RELAY®
THERAPEUTICS

Two Years of Gaussian Processes

Brian Kelley

Outline

Basic Definitions

Gaussian Process Theory

Gaussian Process in Practice

Error Analysis

What are we modelling?

DMPK

- **Fraction Unbound**
- **Unbound Clearance**
- **Solubility**
- **Mean Residence Time (half life)**

Why?

- **Answers the question: how potent do we need to make our drugs.**
- **If DMPK is not good for a series, should we continue it?**

Aleatoric vs. epistemic uncertainty

- **Aleatoric, /ˌālēəˈtôrik,ˈˌalēəˈtôrik/, *adjective:* depending on the throw of a dice or on chance; random.**
- **Epistemic, /ˌepəˈstemik,ˌepəˈstēmik/, *adjective:* relating to knowledge or to the degree of its validation.**
- **Aleatoric uncertainty is eventually irreducible even with enough data**
  - **Your "ground truth" may have noise**
  - **Your input representation and model assumptions simply can't represent the true function**
- **Epistemic uncertainty comes from insufficient "similar" training data**
- **It's a useful distinction but not completely formal**

- **In drug discovery we have both types of uncertainty:**
  - **Aleatoric: assays are noisy**
  - **Epistemic: the whole patently distinct thing**

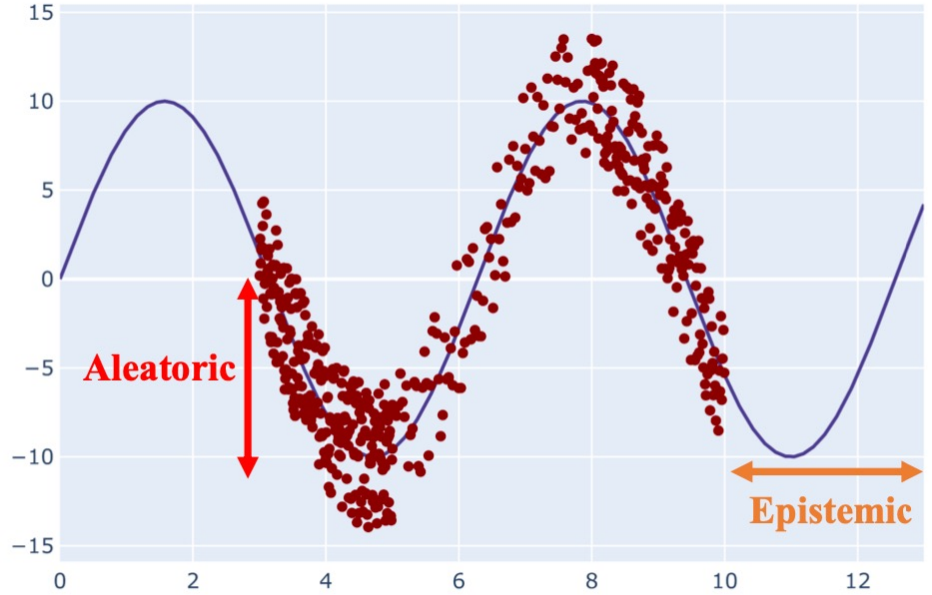


Fig. 1: A schematic view of main differences between aleatoric and epistemic uncertainties.

Abdar, Moloud, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, et al. 2020. "A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/2011.06225.

Gaussian Process

- **Bayesian regression technique**
- **Provides a means of calculating variance**
- **Gaussian Process specified by:**
  - **Mean Value function**
  - **Kernel function (e.g. Quadratic Exponential Kernel, Radial Basis Function)**
- **Kernel (covariance) function captures similarity between data points**
  - **Depends on feature representation**
  - **Chemistry has a slew of them!**

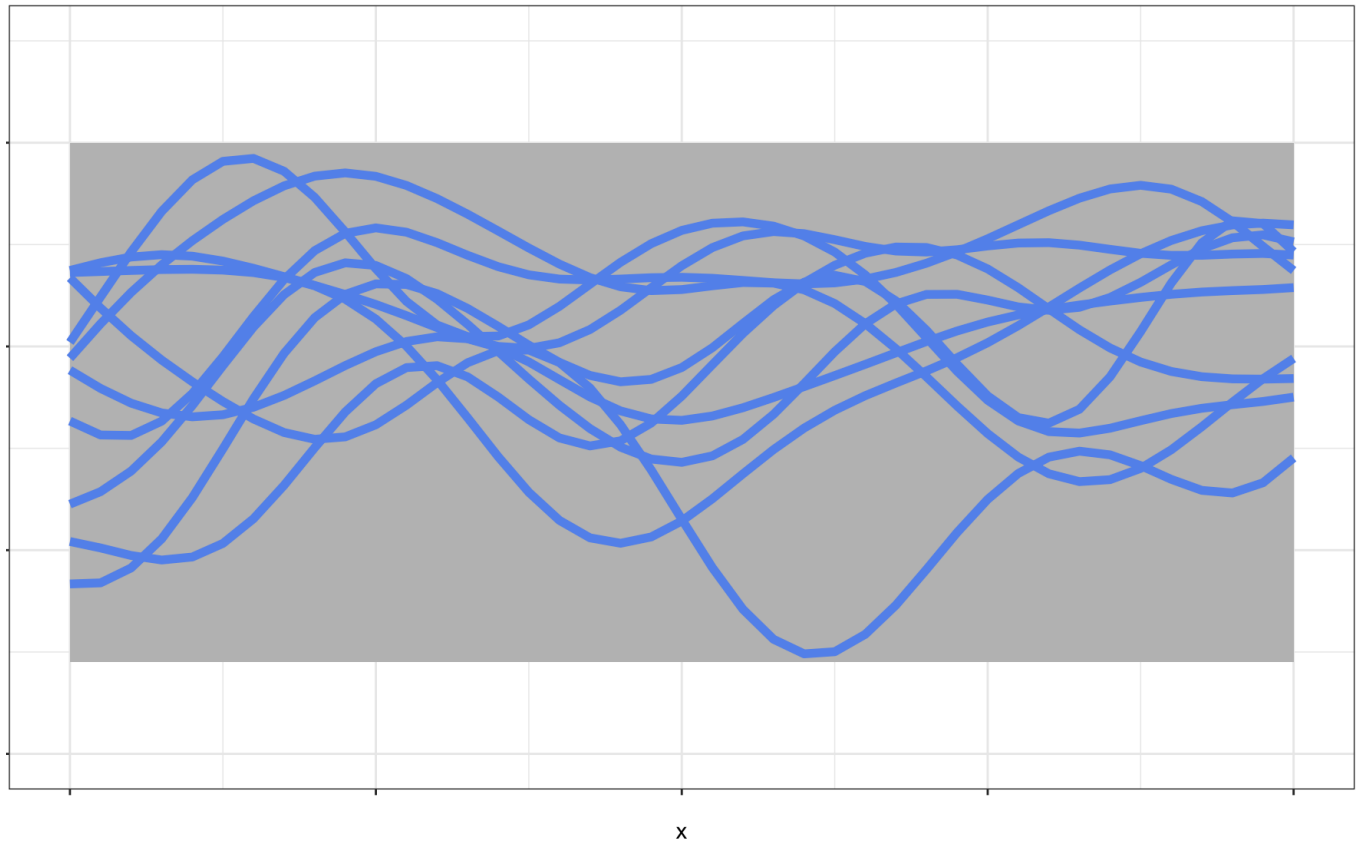
➢ **Covariance function can be a measure of molecular similarity**

Gaussian Process Regression

Think about Gaussian Processes as distributions over functions

Decide on a 'kernel' function, this is your prior distribution of functions

$$f \sim \mathcal{GP}$$

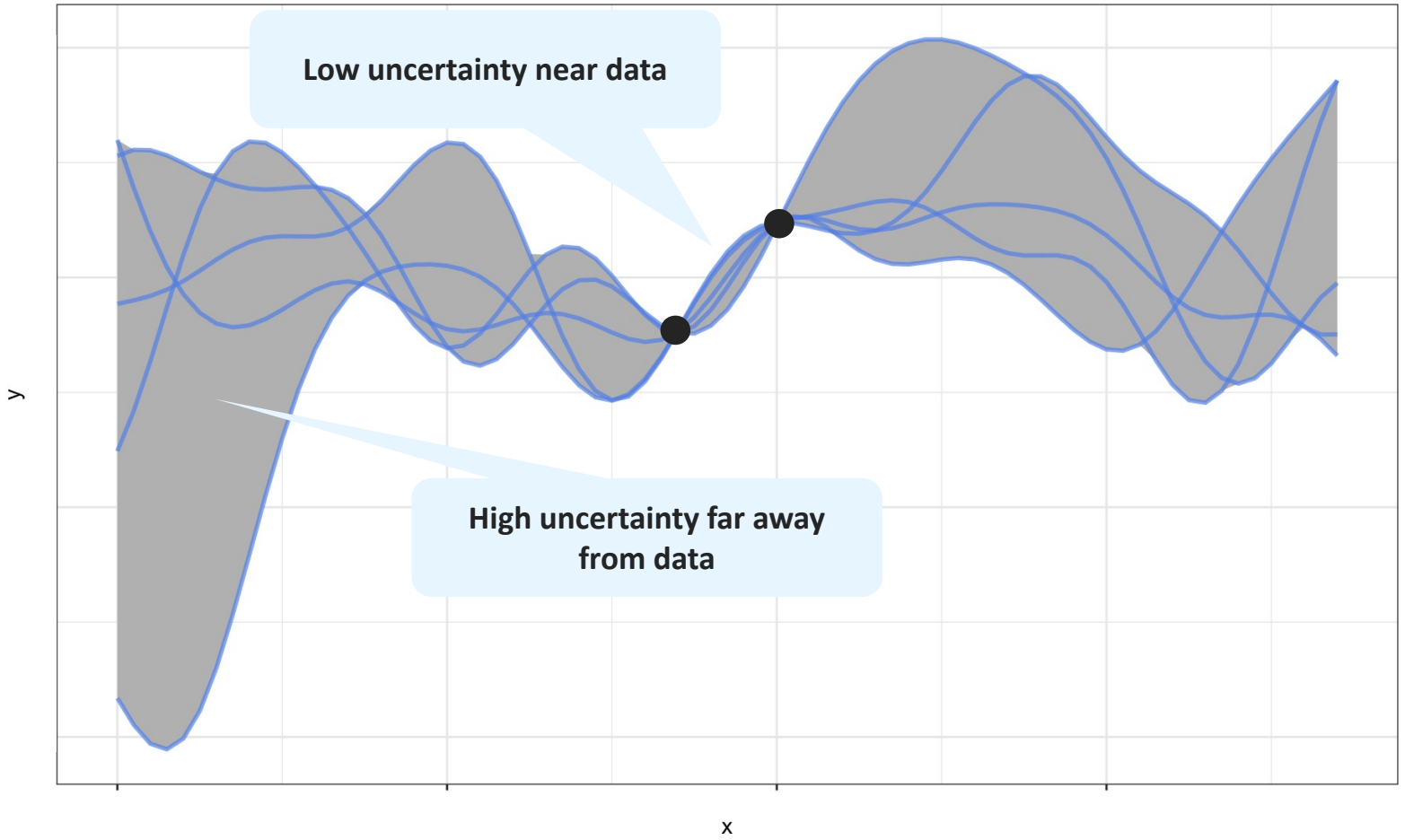


10 samples from a squared exponential prior

[Iain Murray, CSC2515 class slides](#)

Gaussian Process Regression

- **Select functions from prior that agree with seen data**



5 samples from posterior after seeing 2 points

Tanimoto Kernel

Using Ryan-Rhys Griffith's kernel and GPFlow 2.0

https://github.com/Ryan-Rhys/The-Photoswitch-Dataset/blob/master/examples/gp_regression_on_molecules.ipynb

$$k_{Tanimoto}(\mathbf{x}, \mathbf{x}') = \frac{\sigma < \mathbf{x}, \mathbf{x}' >}{||\mathbf{x}||^2 + ||\mathbf{x}'||^2 - < \mathbf{x}, \mathbf{x}' >}$$

Easy, right?

https://gpflow.github.io/GPflow/2.6.2/index.html

https://towardsdatascience.com/gaussian-process-regression-on-molecules-in-gpflow-ee6fedab2130

Tanimoto Kernel

Using Ryan-Rhys Griffith's kernel

$$k_{Tanimoto}(\mathbf{x}, \mathbf{x}') = \frac{\text{Intersection of A and B}}{\text{Union of A and B}}$$

For Fingerprints

$$\frac{\text{Bits in common between A and B}}{|\text{Bits in A}| + |\text{Bits in B}| - \text{Bits in Common Between A and B}}$$

https://towardsdatascience.com/gaussian-process-regression-on-molecules-in-gpflow-ee6fedab2130

Tanimoto Kernel

Using Ryan-Rhys Griffith's kernel

X is now the input matrix not a single vector (X==X2 in our case)

mol1_bit1 mol1_bit2 mol1_bit3 ….
mol2_bit1 mol2_bit2 mol2_bit3 ….
…

https://towardsdatascience.com/gaussian-process-regression-on-molecules-in-gpflow-ee6fedab2130

Tanimoto Kernel

Using Ryan-Rhys Griffith's kernel (compute Tanimoto over matrix input)

$||\mathbf{x}||^2$ **Xs = tf.reduce_sum(tf.square(X), axis=-1) # Squared L2-norm of X**

$||\mathbf{x}'||^2$ **X2s = tf.reduce_sum(tf.square(X2), axis=-1) # Squared L2-norm of X2**

$< \mathbf{X}, \mathbf{X}' >$ **outer_product = tf.tensordot(X, X2, [[-1], [-1]]) # outer product of the matrices X and X2**

https://towardsdatascience.com/gaussian-process-regression-on-molecules-in-gpflow-ee6fedab2130

Tanimoto Kernel

Using Ryan-Rhys Griffith's kernel

$$X = X2 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Molecule 1

Molecule 2

Molecule 3

$||\mathbf{x}||^2 = [2\ 3\ 1]$ i.e. bits set in mol1, bits set in mol2

$$\langle \mathbf{x}, \mathbf{x}' \rangle \quad \begin{bmatrix} 2 & 2 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Symmetric Matrix, bits in common between I and j

https://towardsdatascience.com/gaussian-process-regression-on-molecules-in-gpflow-ee6fedab2130

Tanimoto Kernel

Using Ryan-Rhys Griffith's kernel (compute Tanimoto over matrix input)

$||\mathbf{x}||^2$ **Xs = tf.reduce_sum(tf.square(X), axis=-1) # Squared L2-norm of X**

$||\mathbf{x}'||^2$ **X2s = tf.reduce_sum(tf.square(X2), axis=-1) # Squared L2-norm of X2**

$< \mathbf{x}, \mathbf{x}' >$ **outer_product = tf.tensordot(X, X2, [[-1], [-1]]) # outer product of the matrices X and X2**

$$||\mathbf{x}||^2 + ||\mathbf{x}'||^2 - < \mathbf{x}, \mathbf{x}' >$$

denominator = -outer_product + broadcasting_elementwise(tf.add, Xs, X2s)

kernel = self.variance * outer_product / denominator

Variance is what is being trained (in addition to noise)

This just makes tensorflow fast

https://towardsdatascience.com/gaussian-process-regression-on-molecules-in-gpflow-ee6fedab2130

We can also add RDKit Features

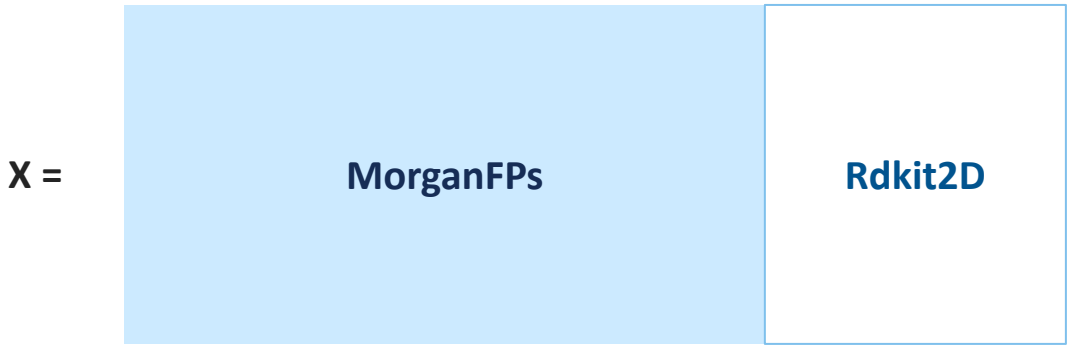
RDKit Features are scaled to CDFs (continuous density functions)

- a property value of 0.5 means that half the population lies below this value.

```
from descriptastorus import descriptors
from tanimotogp_example import TanimotoGaussianProcess()

generator = descriptors.MakeGenerator(["Morgan3", "RDKit2DHistogramNormalized"])
features = generator.processSmiles(list_of_smiles)
```

Note – we are using Counts not Binary Morgans!
Tanimoto is still well defined

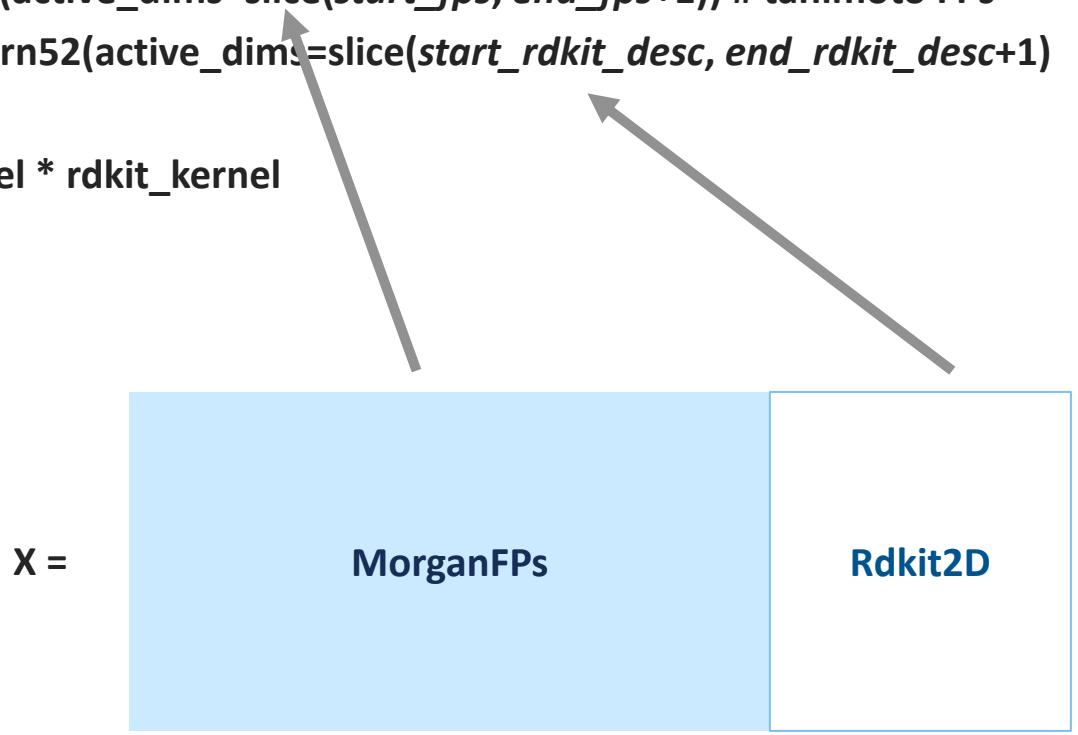


In addition, we can add RDKit features!

We add the RDKit 2D features to the mix by making a second kernel and optimizing on both.

```
tanimoto_kernel = TanimotoKernel(active_dims=slice(start_fps, end_fps+1)) # tanimoto FPs
rdkit_kernel = gpflow.kernels.Matern52(active_dims=slice(start_rdkit_desc, end_rdkit_desc+1)

combined_kernel = tanimoto_kernel * rdkit_kernel
```



Is it better?

What I care about is predicting the future.

Question: are Gaussian Processes better with future predictions?

Experiment:

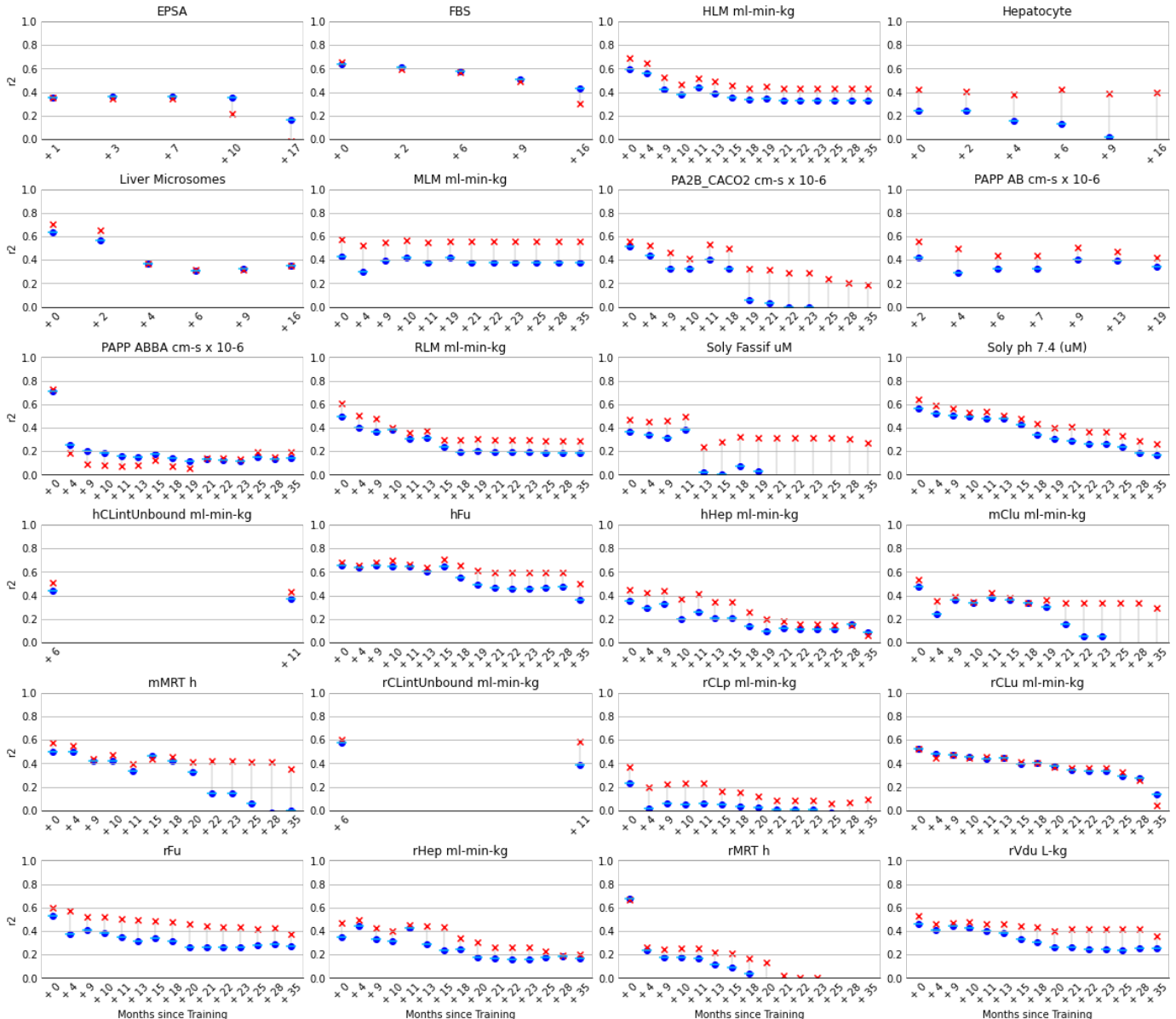
Make a XGB Model and a Gaussian Process model

run that model on the next two years of collected data

Is it better?

Predictions Made N Months after training,

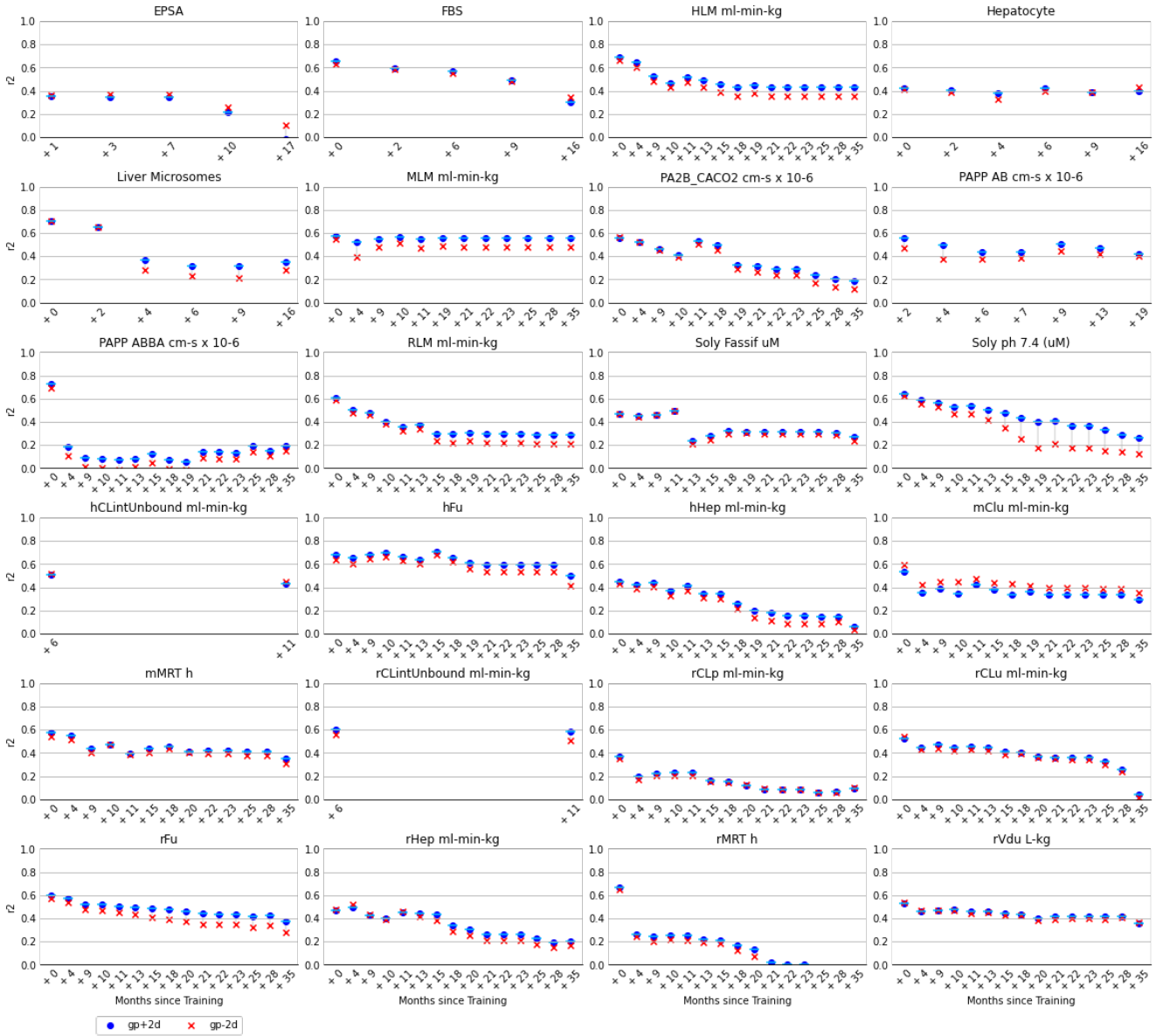
XGBoost versus Gaussian Processes



Did the combined kernel help?

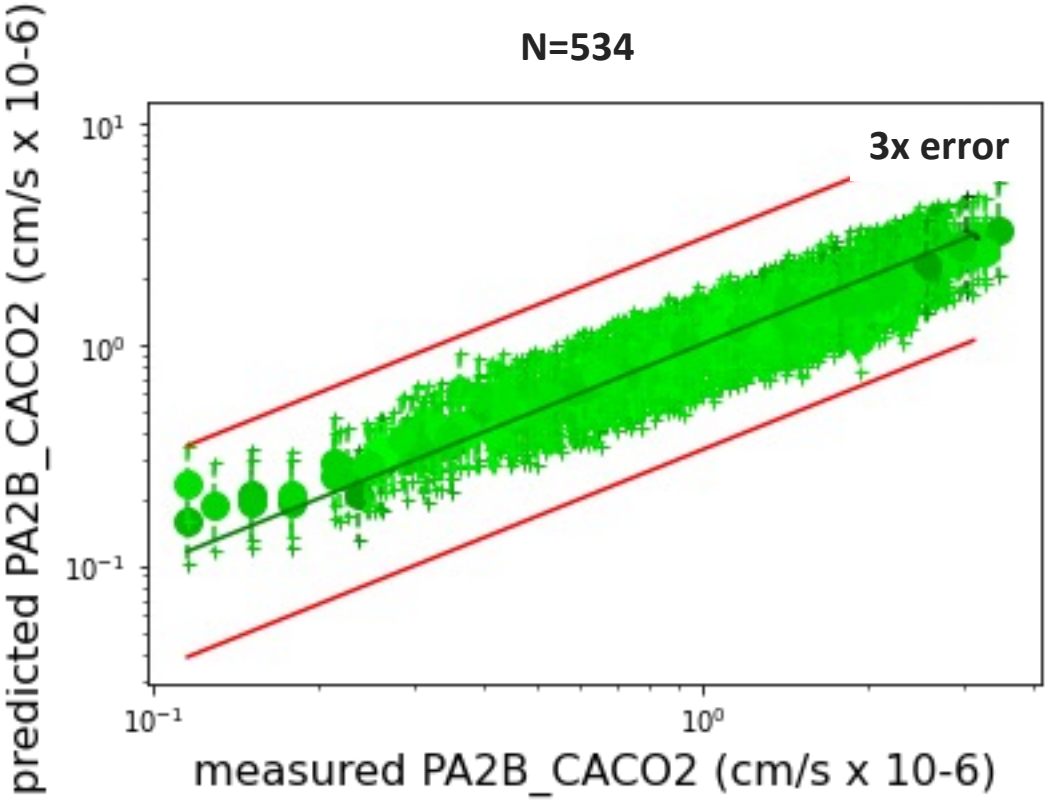
Predictions Made N Months after training,

With 2D Features, without 2D features.



Gaussian Uncertainty in Practice

Gaussian 95% Confidence Interval



Notes

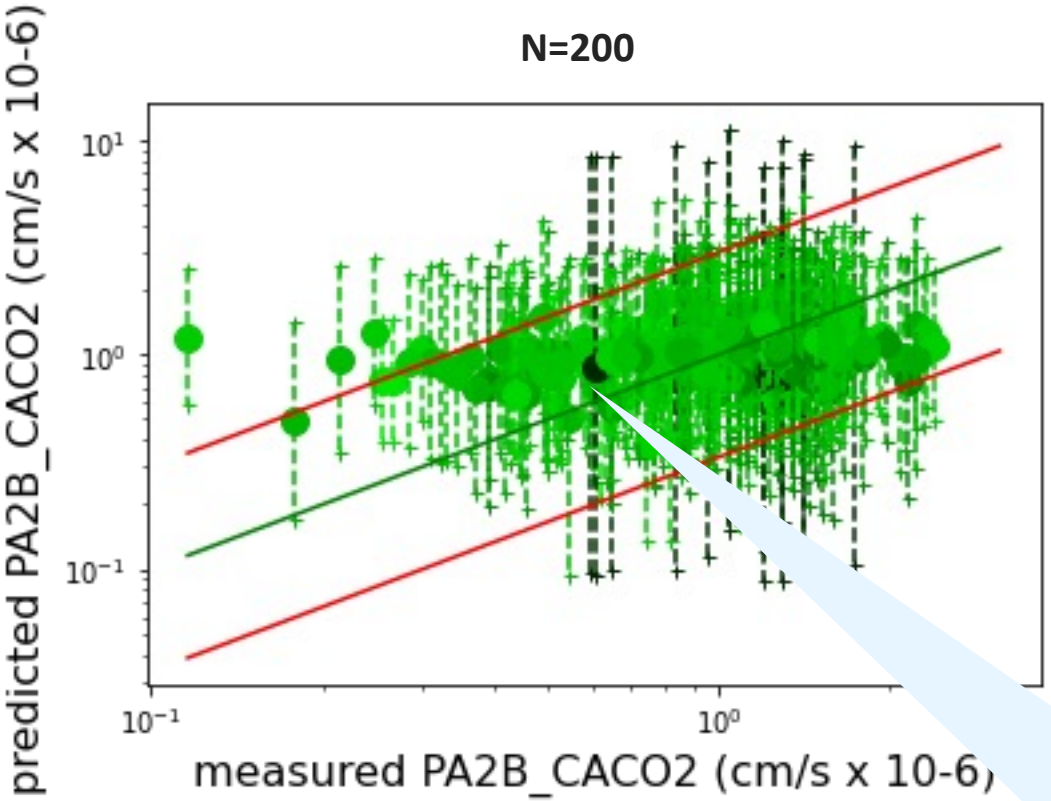
The training set predictions look pretty good

The 95% prediction error is within 3x error, yay?

Past performance is no guarantee of future success

New Data For three months post model build

N=200

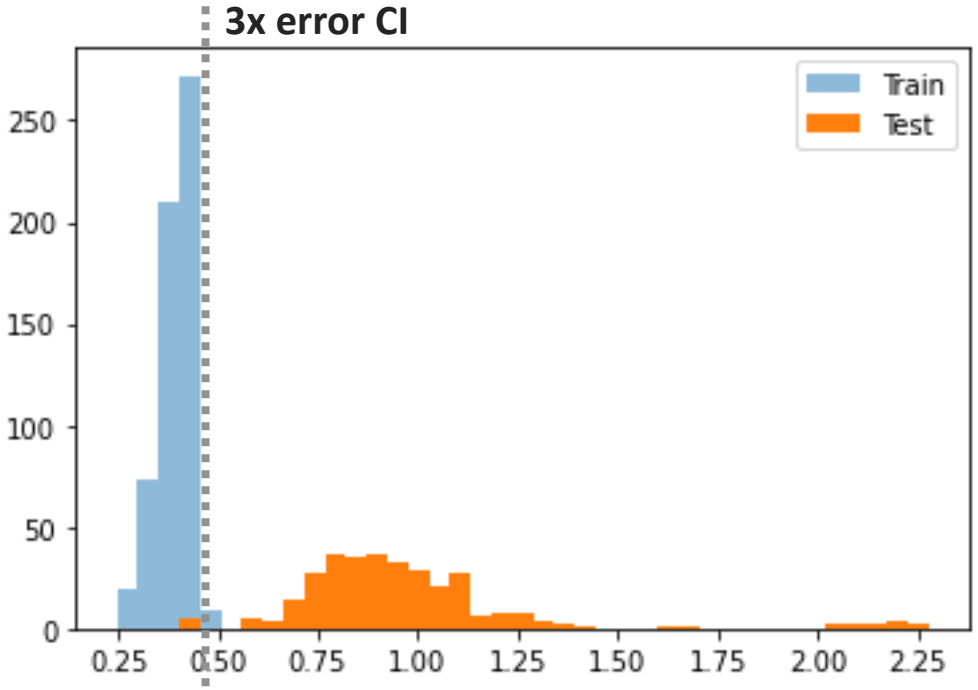


Notes

Yikes.

At least the model errors are bigger.

3x error CI

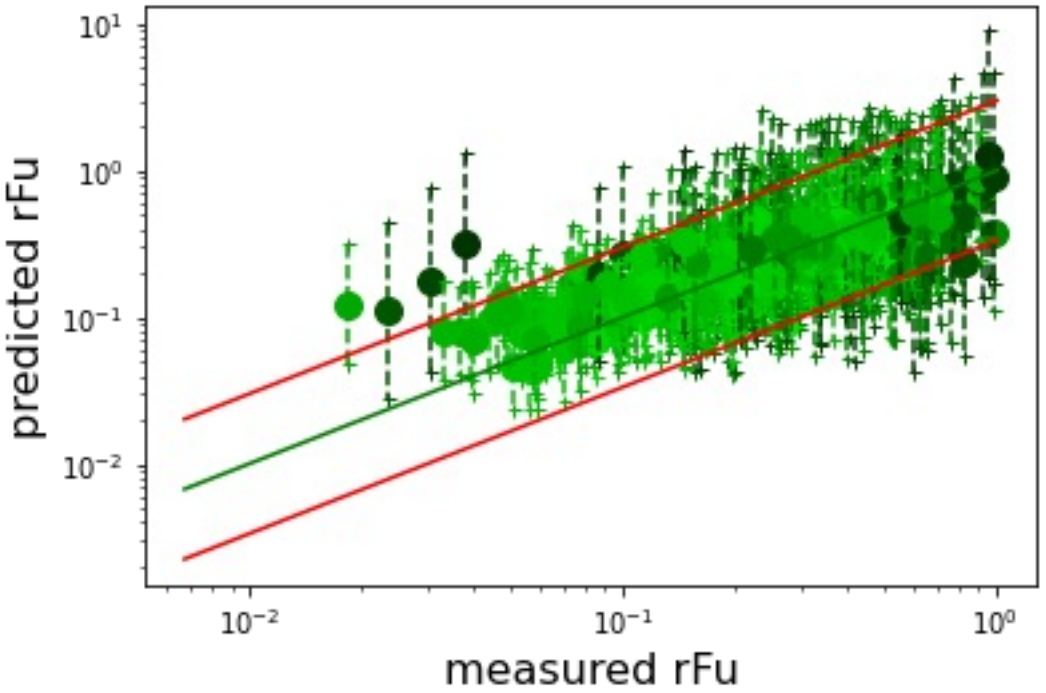


Dark points are dissimilar via morgan fingerprints

Slightly better performing model (rFu)

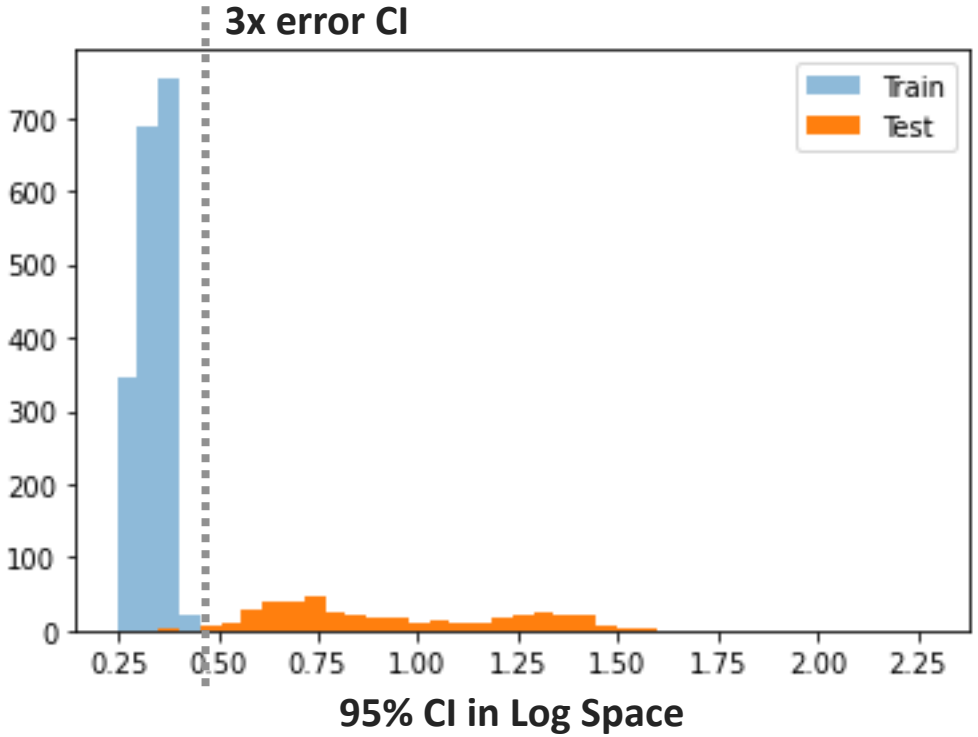
New Data For three months post model build

N=200



Notes

Model errors are more reasonable, but what's up with the bimodal distribution?

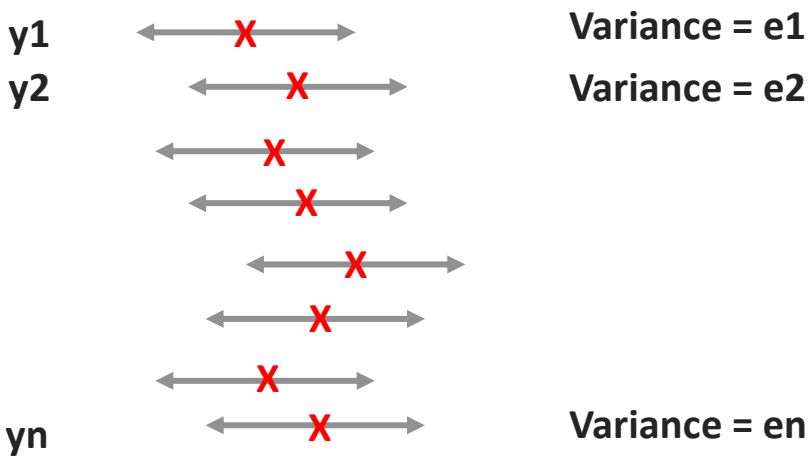


Can we estimate odds of being within 3x?

JackKnife+

Train a bunch of models

- 100 Models with 70% train test split
- Compare test to get observed distributions (i.e. prediction versus experiment)

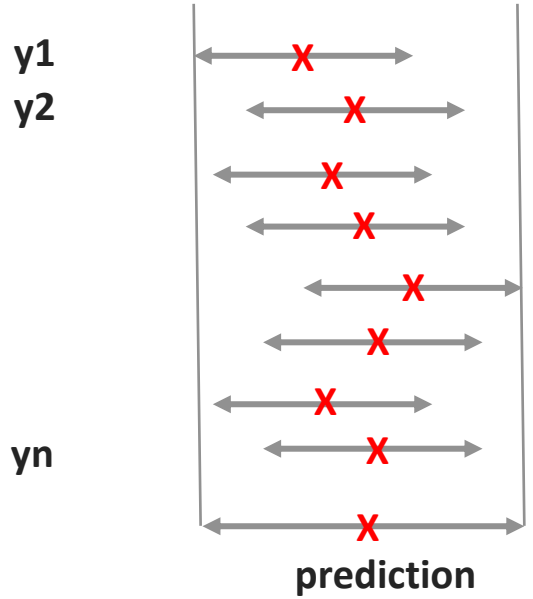


Predictive inference with the jackknife+

Rina Foygel Barber*, Emmanuel J. Candès†, Aaditya Ramdas‡, Ryan J. Tibshirani‡§

Test a new sample

- Run on all models and use the model variance as the models Confidence interval
- Use the combined confidence



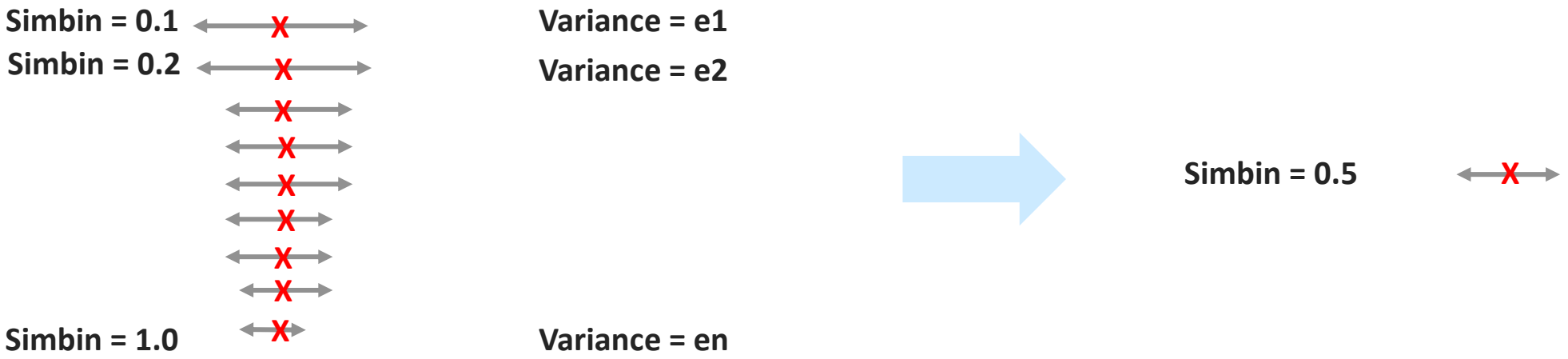
Cross-Conformal prediction using similarity bagging

Train a bunch of models

- **(bootstrapping with train test split),**
- **Get variances (but bin by similarity)**
- **Train on all data to make model**

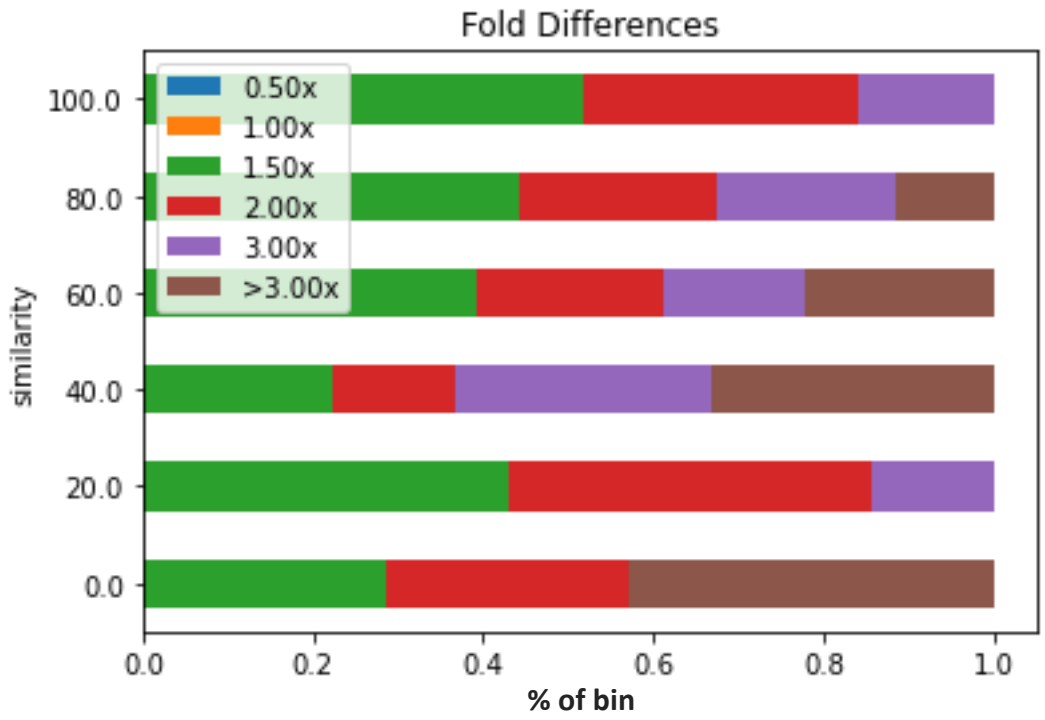
Test a new sample

- **Make prediction**
- **Use variance correlated to similarity bin**

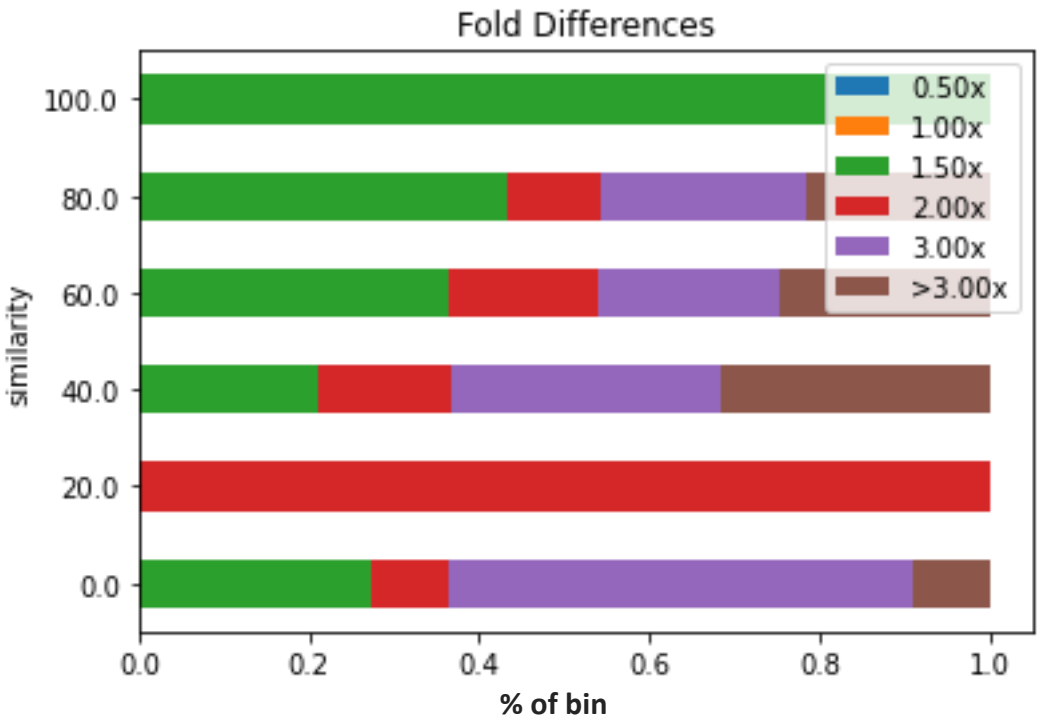


Caco2 similarity conformal analysis

Test and train distributions look similar (N is small in test)

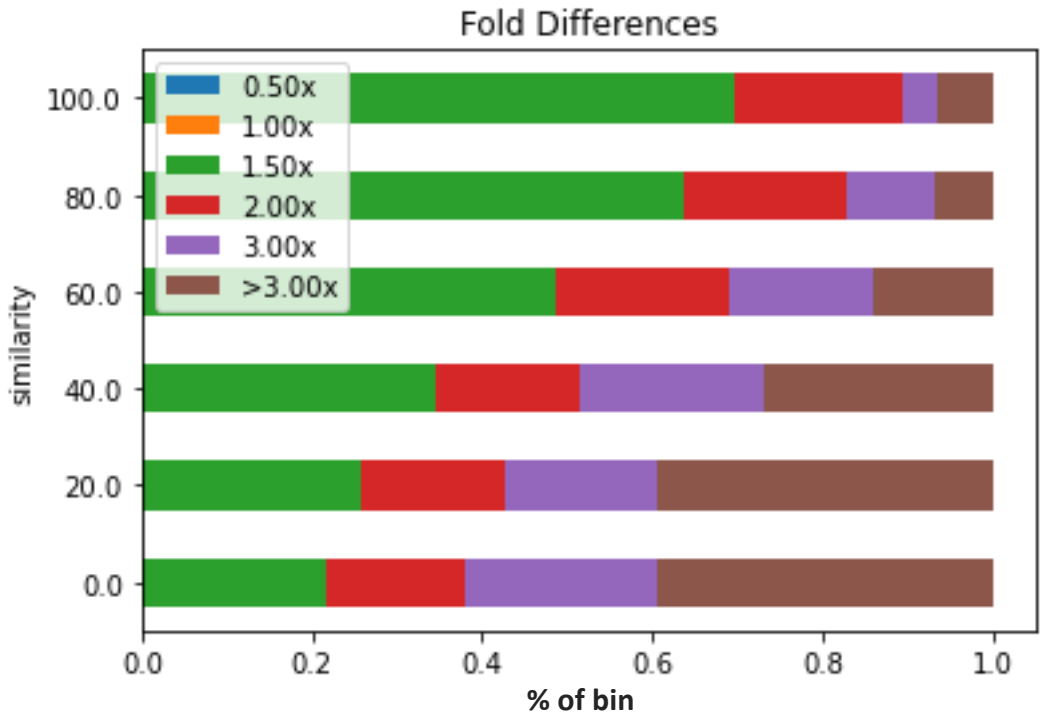


Train

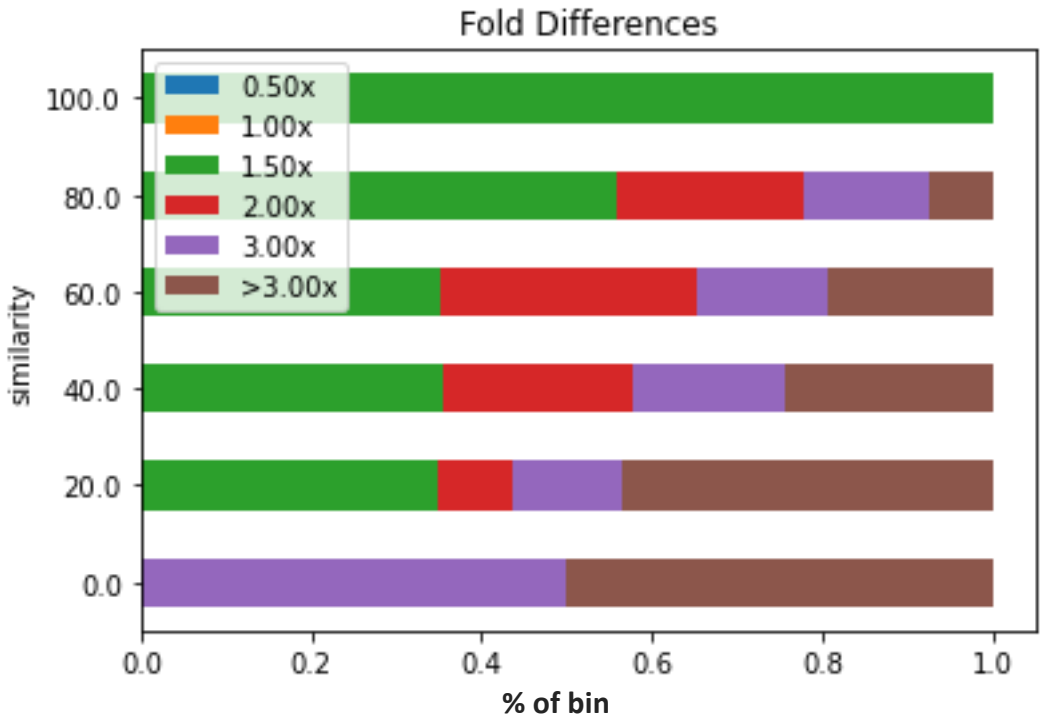


Test

rFu similarity conformal analysis



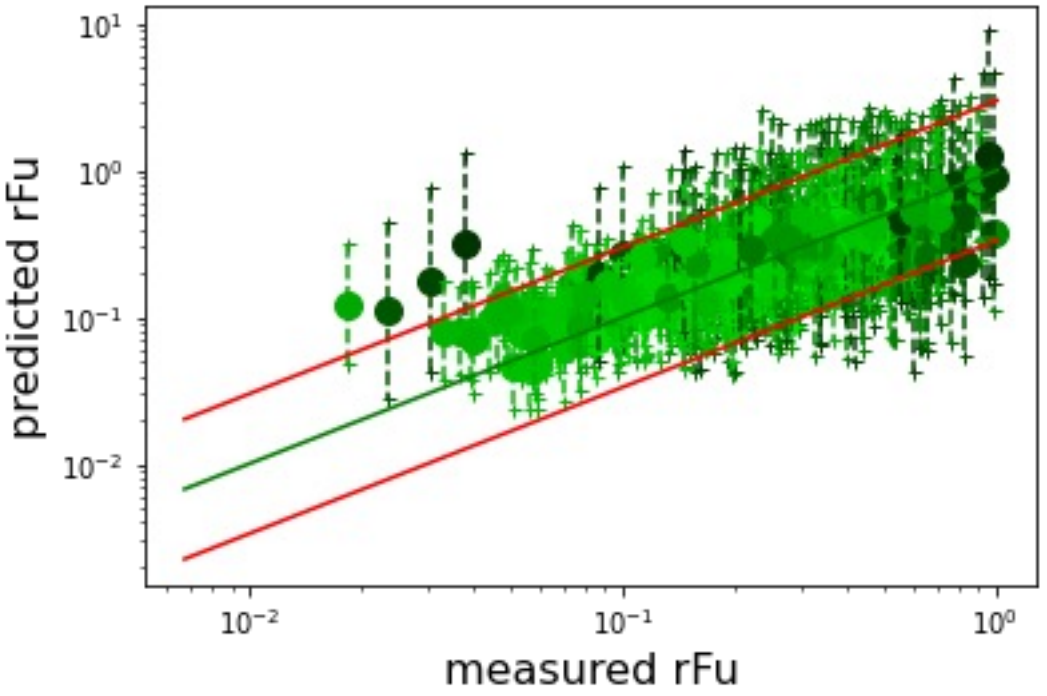
Train



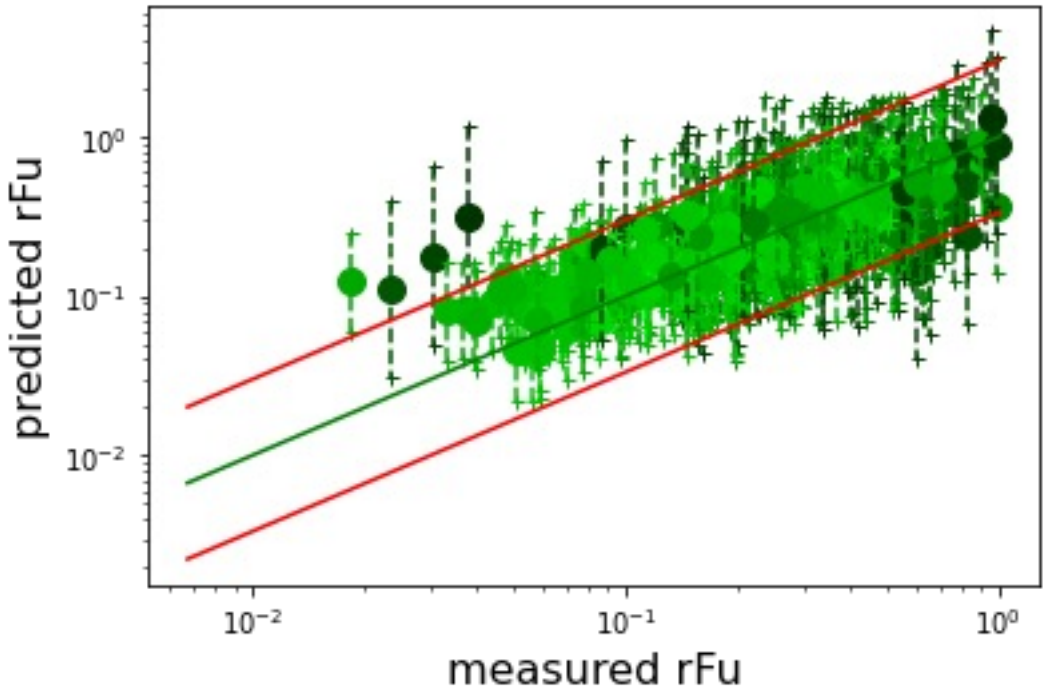
Test

Turns out, that this is virtually identical to Gaussian Process Errors!

Gaussian Process Error



simbin (6 bins)



To Recap

Gaussian Processes are more consistent in predicting DMPK assays than XGB

Changing confidence intervals into predicted %percent below 1.5x and %percent below 3.0x seems to fit chemists and modelers brains better.

BUT:

The best use of confidence intervals is getting low confidence molecules tested!

RELAY®

THERAPEUTICS