is reasonably close to Y .

For any $\hat{\beta}_0$ and $\hat{\beta}_1$, we predict $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

The **residual** $e_i = y_i - \hat{y}_i$ is the difference between the i -th observed value and its fitted value.

Back to basics: developing a simple regression model

goal



data



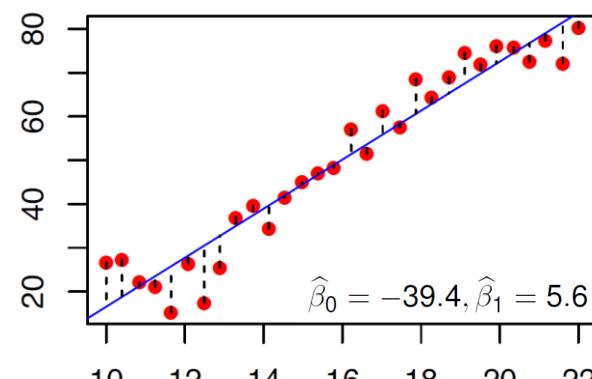
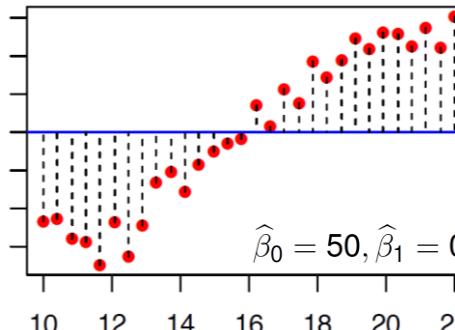
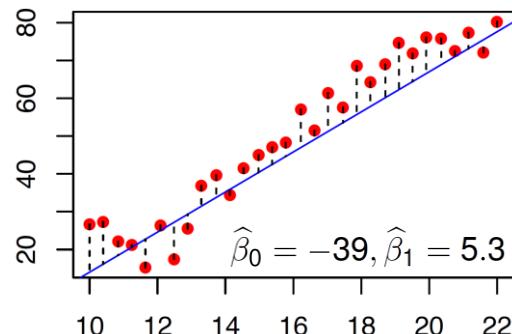
select a model



train



evaluate



$$\widehat{\text{Income}} = -39.45 + 5.60 \cdot \text{Education}$$

Linear model

$$f(X) = \beta_0 + \beta_1 X \quad \epsilon \sim N(0, \sigma^2)$$

find **parameters** $\hat{\beta}_0$ and $\hat{\beta}_1$ such that $\hat{Y} = \hat{f}(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ is reasonably close to Y .

For any $\hat{\beta}_0$ and $\hat{\beta}_1$, we predict $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

The **residual** $e_i = y_i - \hat{y}_i$ is the difference between the i -th observed value and its fitted value.

We aim to minimize the **objective function**, which measures how well model predictions match the data ("ground truth").

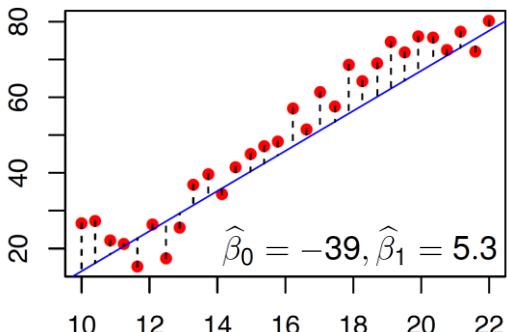
In least squares regression, we want to find model parameters $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize the residual sum of squares (RSS).

objective function/loss function

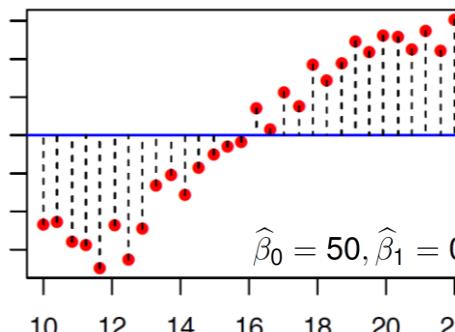
$$RSS(\beta_0, \beta_1) = \sum_{i=1}^n e_i^2 = (y_1 - \beta_0 - \beta_1 x_1)^2 + \cdots + (y_n - \beta_0 - \beta_1 x_n)^2$$

Back to basics: developing a simple regression model

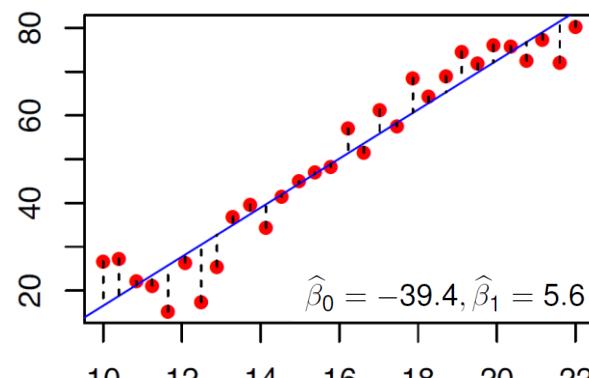
goal



data



select a model



train

$$\widehat{\text{Income}} = -39.45 + 5.60 \cdot \text{Education}$$

evaluate

$$\underbrace{\sum (y_i - \bar{y})^2}_{\text{total sum of squares (TSS)}} = \underbrace{\sum (\hat{y}_i - \bar{y})^2}_{\text{explained sum of squares (ESS)}} + \underbrace{\sum (y_i - \hat{y}_i)^2}_{\text{residual sum of squares (RSS)}}$$

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

R^2 : proportion of variability in y explained by the model

Linear model

$$f(X) = \beta_0 + \beta_1 X \quad \epsilon \sim N(0, \sigma^2)$$

find **parameters** $\hat{\beta}_0$ and $\hat{\beta}_1$ such that $\hat{Y} = \hat{f}(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ is reasonably close to Y .

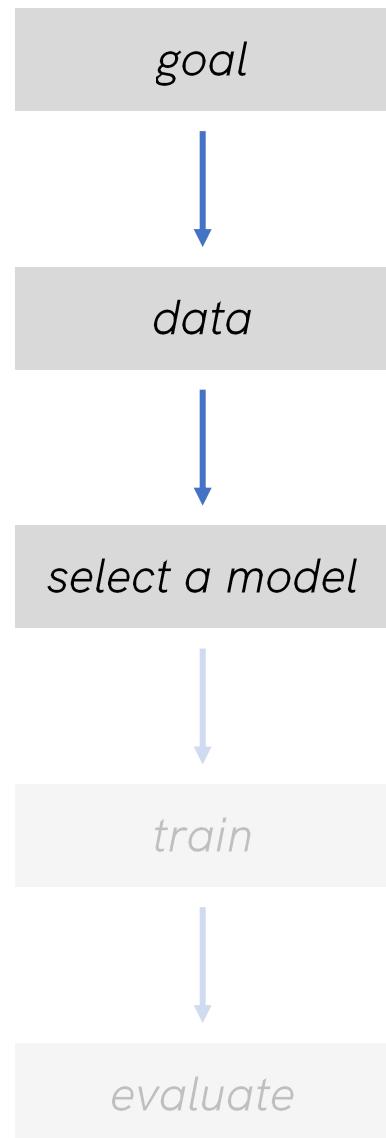
For any $\hat{\beta}_0$ and $\hat{\beta}_1$, we predict $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

The **residual** $e_i = y_i - \hat{y}_i$ is the difference between the i -th observed value and its fitted value.

We aim to minimize the **objective function**, which measures how well model predictions match the data ("ground truth").

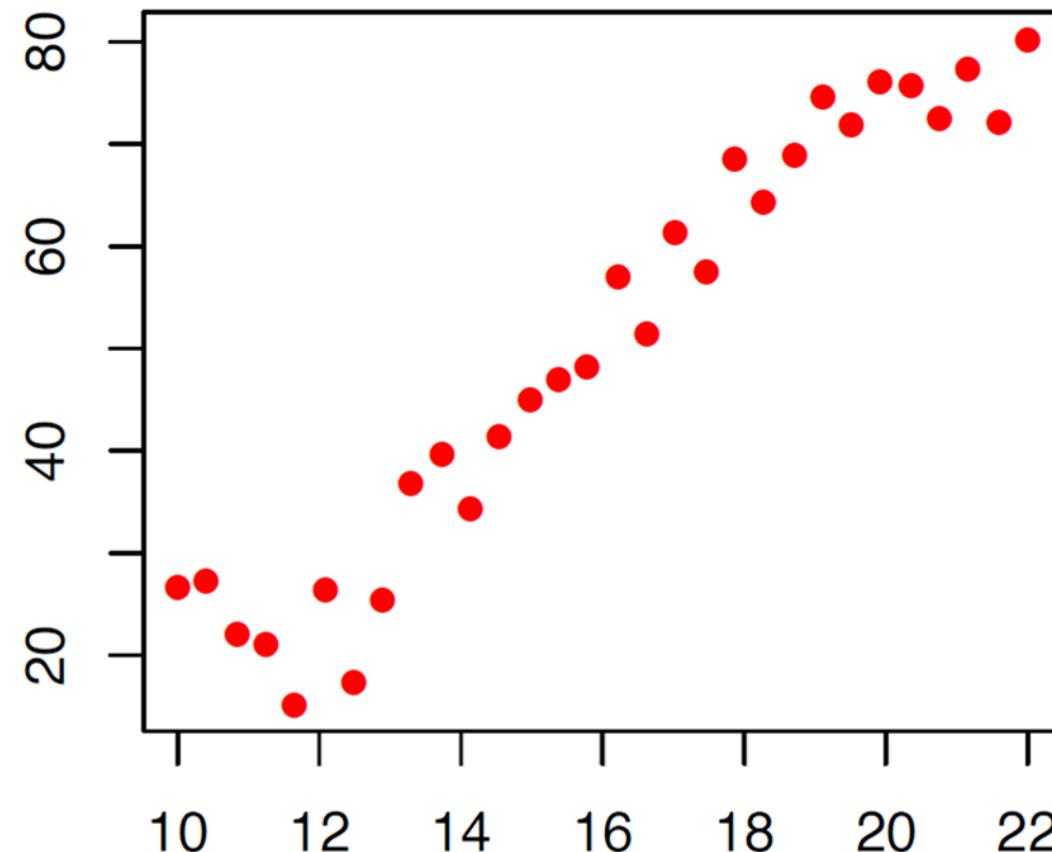
In least squares regression, we want to find model parameters $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize the residual sum of squares (RSS).

Back to basics: developing a simple regression model



Predict income (Y) using education (X)

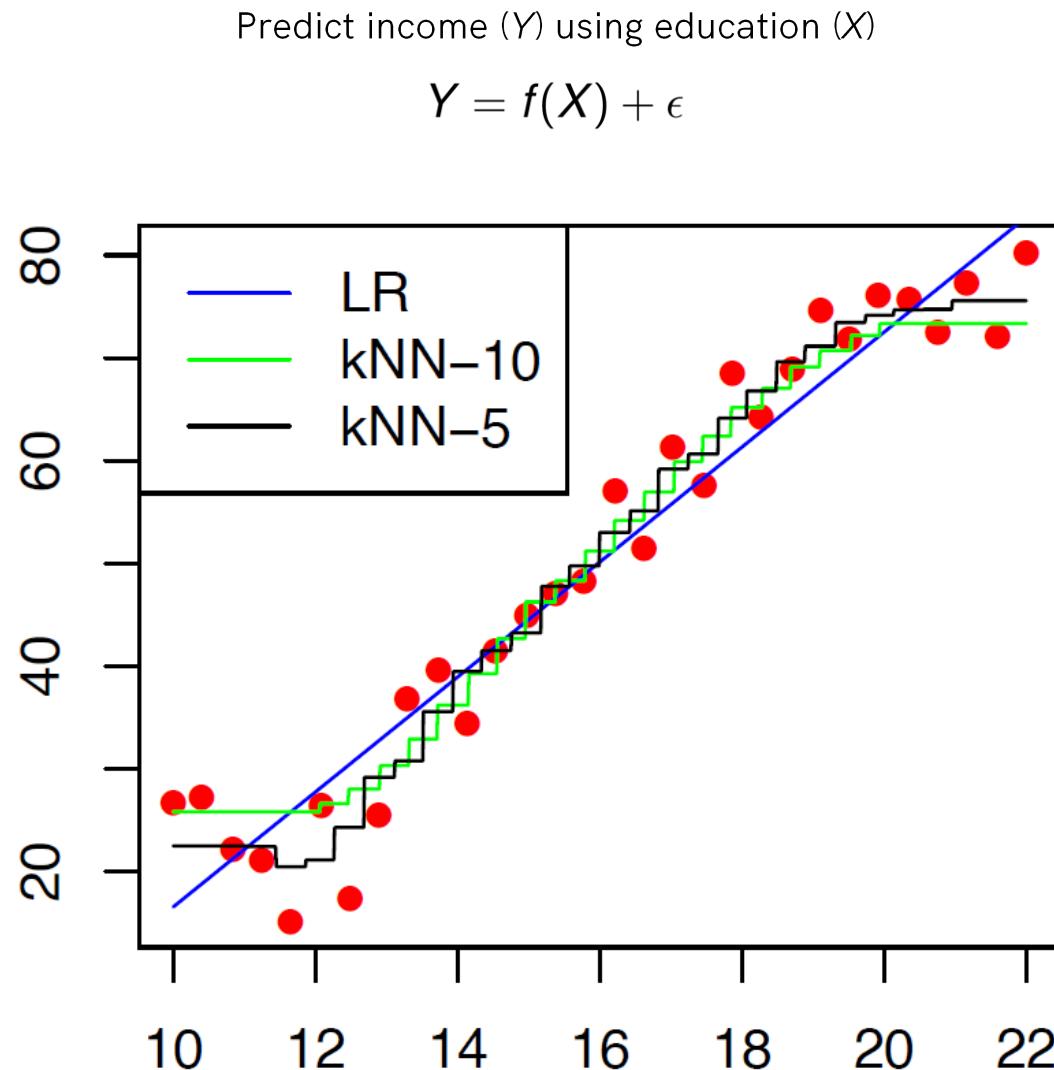
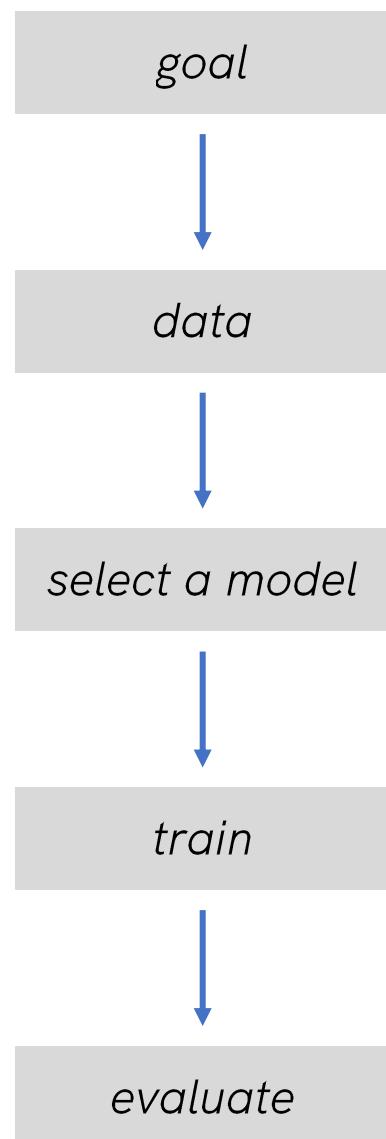
$$Y = f(X) + \epsilon$$



k-nearest neighbors regression

Average together the y_i for the k -closest x_i to x

Back to basics: developing a simple regression model



k-nearest neighbors regression

Average together the y_i for the k -closest x_i to x

Measure performance using Mean Squared Error:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2$$

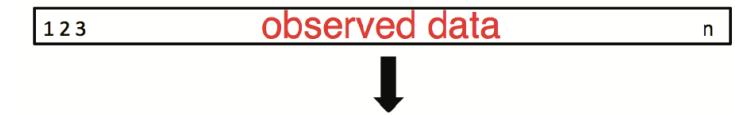
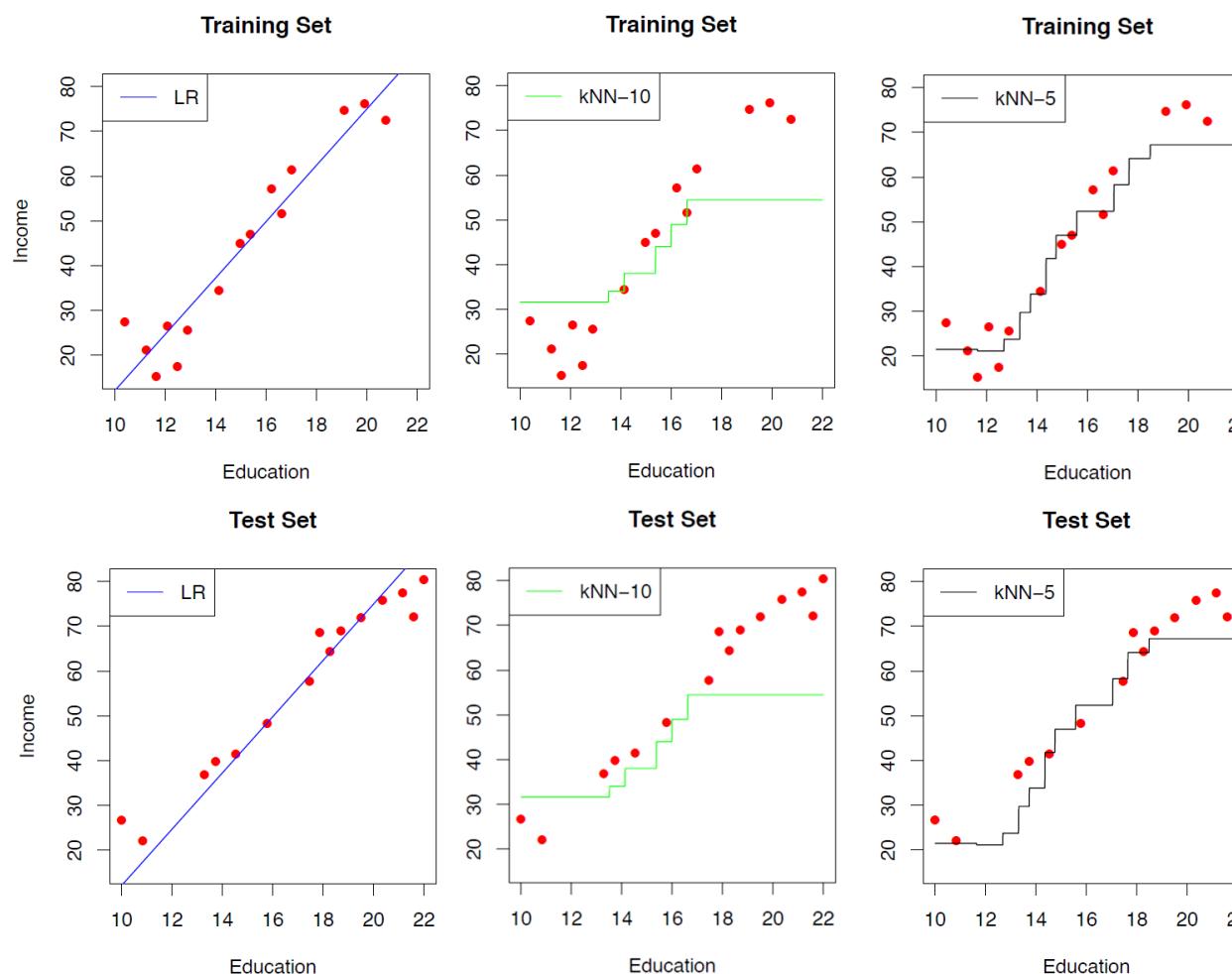
model	MSE
LR	29.829
kNN-10	23.519
kNN-5	16.210

kNN-5 achieves lowest error—is it the best among the three?

Training vs. Test Loss

MSE was calculated based on data used to fit the model (**training loss**), but we are more interested in the MSE for *unseen data* (**test loss**)!

We randomly split our data into a training set and a test set.



Test MSE

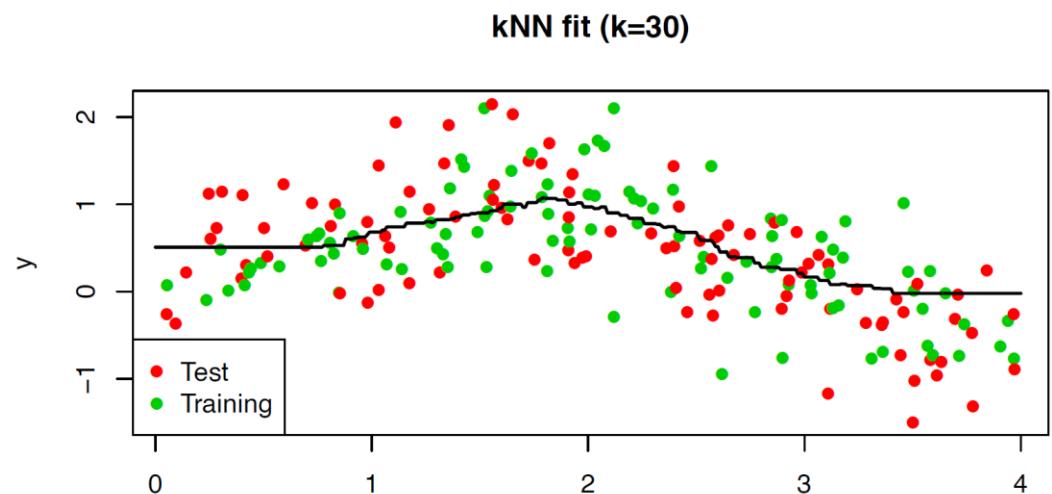
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2$$

model	training MSE	test MSE
LR	27.4	37.9
kNN-10	21.2	198
kNN-5	15.3	48.7

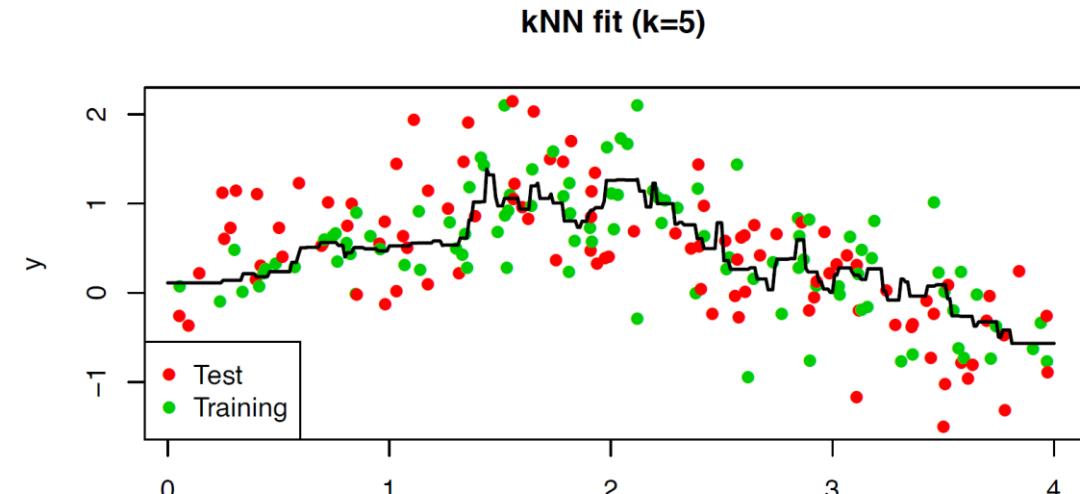
So it appears that LR wins, but does it?

kNN-5 has a small training loss but a *large test loss*.
This means it is **overfitting** the data.

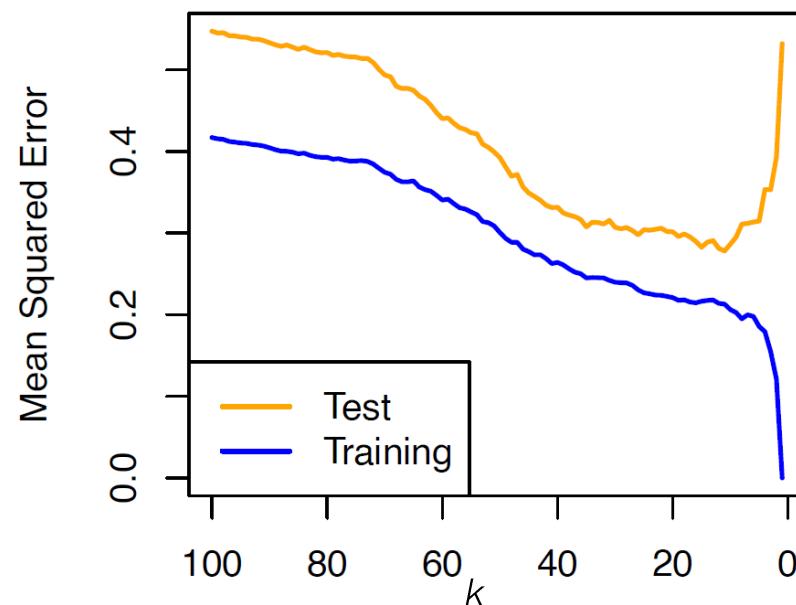
Overfitting reflects the bias–variance tradeoff



A less flexible model fits worse to training data, but is more generalizable to unseen data



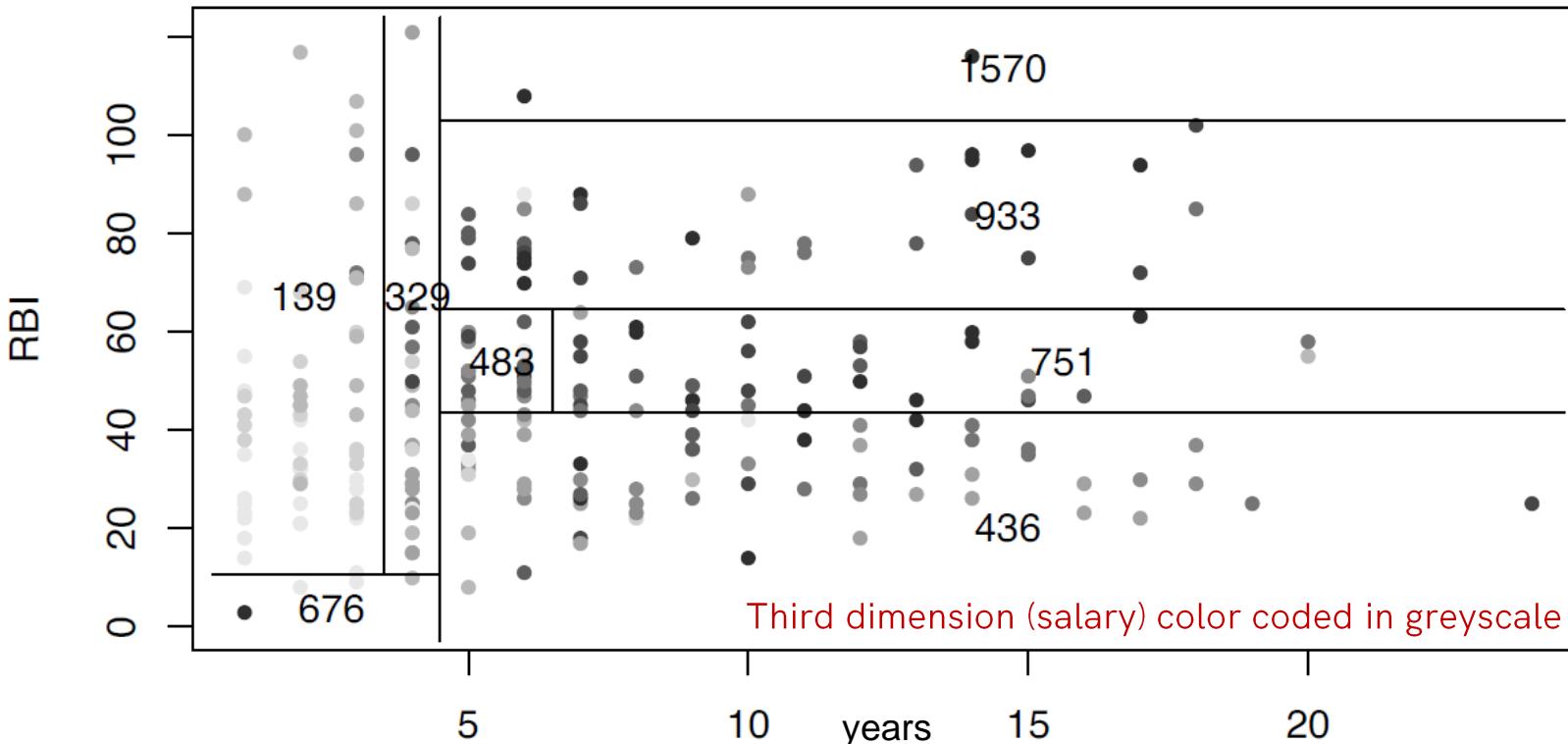
A more flexible model fits better to training data, but is less generalizable to unseen data



There is an optimal k that balances accuracy (bias) and generalizability (variance).

Decision trees enable classification and regression tasks

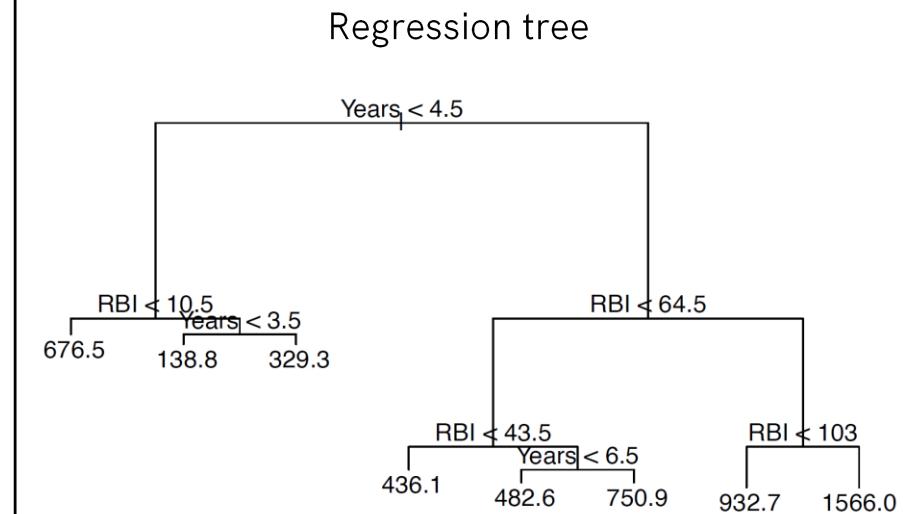
Baseball hitter salaries predicted by **career years** and **RBI**



To build a tree, iteratively find the line that splits the data into two homogeneous (by Y) portions.

To predict, trace each test observation into a leaf.

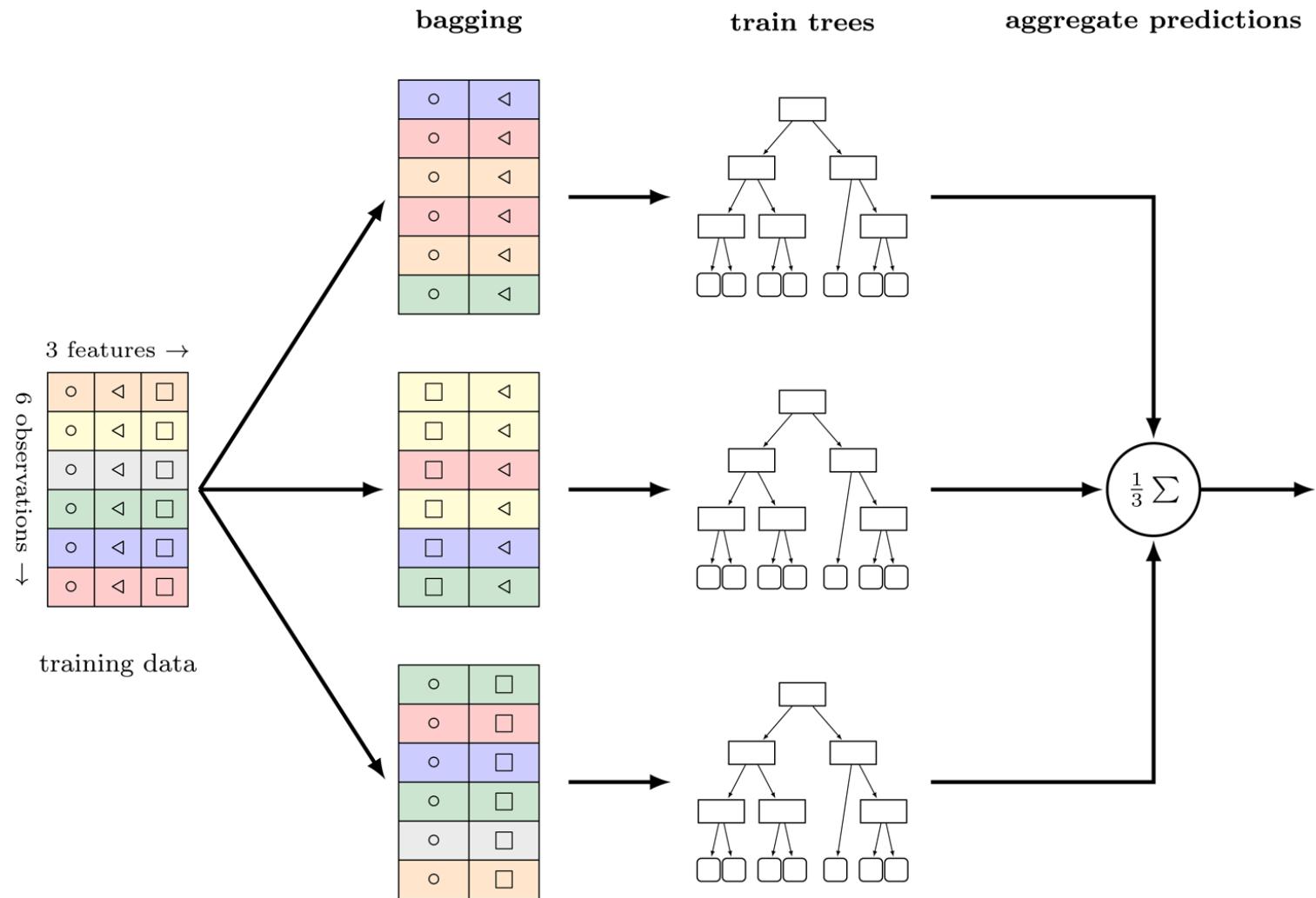
The output is the average of values in that rectangle.



How fine-grained do we want to get?

Tree depth is a **hyperparameter**, i.e. a tunable knob that affects tree performance and therefore can also be optimized

Random forests are made of trees



RF is an ensemble of trees, where each tree is trained on a different random subset of features.

This increases both accuracy and generalizability!

Each tree is cheap, so hundreds to thousands of trees is typical.

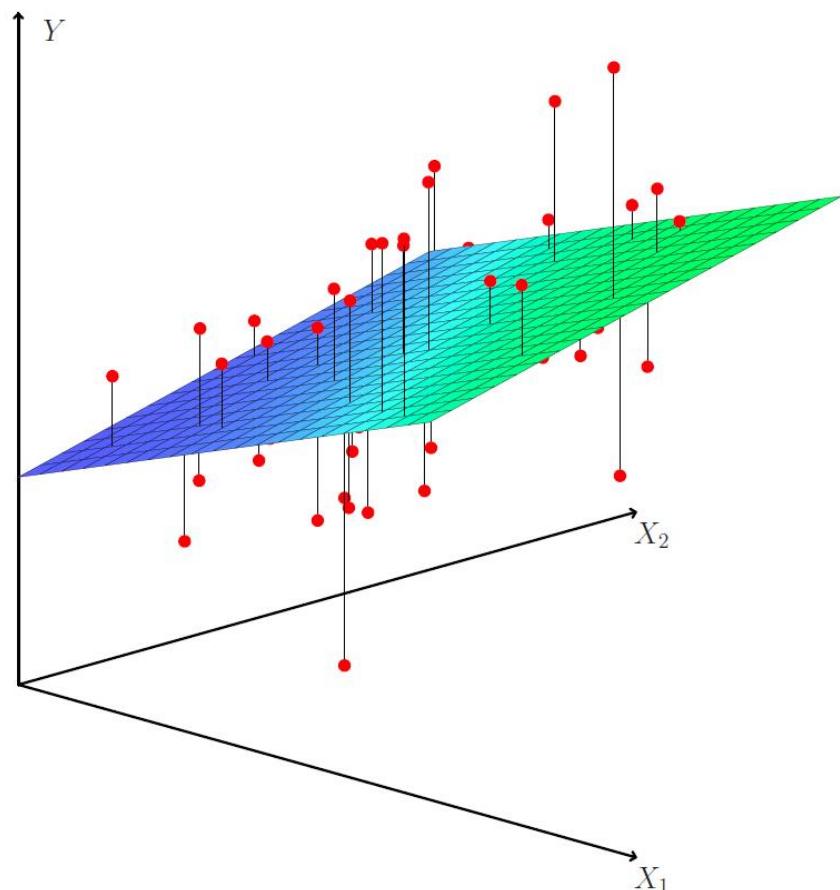
Lower interpretability than a single tree, but we get feature importances for free!

More hyperparameters now:

1. Number of trees
2. Proportion of features to use for each tree
3. Tree-specific hyperparameters

Neural networks are simply multiple regression with restricted nonlinearity

Consider multiple linear regression, which is simply regression with multiple X values:



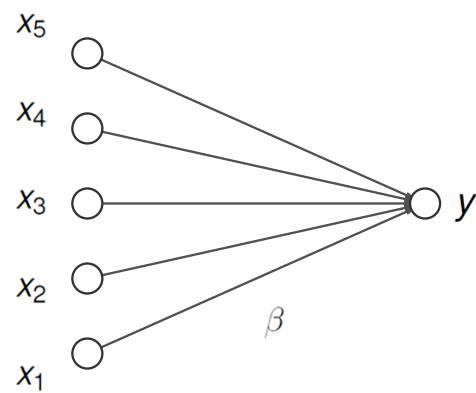
With p predictors x_1, \dots, x_p ,

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \epsilon$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & \ddots & & x_{2,p} \\ \vdots & \ddots & \ddots & \vdots & \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

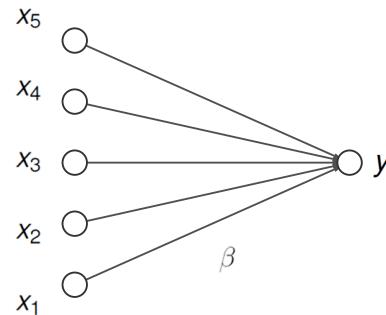
Or in matrix notation

$$y = \beta^T \mathbf{X} + \beta_0$$

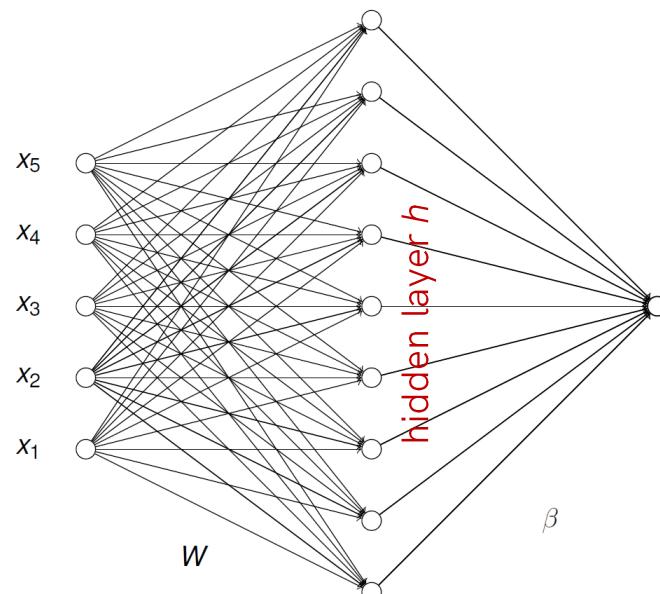


Neural networks are simply multiple regression with restricted nonlinearity

With p predictors x_1, \dots, x_p , $y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \epsilon$



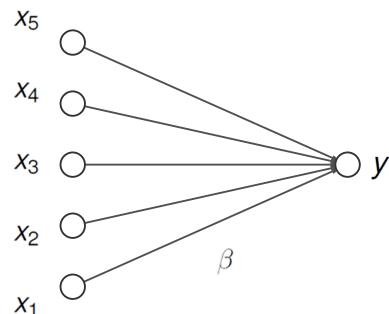
Let $y = \beta^T h + \beta_0$, where $h = Wx + b$



This is still linear regression though.

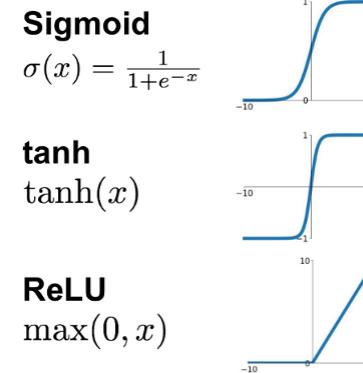
Neural networks are simply multiple regression with restricted nonlinearity

With p predictors x_1, \dots, x_p , $y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \epsilon$

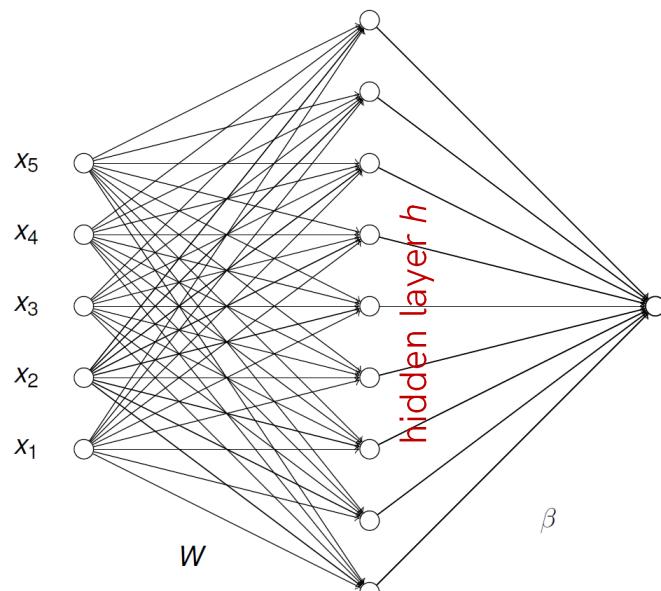


What if we apply a nonlinear activation function to each node in the hidden layer?

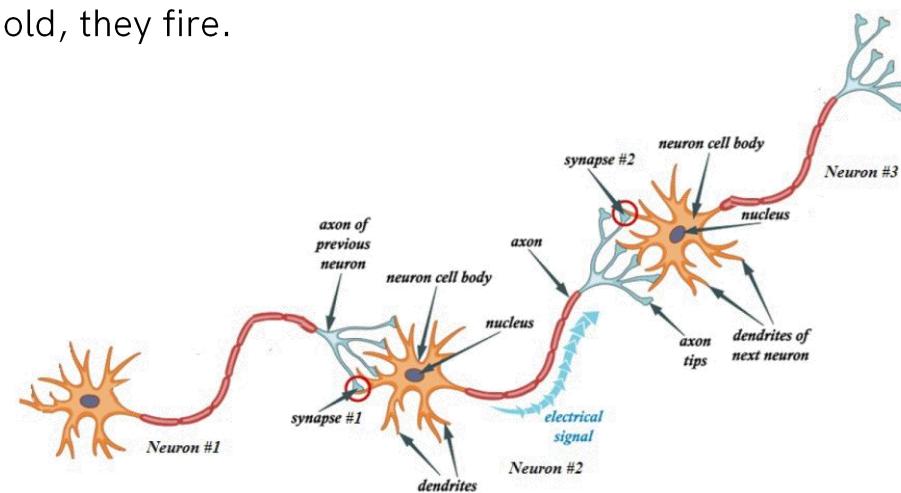
$$h = \sigma(Wx + b)$$



Let $y = \beta^T h + \beta_0$, where $h = Wx + b$



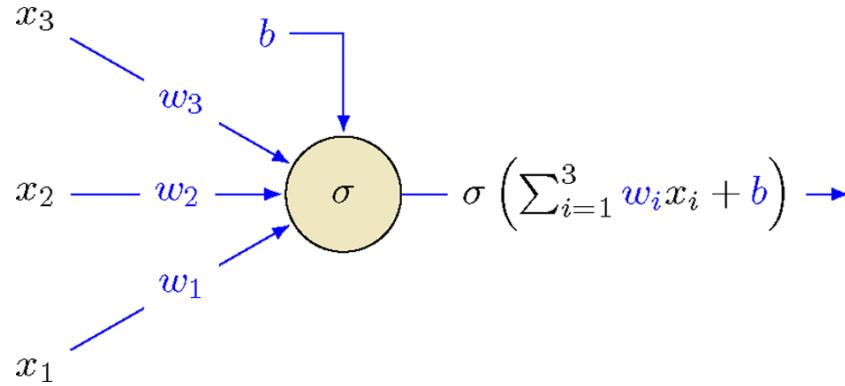
Neurons sum up multiple inputs; if the sum exceeds the activation threshold, they fire.



This is still linear regression though.

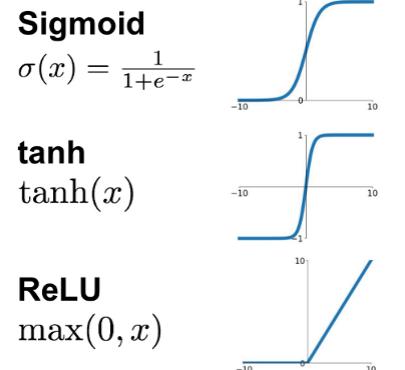
Neural networks are simply multiple regression with restricted nonlinearity

Close-up view of a neuron:

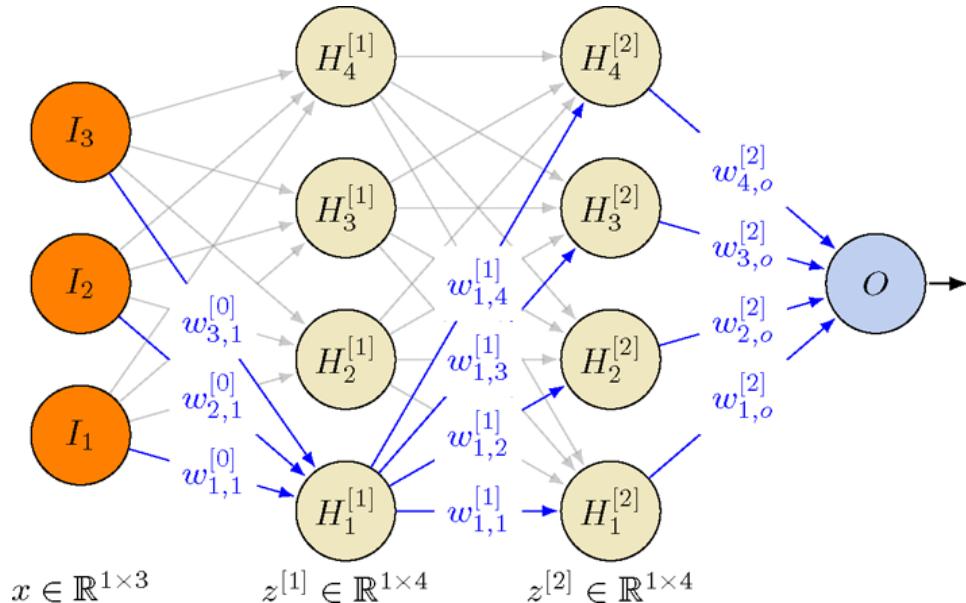


What if we apply a nonlinear activation function to each node in the hidden layer?

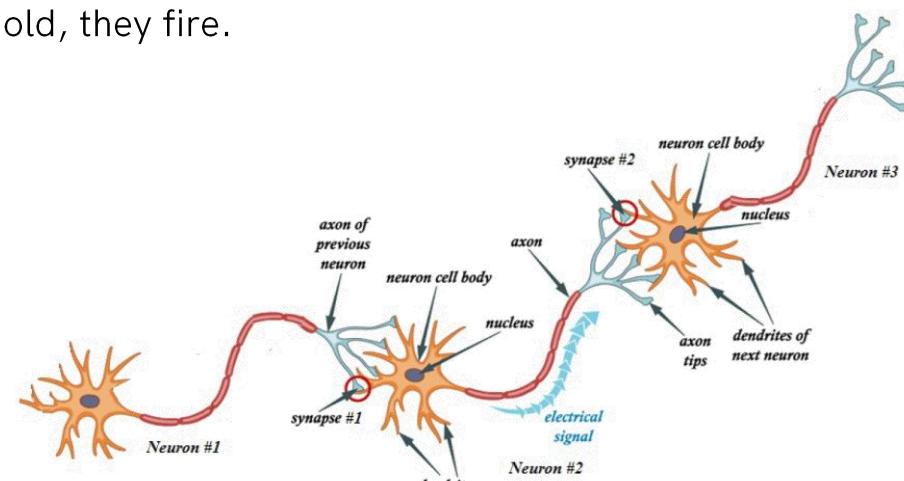
$$h = \sigma(Wx + b)$$



A multilayer network:

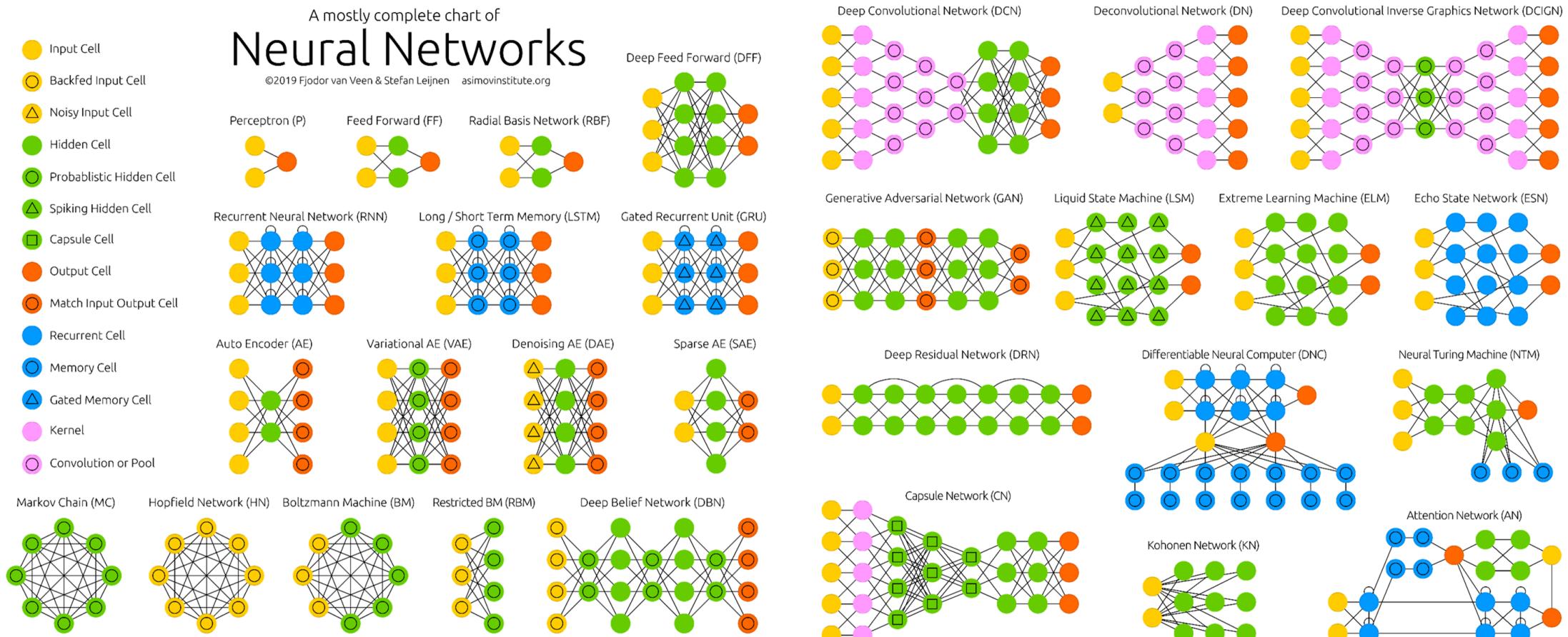


Neurons sum up multiple inputs; if the sum exceeds the activation threshold, they fire.



This framework underpins most sophisticated applications and notable achievements in ML today.

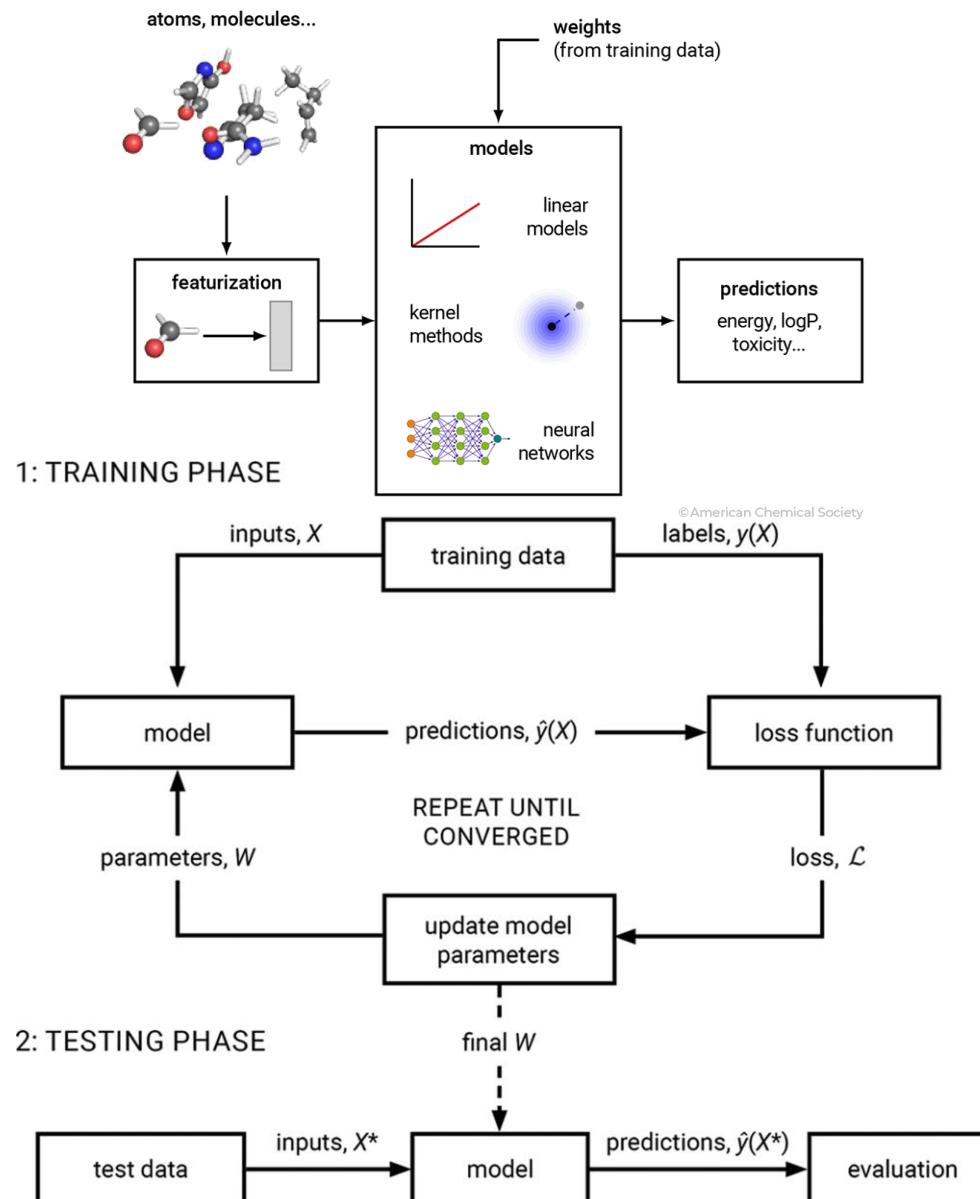
Neural networks are extremely versatile and powerful



Here we have *many* hyperparameters:

1. How many layers?
 2. How many neurons and types of neurons per layer?
 3. What activation functions to use?
 4. Training (batch size, learning rate, number of epochs, etc.)
- and more

Training a ML model in general for chemical applications



1. Define objectives

- a) What do we hope to achieve?
- b) How much data are available and what is their quality?
- c) How accurate does the model need to be to be useful?

2. Choose a representation/featurization

- a) How do we convert structure into numbers? (2D/3D/higher-D)
- b) How many features? Do we need algorithms to select features?
- c) Does a simple ML model validate the proposed featurization?

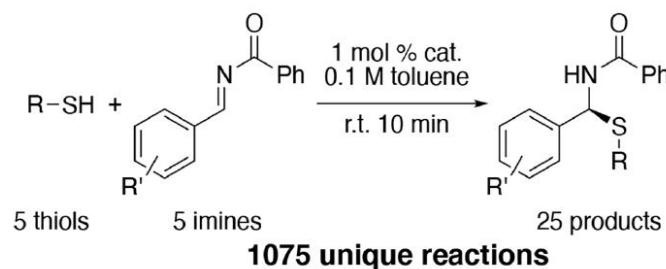
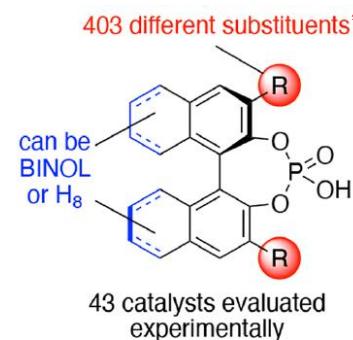
3. Generate and process raw data (often time-intensive)

4. Train/select a model

- a) Partition data into training and test sets
- b) Hyperparameter optimization (grid search, nested)
- c) Select metric(s) to evaluate model performance

5. Make predictions on test set and evaluate

Prediction of higher-selectivity catalysts by computer-driven workflow and ML



1. Define objectives

- a) What do we hope to achieve?

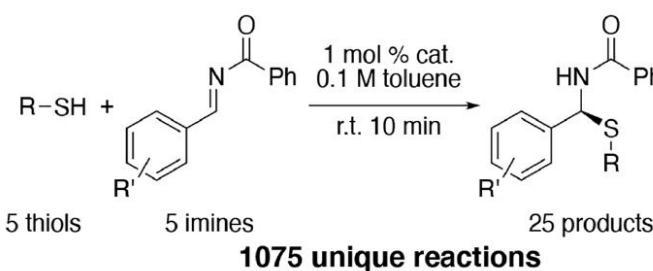
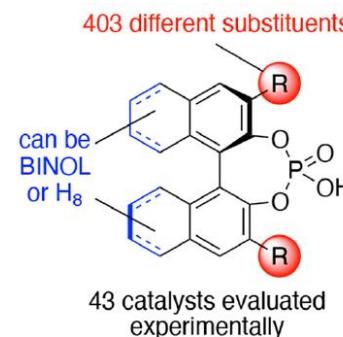
Predict CPAs that yield higher enantioselectivities for a specific reaction

- b) How much data are available and what is their quality?

- c) How accurate does the model need to be to be useful?

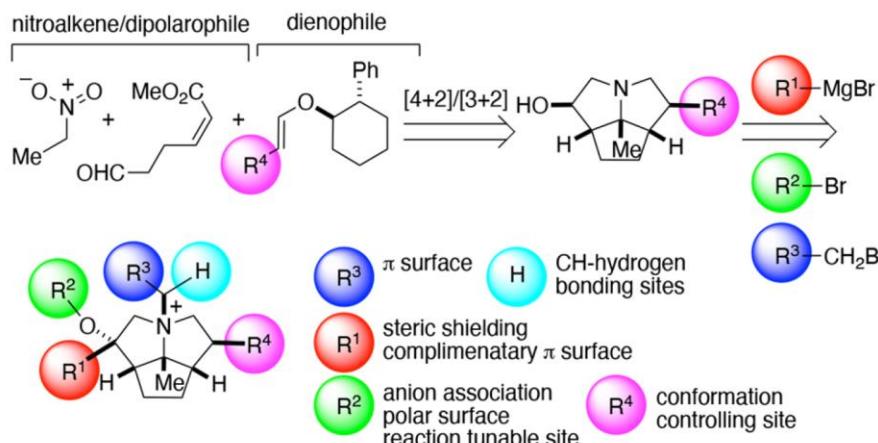
*based on commercially available ArB(OR)₂, ArX, ArMgX, etc.

Prediction of higher-selectivity catalysts by computer-driven workflow and ML



1. Define objectives

- What do we hope to achieve?
Predict CPAs that yield higher enantioselectivities for a specific reaction
- How much data are available and what is their quality?
Relatively small datasets, ~10³ reactions with reproducible ees are feasible
- How accurate does the model need to be to be useful?

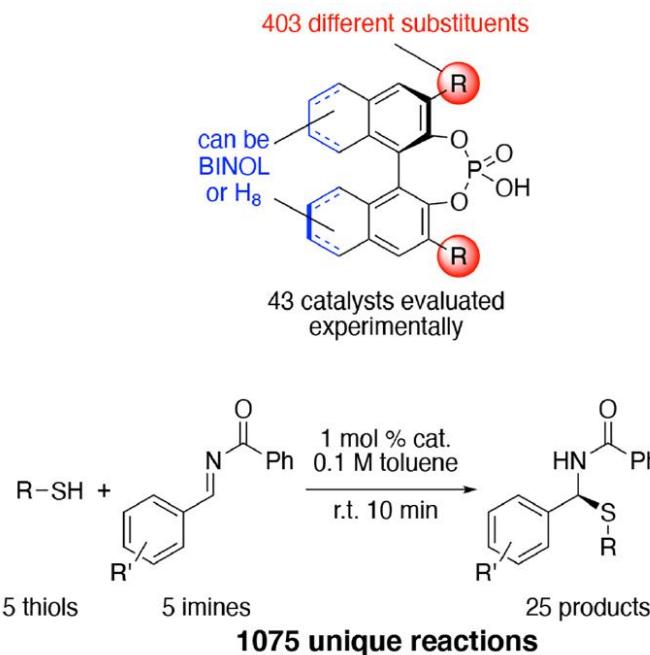


160 CPP catalysts were synthesized in a years-long attempt to build a 3D-QSAR model for enantioselectivity

The model never achieved high predictive accuracy

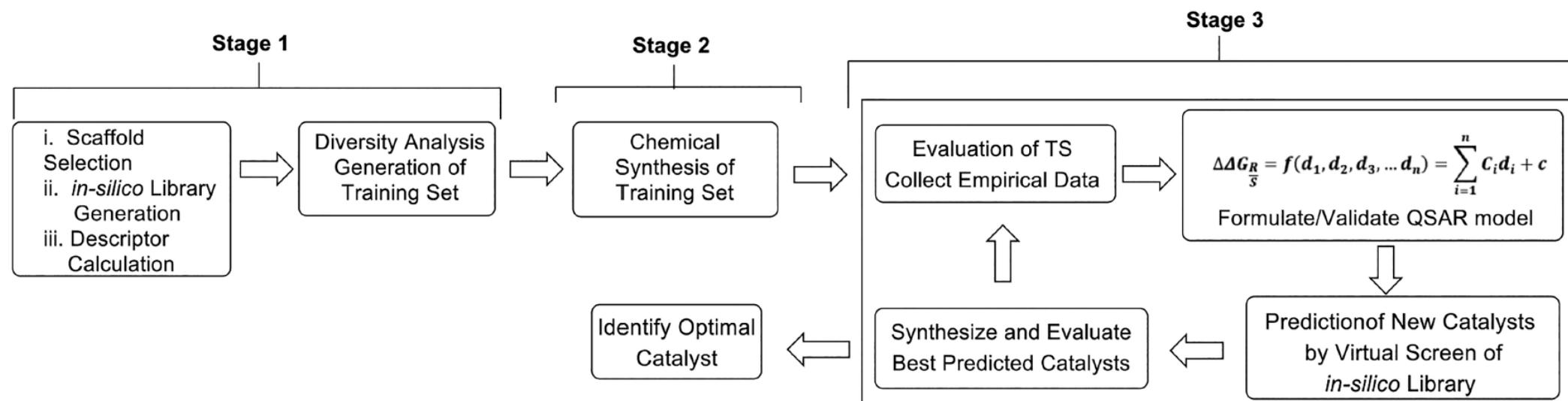
"The lesson from this years-long enterprise was that to create reliable extrapolative models from 3D-QSAR analyses, better methods are needed to obtain the appropriate training set data."

Prediction of higher-selectivity catalysts by computer-driven workflow and ML

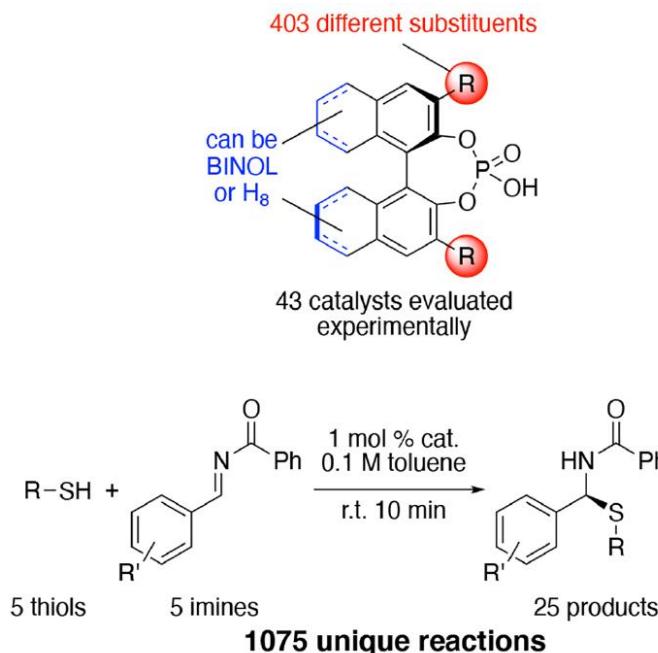


1. Define objectives

- What do we hope to achieve?
Predict CPAs that yield higher enantioselectivities for a specific reaction
- How much data are available and what is their quality?
Relatively small datasets, $\sim 10^3$ reactions with reproducible ees are feasible
Data size limitation necessitates algorithmically guided curation of representative catalysts
- How accurate does the model need to be to be useful?

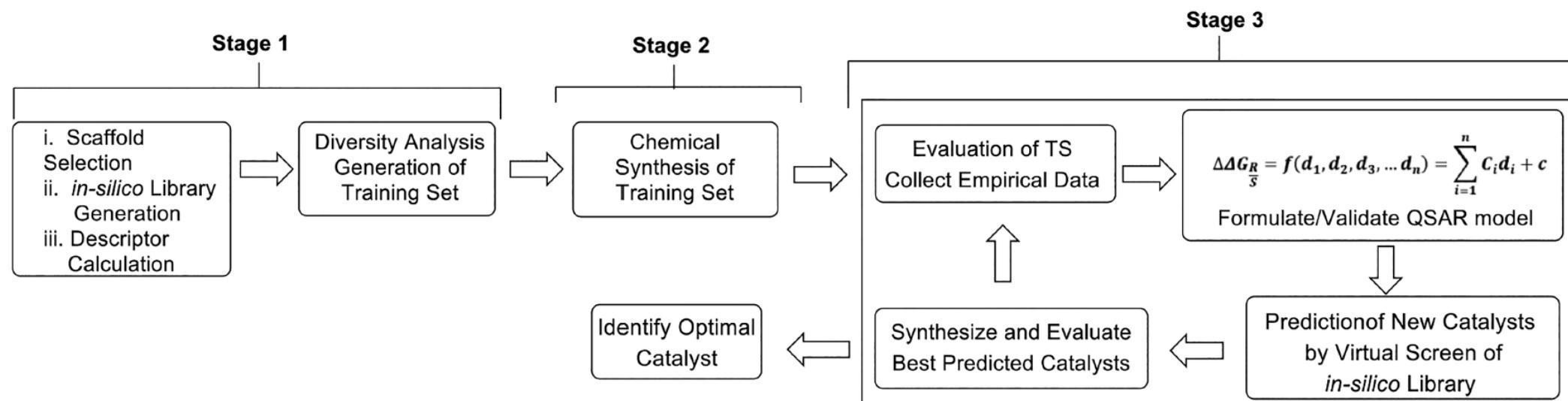


Prediction of higher-selectivity catalysts by computer-driven workflow and ML



1. Define objectives

- What do we hope to achieve?
Predict CPAs that yield higher enantioselectivities for a specific reaction
- How much data are available and what is their quality?
Relatively small datasets, $\sim 10^3$ reactions with reproducible ees are feasible
Data size limitation necessitates algorithmically guided curation of representative catalysts
- How accurate does the model need to be to be useful?
Chemical accuracy (1 kcal mol^{-1}) is insufficient
Model must also predict selectivities higher than those present in the training set



Conventional descriptors did not lead to meaningful models

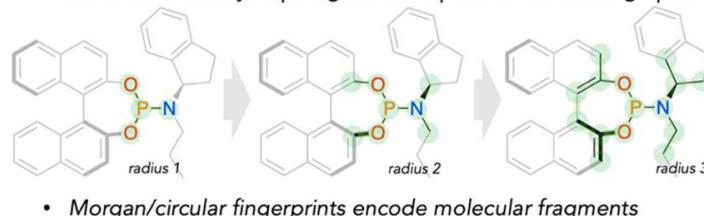
1D representations

Constitutional descriptors

- Num. C atoms = 32 Num. Rings = 7

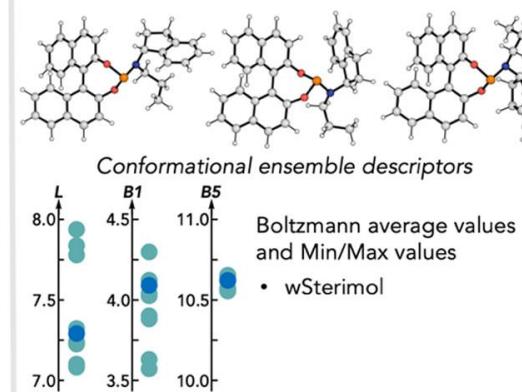
2D representations

Atomic connectivity: topological descriptors & molecular graphs



4D representations

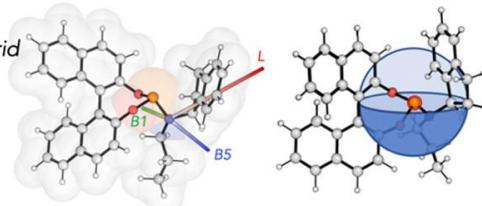
Conformational ensemble descriptors



3D representations

Grid-based descriptors map properties to 3D-grid points (orientation dependent)

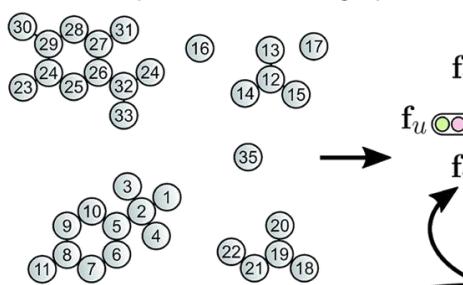
- CoMFA
- Average Steric Occupancy (ASO)



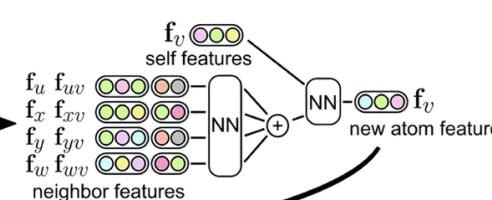
Optimized 3D-geometry described by various shape/steric descriptors:

- Sterimol (multi-dimensional)
- Buried volume (scalar)

A. Reactant pool as attributed graph



B. Iterative convolutional embedding of atom feature vectors



2. Choose a representation/featurization

- a) How do we convert structure into numbers? (2D/3D/higher-D)

For most BINOL-CPAs, 40–70% of connectivity is conserved, so 3D is necessary

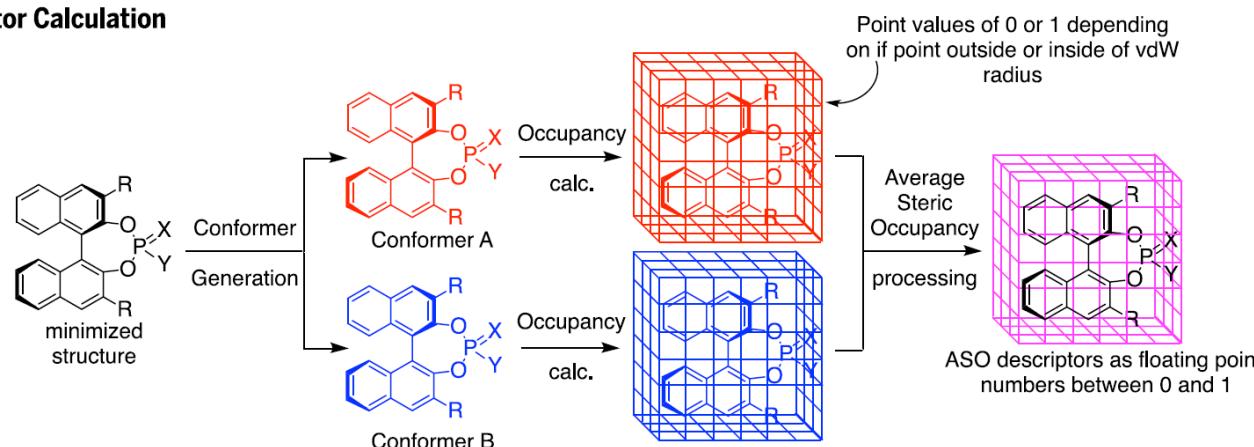
- b) How many features? Do we need algorithms to select features?

e.g. a graph convolutional neural network (GCN) represents molecules as 2D graphs

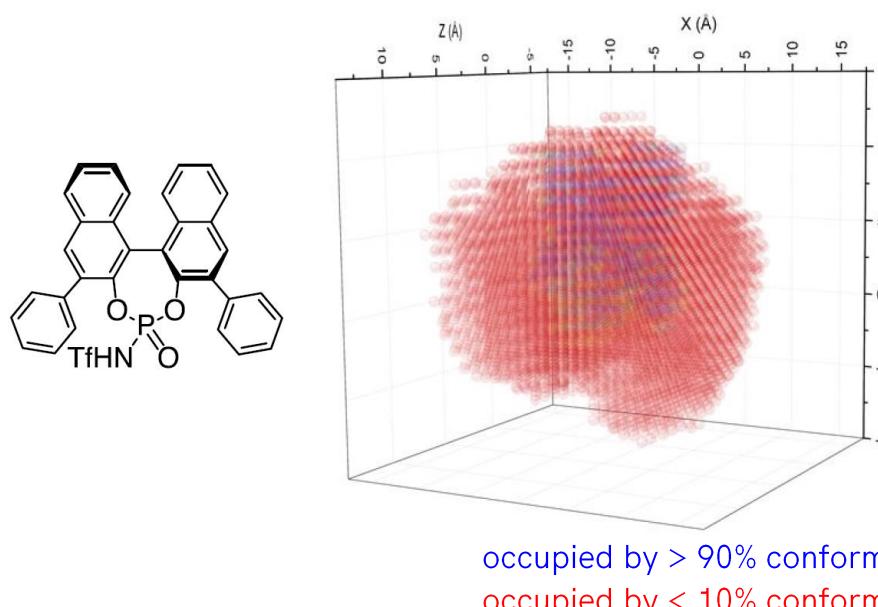
A new set of descriptors to represent conformational ensemble values

Average Steric Occupancy (ASO) as geometric descriptor

A Descriptor Calculation



B ASO values with color coded grid points



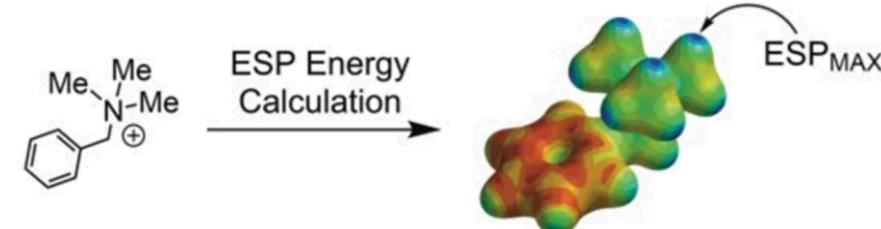
2. Choose a representation/featurization

- a) How do we convert structure into numbers? (2D/3D/higher-D)

For most BINOL-CPAs, 40–70% of connectivity is conserved, so 3D is necessary

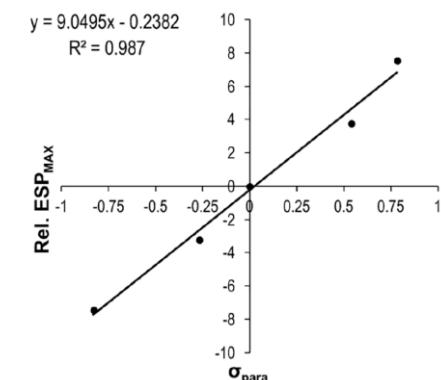
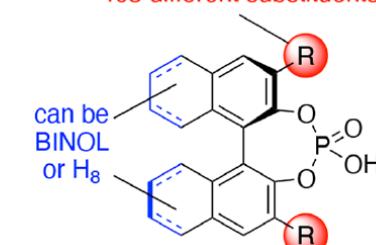
- b) How many features? Do we need algorithms to select features?

ESPM_{Max} as electronic descriptor



Use maximum ESP on $\text{R}-\text{CH}_2\text{NMe}_3$ to describe through-bond electronic effects

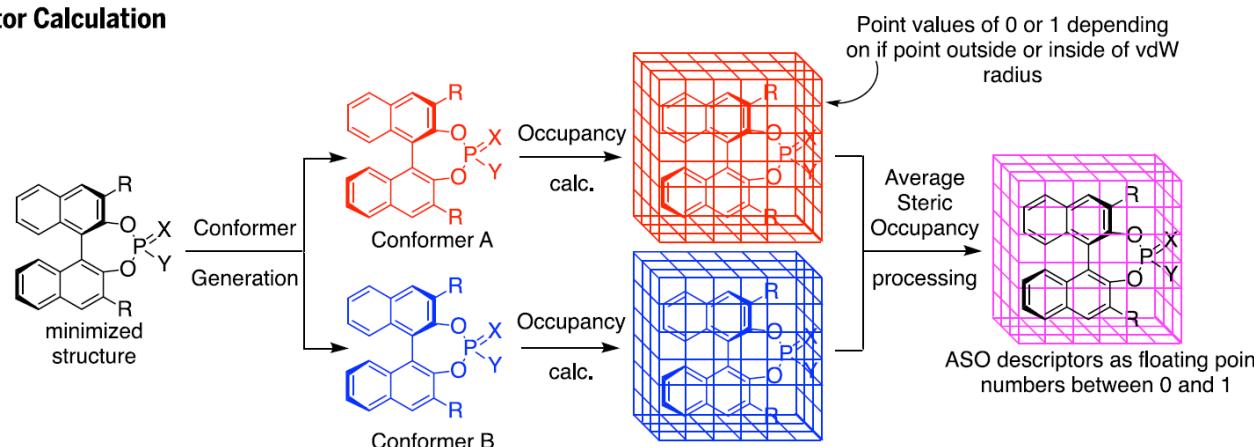
403 different substituents



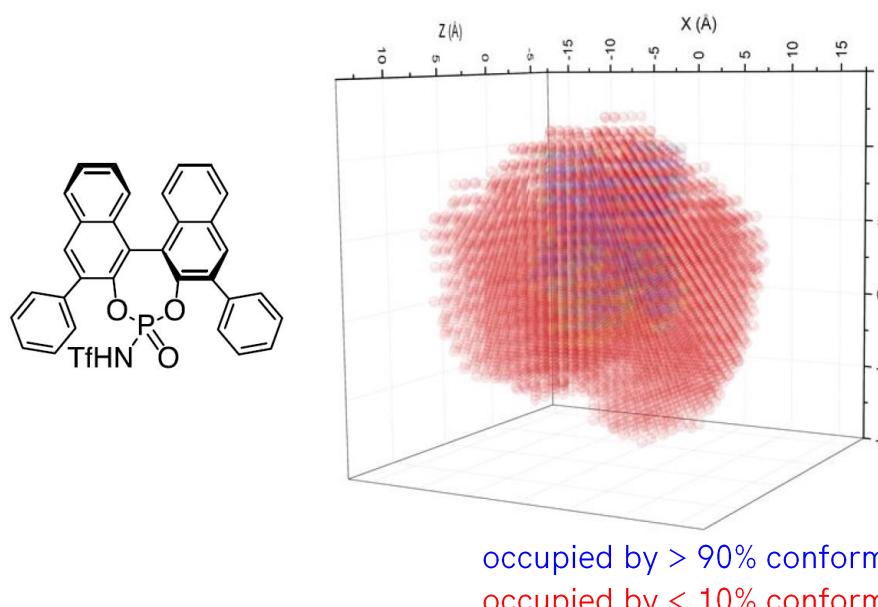
A new set of descriptors to represent conformational ensemble values

Average Steric Occupancy (ASO) as geometric descriptor

A Descriptor Calculation



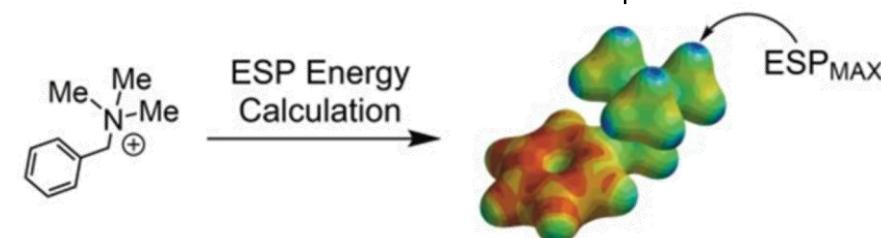
B ASO values with color coded grid points



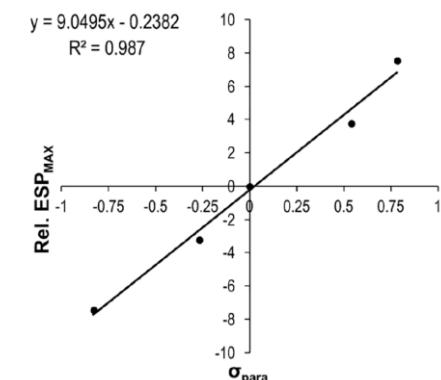
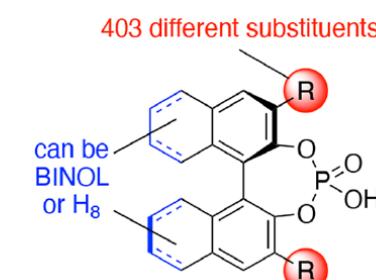
2. Choose a representation/featurization

- How do we convert structure into numbers? (2D/3D/higher-D)
For most BINOL-CPAs, 40–70% of connectivity is conserved, so 3D is necessary
- How many features? Do we need algorithms to select features?
16384 features/catalyst. Used all of them

ESPM_{Max} as electronic descriptor

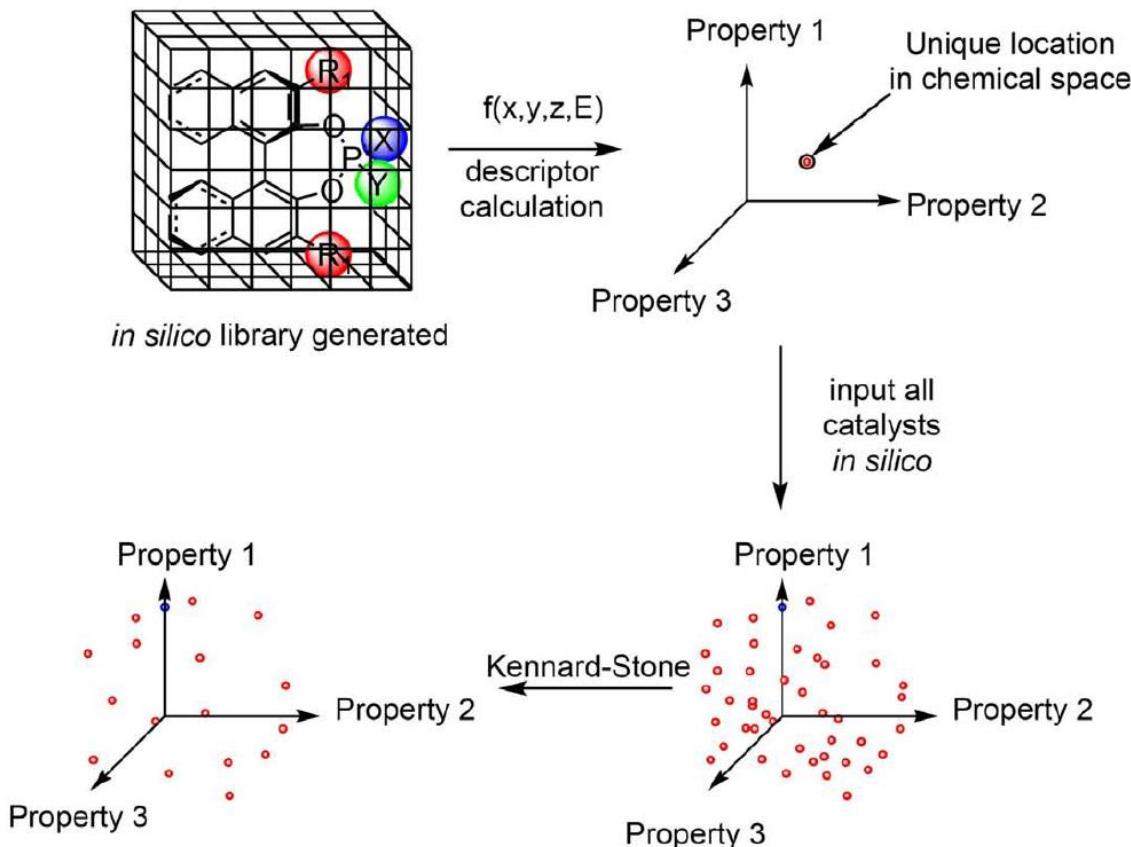


Use maximum ESP on R-CH₂NMe₃ to describe through-bond electronic effects



Data generation for all 806 catalysts is not feasible, necessitating algorithmic selection of a representative set

A Subset Selection



3. Generate and process raw data (often time-intensive)

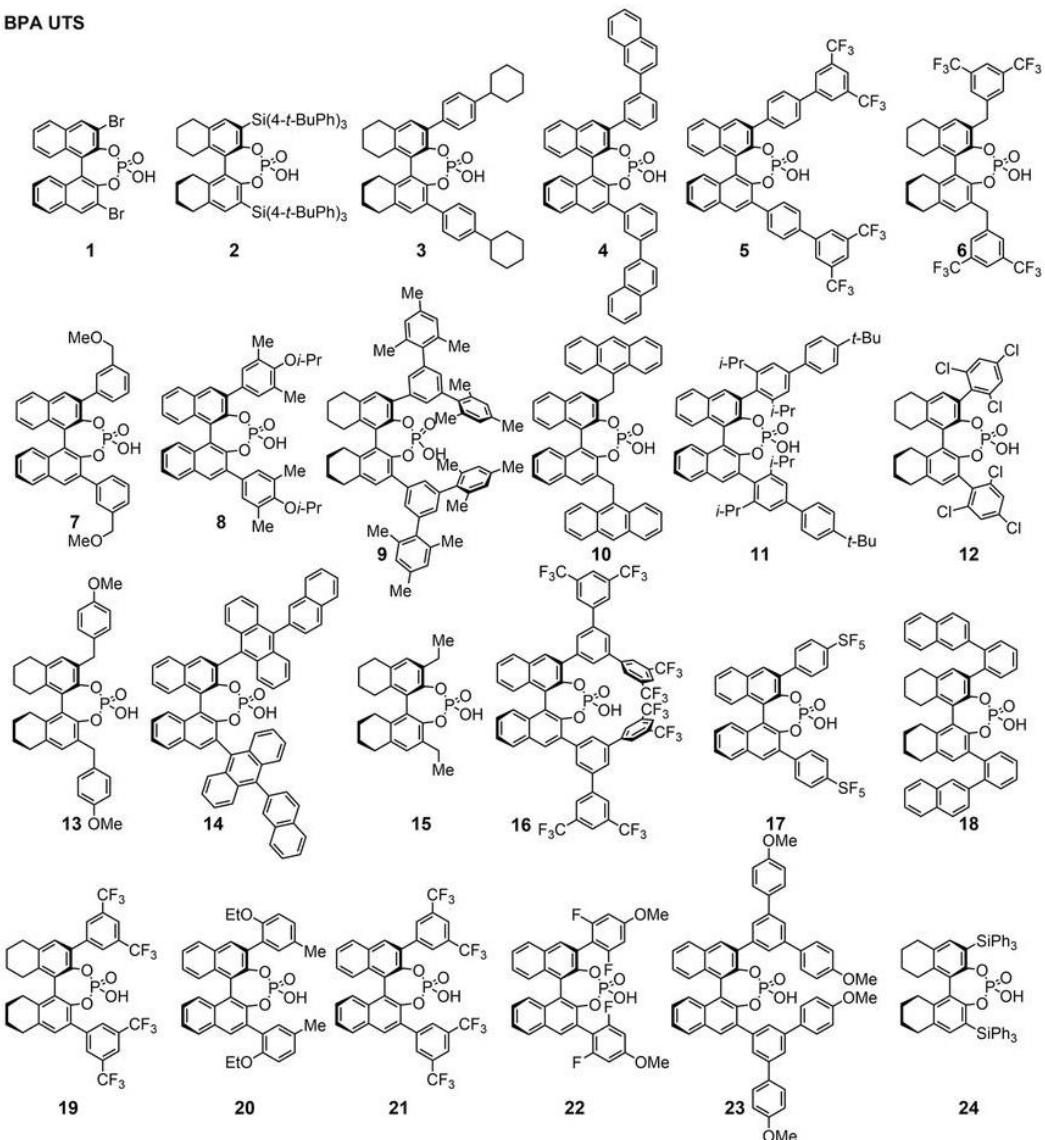
Library of 806 *in silico* catalysts span “entire” chemical space for CPAs

Collation of ASO and ESPMax features defines chemical space of the library

Kennard-Stone: select n points that are farthest apart in a multidimensional space

Data generation for all 806 catalysts is not feasible, necessitating algorithmic selection of a representative set

A BPA UTS



3. Generate and process raw data (often time-intensive)

Library of 806 *in silico* catalysts span “entire” chemical space for CPAs

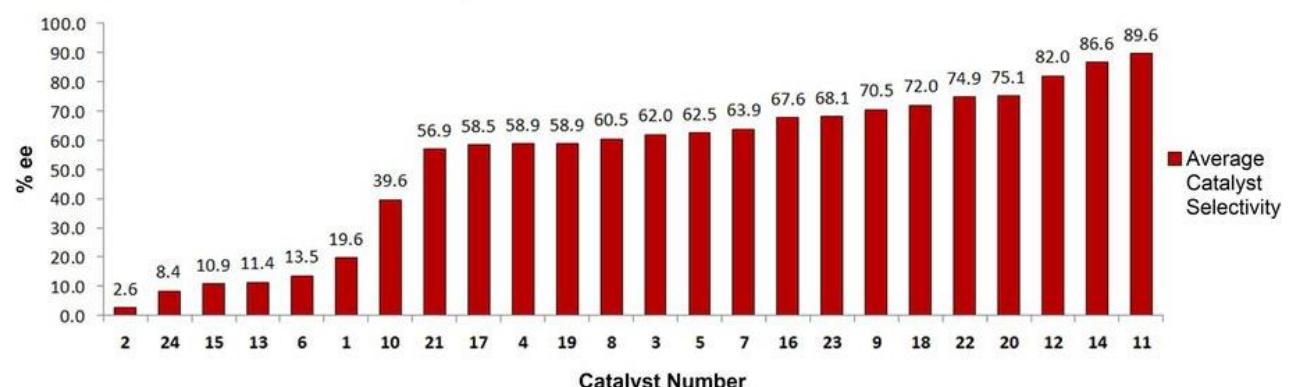
Collation of ASO and ESPMax features defines chemical space of the library

Catalysts span uniform regions of feature space

Synthetically inaccessible catalysts were substituted with their nearest neighbor

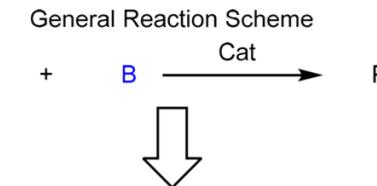
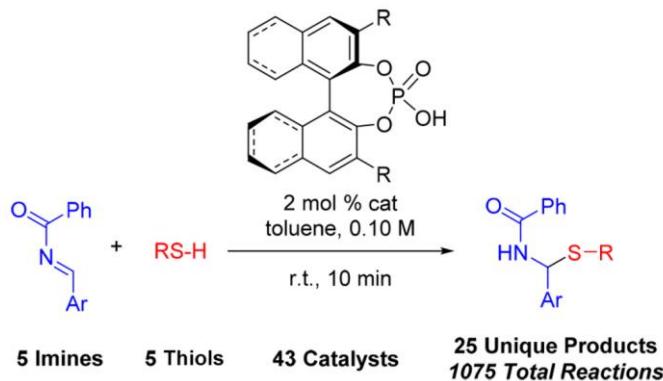
19 additional catalysts were synthesized for testing

B Average Selectivities of Training Set Compounds



UTS increases probability of finding selective catalyst right away

Setup: featurizing all reaction components



Individual Reaction Component Parameterization

$A_1 ; a_1, a_2, a_3$	$B_1 ; b_1, b_2, b_3$	$\text{cat}_1 ; c_1, c_2, c_3$
$A_2 ; a_4, a_5, a_6$	$B_2 ; b_4, b_5, b_6$	$\text{cat}_2 ; c_4, c_5, c_6$

Combinatorial Generation of Reaction Parameterization

$R_1 ; a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3$
$R_2 ; a_1, a_2, a_3, b_4, b_5, b_6, c_1, c_2, c_3$
$R_3 ; a_4, a_5, a_6, b_1, b_2, b_3, c_1, c_2, c_3$
$R_4 ; a_4, a_5, a_6, b_4, b_5, b_6, c_1, c_2, c_3$
$R_5 ; a_1, a_2, a_3, b_1, b_2, b_3, c_4, c_5, c_6$
$R_6 ; a_1, a_2, a_3, b_4, b_5, b_6, c_4, c_5, c_6$
$R_7 ; a_4, a_5, a_6, b_1, b_2, b_3, c_4, c_5, c_6$
$R_8 ; a_4, a_5, a_6, b_4, b_5, b_6, c_4, c_5, c_6$

Each reaction has identical features with some other reactions - models fit on the structure in this data rather than learning chemical information.

Randomly Partition Training and Test Sets

Training Set

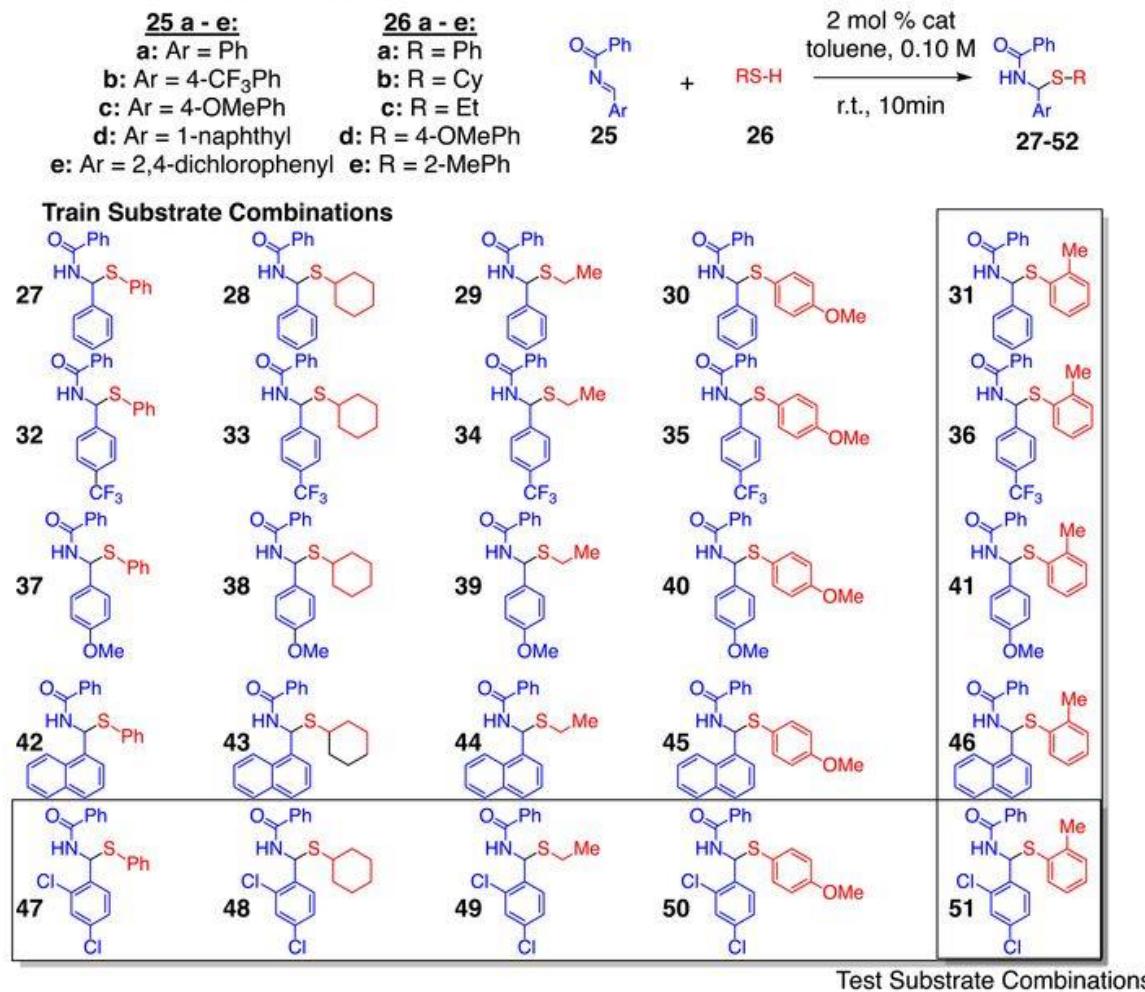
$R_1 ; a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3$
$R_2 ; a_1, a_2, a_3, b_4, b_5, b_6, c_1, c_2, c_3$
$R_4 ; a_4, a_5, a_6, b_4, b_5, b_6, c_1, c_2, c_3$
$R_5 ; a_1, a_2, a_3, b_1, b_2, b_3, c_4, c_5, c_6$
$R_6 ; a_1, a_2, a_3, b_4, b_5, b_6, c_4, c_5, c_6$
$R_8 ; a_4, a_5, a_6, b_4, b_5, b_6, c_4, c_5, c_6$

Test Set

$R_3 ; a_4, a_5, a_6, b_1, b_2, b_3, c_1, c_2, c_3$
$R_7 ; a_4, a_5, a_6, b_1, b_2, b_3, c_4, c_5, c_6$

Test 1: Does the UTS approach enable prediction of selectivities for unseen reaction components?

A Model Reaction with Training and Test Substrate Combinations

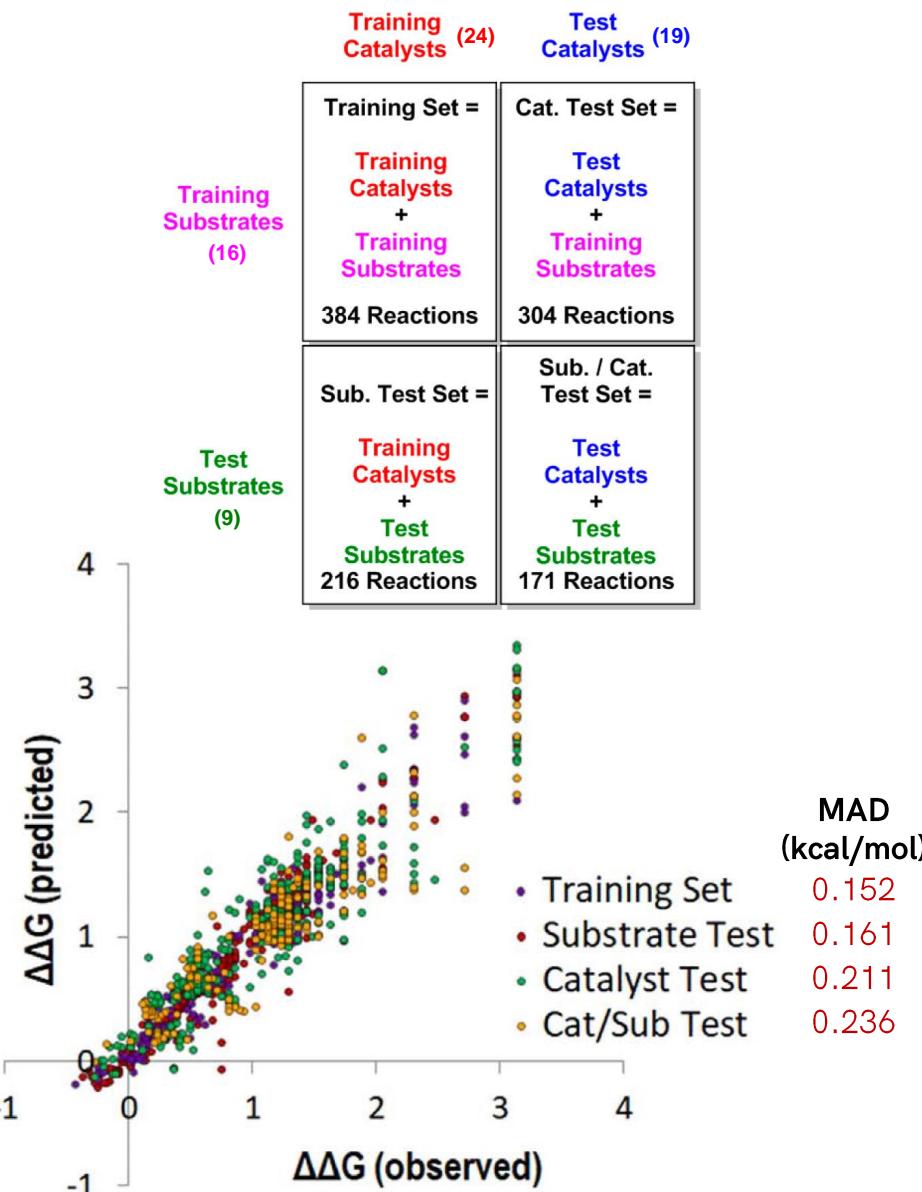


Train/test partitioning in this way is crucial:
every test reaction has one component not present
in any training reaction

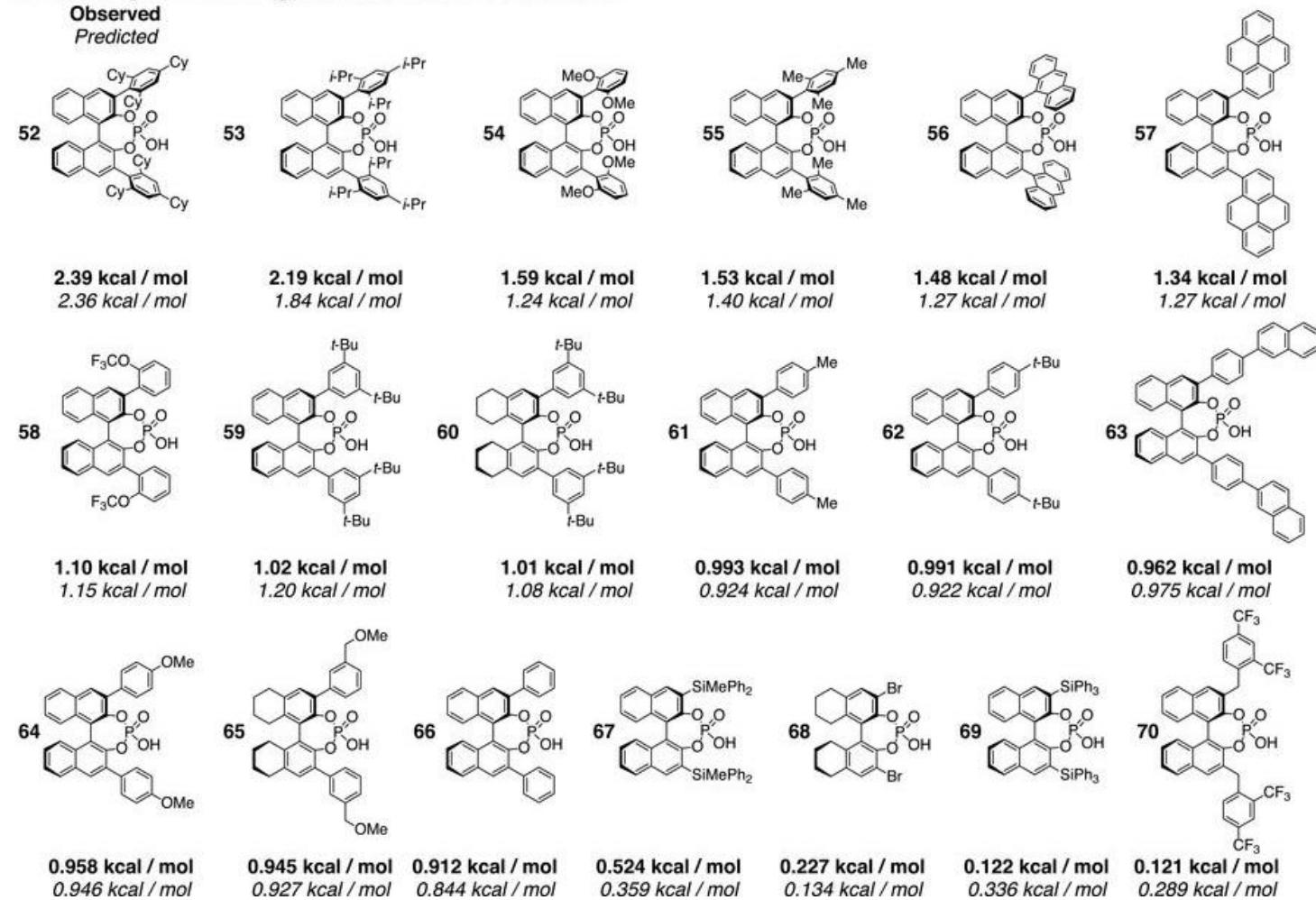
Training Catalysts (24)	Test Catalysts (19)
Training Set = Training Catalysts + Training Substrates	Cat. Test Set = Test Catalysts + Training Substrates
384 Reactions	304 Reactions
Sub. / Cat. Test Set = Training Catalysts + Test Substrates	Sub. / Cat. Test Set = Test Catalysts + Test Substrates
216 Reactions	171 Reactions

Training Substrates (16)
Test Substrates (9)

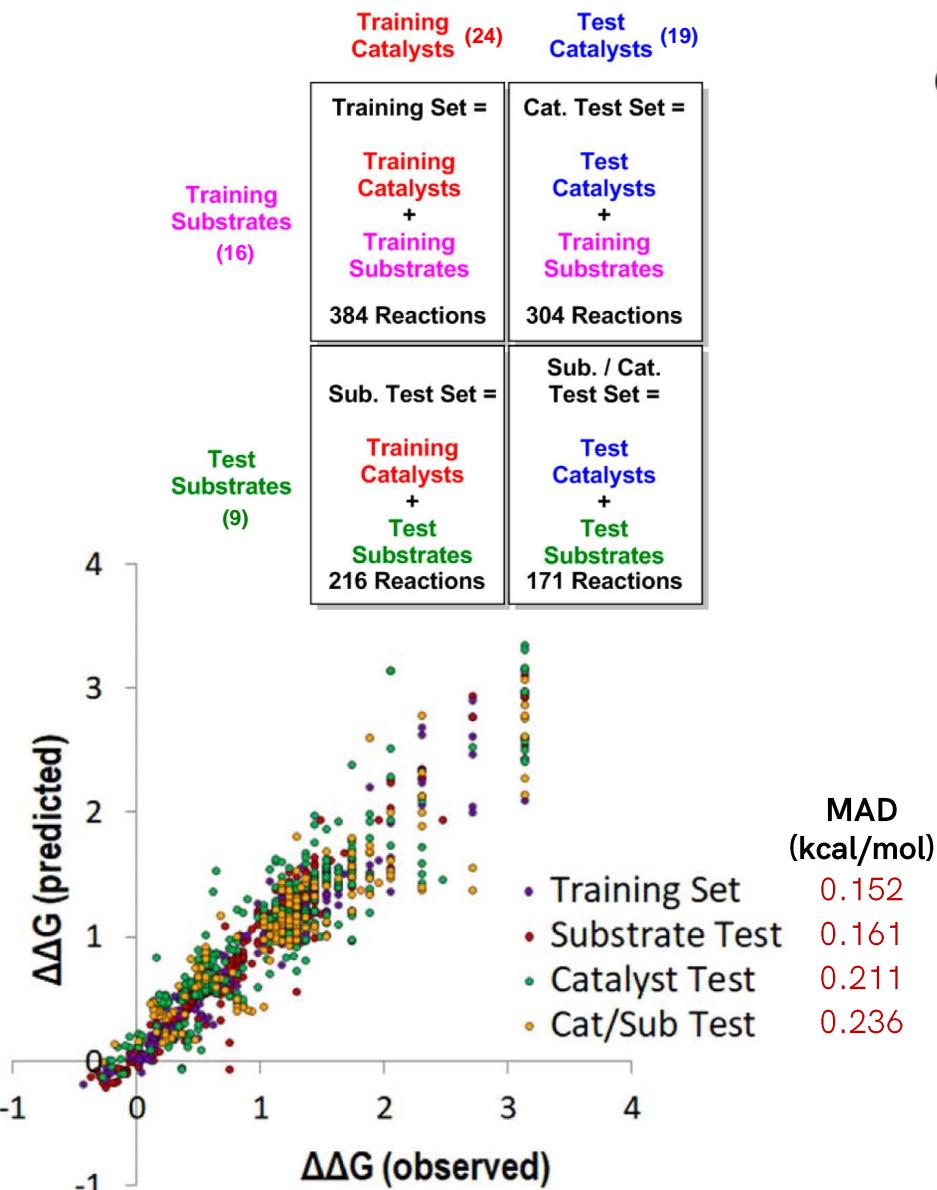
Test 1: Does the UTS approach enable prediction of selectivities for unseen reaction components?



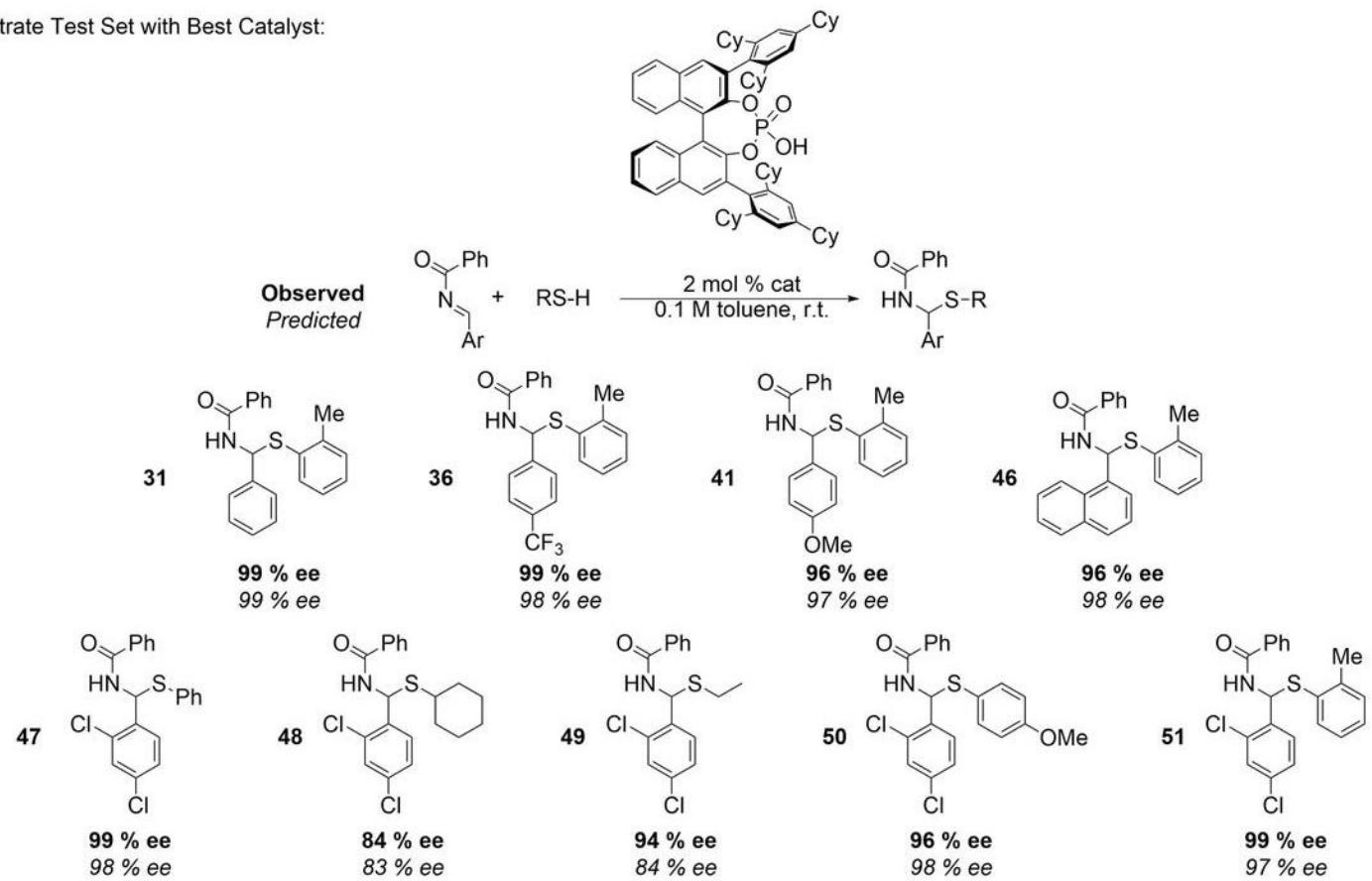
B Test Catalysts with Averages for All Substrate Combinations



Test 1: Does the UTS approach enable prediction of selectivities for unseen reaction components?

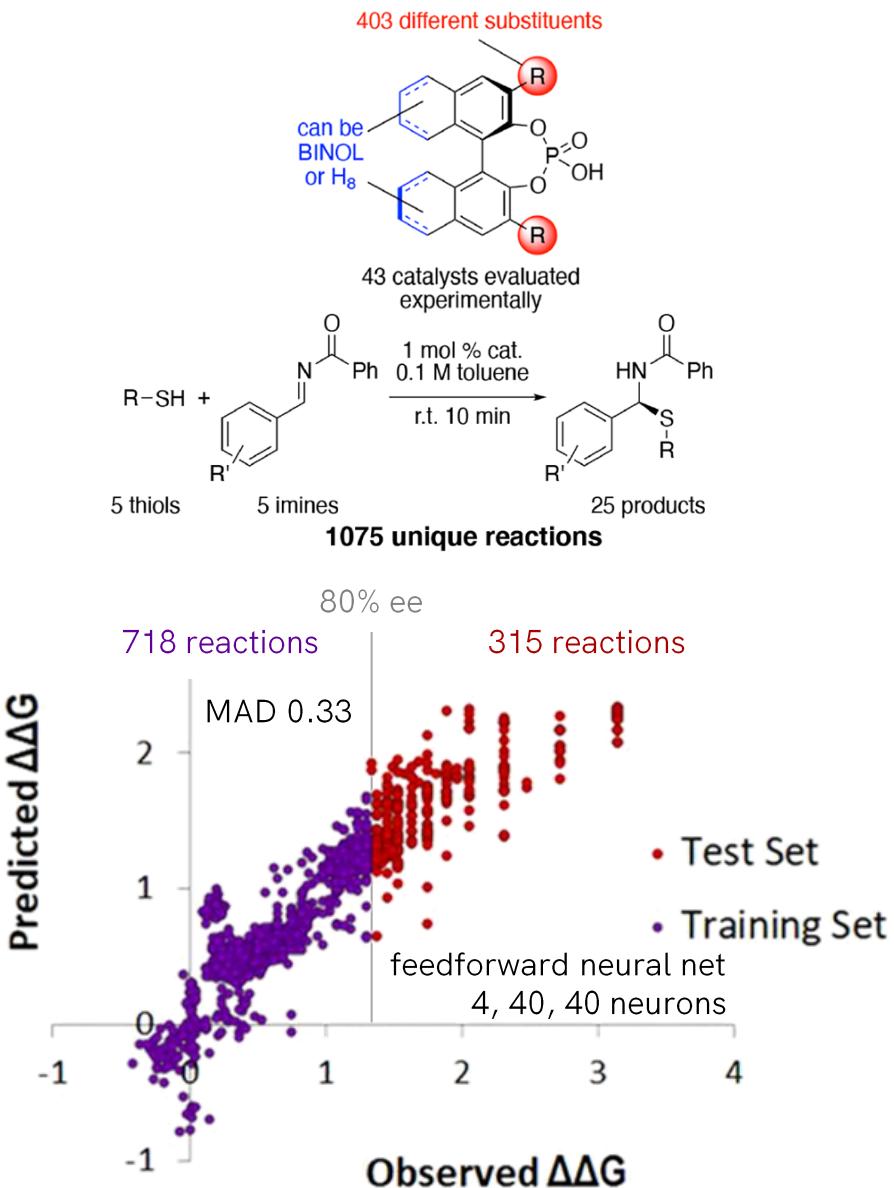


C Substrate Test Set with Best Catalyst:

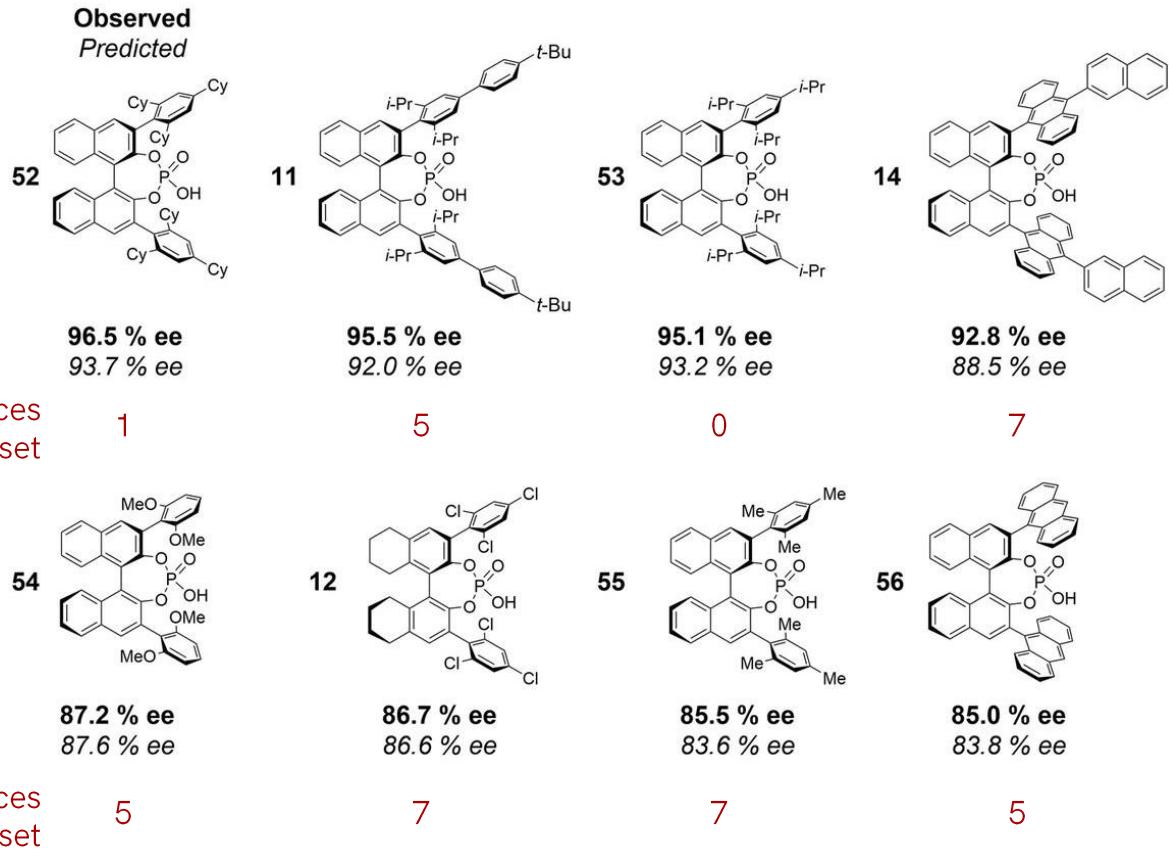


Predicted to be best catalyst despite not being in any training data
ee predictions for test substrates are accurate

Test 2: Can the model trained on less selective reactions predict highly selective reactions?



B Average Test Catalyst Selectivity:



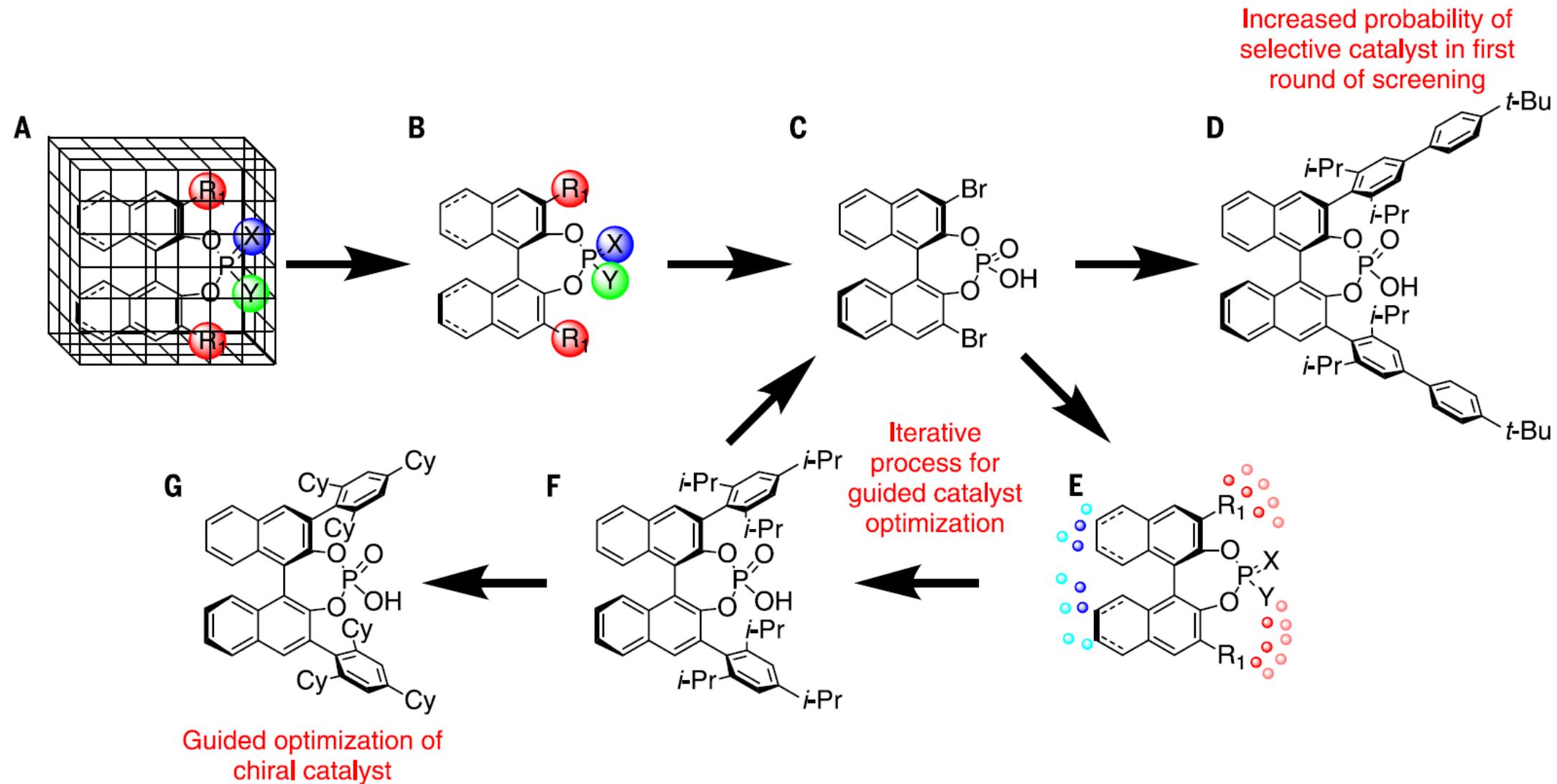
Appearances in training set

Argument: low-ee space contains enough information to "extrapolate" to high-ee predictions

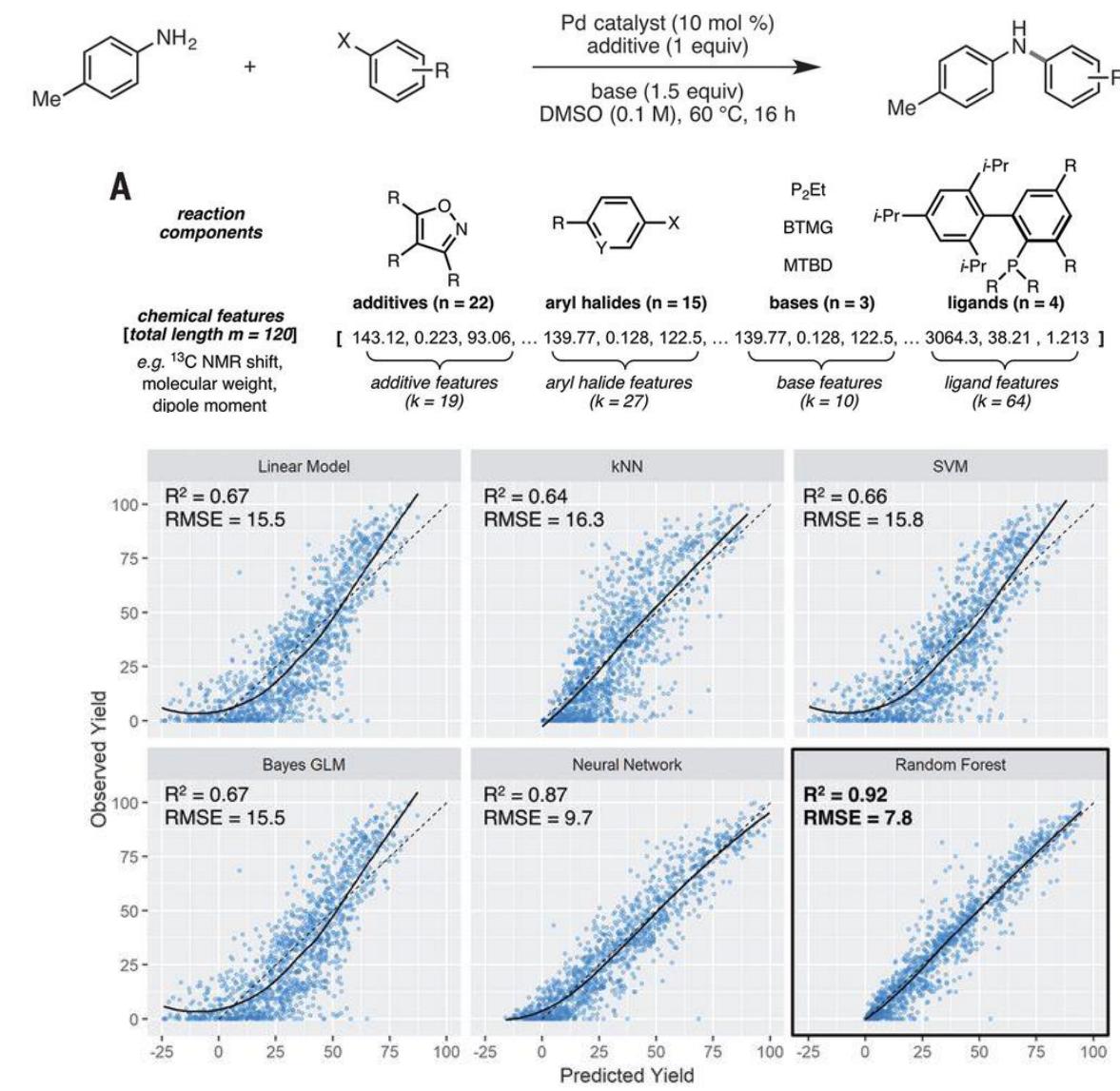
Could this have been achieved with just < 80% ee reactions with the UTS?

No generation of catalyst design hypotheses (interpolation in catalyst space)

A fully chemoinformatics-guided workflow

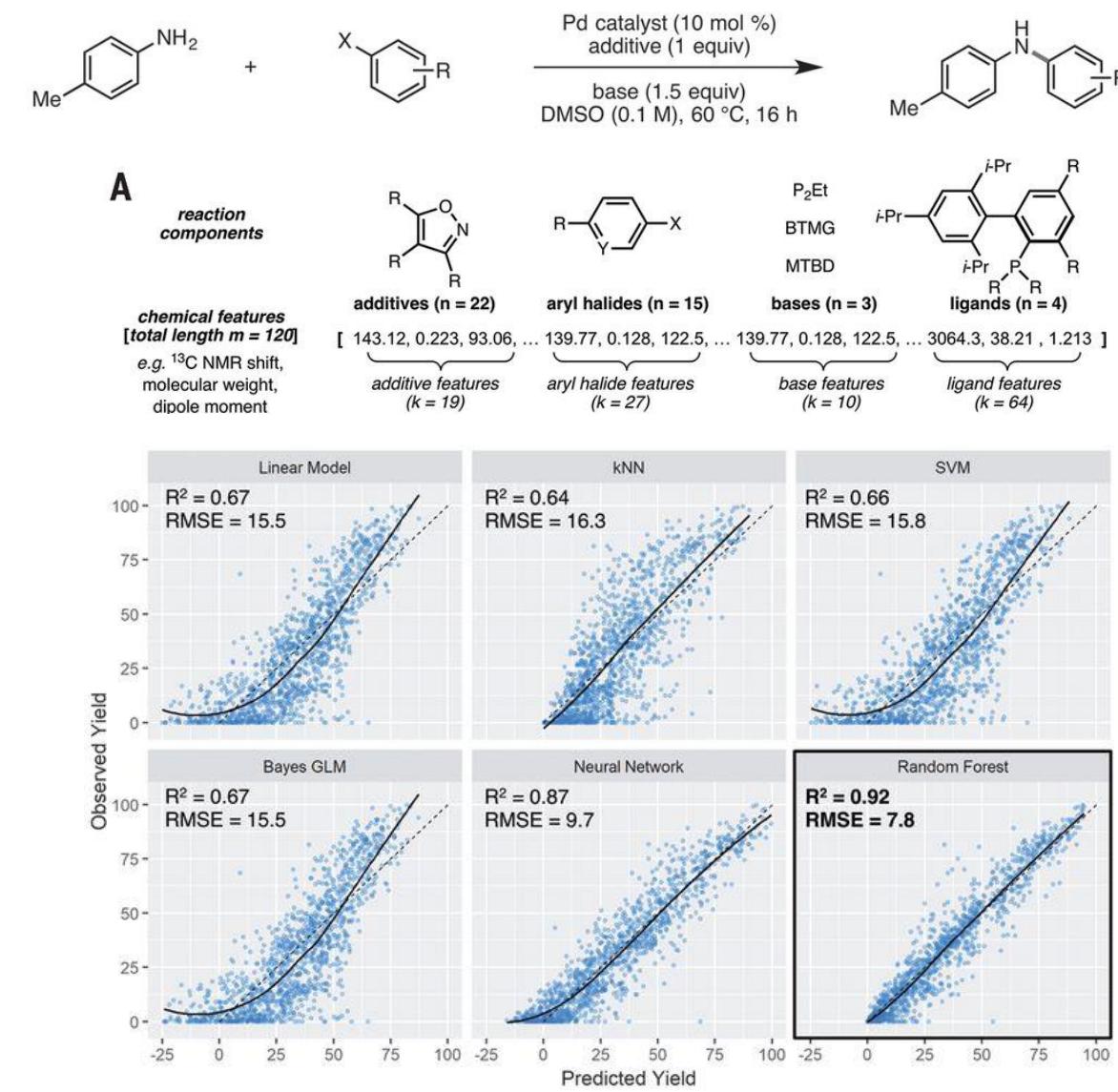


Sanity check: are models finding chemically meaningful trends, or just fitting to patterns in data?

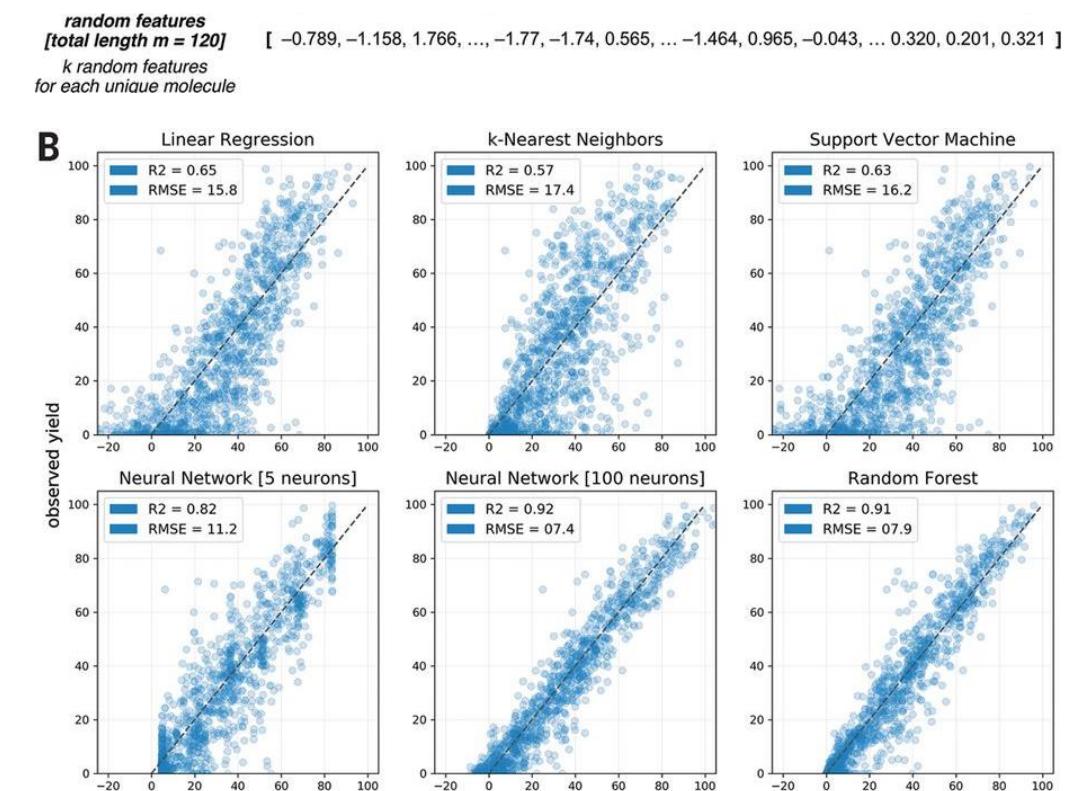


4608 reactions run by HTE
Random 70/30 train/test split
Random forest model performed the best

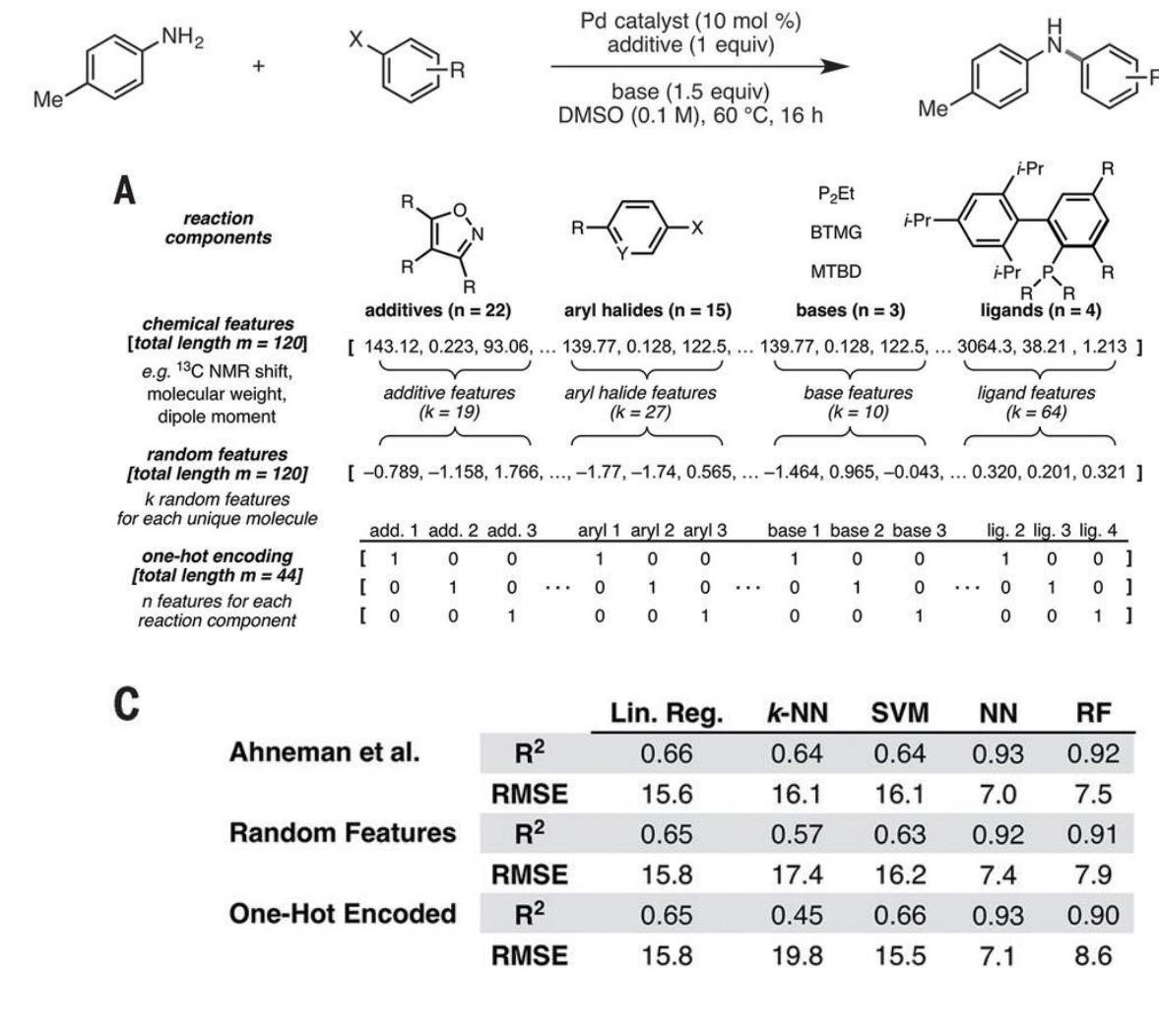
Sanity check: are models finding chemically meaningful trends, or just fitting to patterns in data?



But replacing the chemical descriptors with random numbers and training the model gives equally good results!



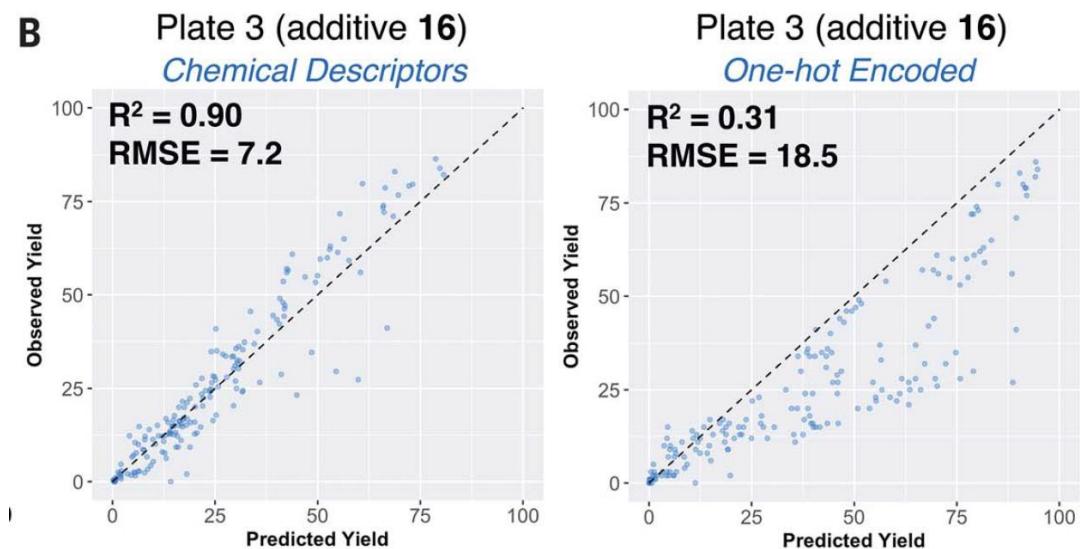
Sanity check: are models finding chemically meaningful trends, or just fitting to patterns in data?



Random data splitting gives overoptimistic test error:
the model may be *fitting to structure in the training data* that leaks over to the test data

"Out-of-sample" test reactions have to be enforced.

Always run control experiments with random features and one-hot encoding.



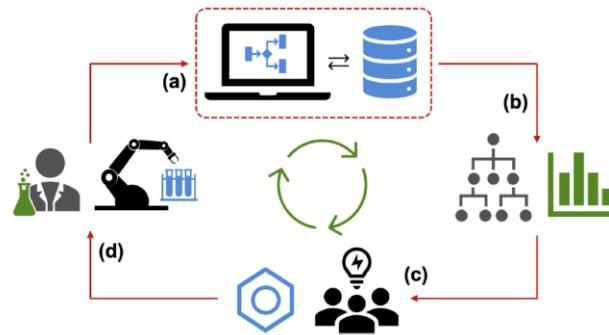
For poorer performing additives, the chemical features still predict their poisoning effect much better

Looking forward

ML for reaction optimization requires structured datasets:

The ability to access structured data sets that represent the chemical space of a reaction and provide more information on reaction progress and outcomes beyond yield and selectivity will continue to aid in the development of data science in organic chemistry.

ACS Cent. Sci. 2021, 7, 1622



Open Reaction Database
J. Am. Chem. Soc. 2021, 143, 18820
Dark Reaction Database
Nature 2016, 533, 73

Chemists try to operate in *heuristically well-understood condition space*, so data generation (even failed reactions) is inherently biased.

HT/automated experimentation coupled with computational simulations is necessary.

What is the point of diminishing returns for “just running the reaction?”

Successful cutting-edge ML applications have unbiased datasets:

1. Sentiment analysis for Amazon reviews
2. Spotify suggestions
3. Gmail spam filtering

What we have not covered

unsupervised learning (PCA, clustering)
data processing and filtering
reinforcement learning

and more:



THIS IS YOUR MACHINE LEARNING SYSTEM?

YUP! YOU POUR THE DATA INTO THIS BIG
PILE OF LINEAR ALGEBRA, THEN COLLECT
THE ANSWERS ON THE OTHER SIDE.

WHAT IF THE ANSWERS ARE WRONG?

JUST STIR THE PILE UNTIL
THEY START LOOKING RIGHT.

