Optimal Learning for Discovering Minimal Peptide Substrates

Jialei Wang, Pu Yang, Peter Frazier (Cornell ORIE)
in collaboration with: Michael Burkart, Nathan Gianneschi, Michael Gilson, Nick Kosa, Michael Rothmann, Lorillee Tallorin (UCSD)

Thursday February 6, 2014
Information Science for Materials Discovery and Design Workshop, Sante Fe, New Mexico
Funded by AFOSR FA9550-12-1-0200

The trip was a bit complicated, but it’s good to be here.

We use optimal learning to create a new biomaterial

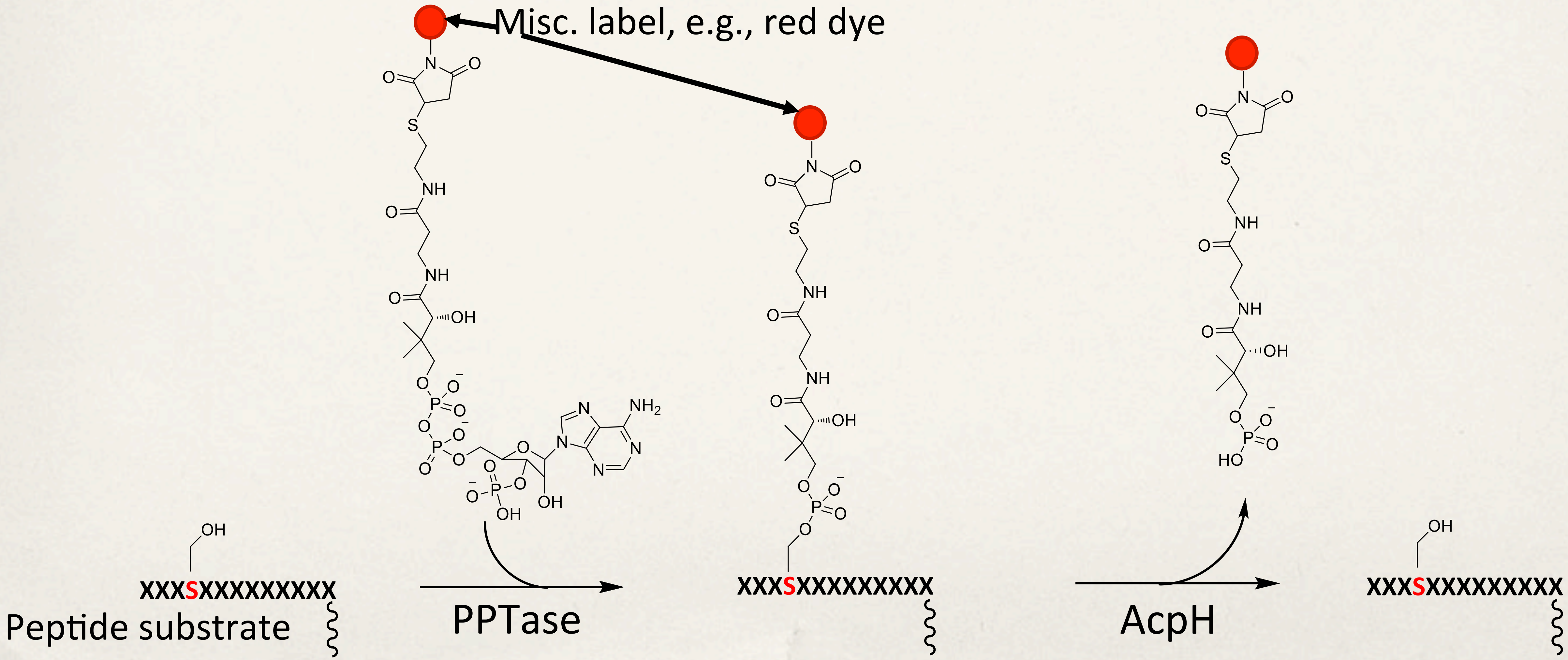
- Goal: find a **short** peptide that allows a certain pair of enzymatic reactions to occur.

- We use Bayesian statistics and value of information analysis to suggest which experiments to perform to find such a peptide.

- Our collaborators (all at UCSD): Mike Burkart, Nathan Gianneschi, Mike Gilson, Nick Kosa, Mike Rothmann, Lori Tallorin.

Our goal:
find a short peptide that is a substrate for two enzymes

- These enzymes are:
  - PPTase: attaches an arbitrary label to the peptide substrate, at a conserved serene.
  - AcpH: removes the label

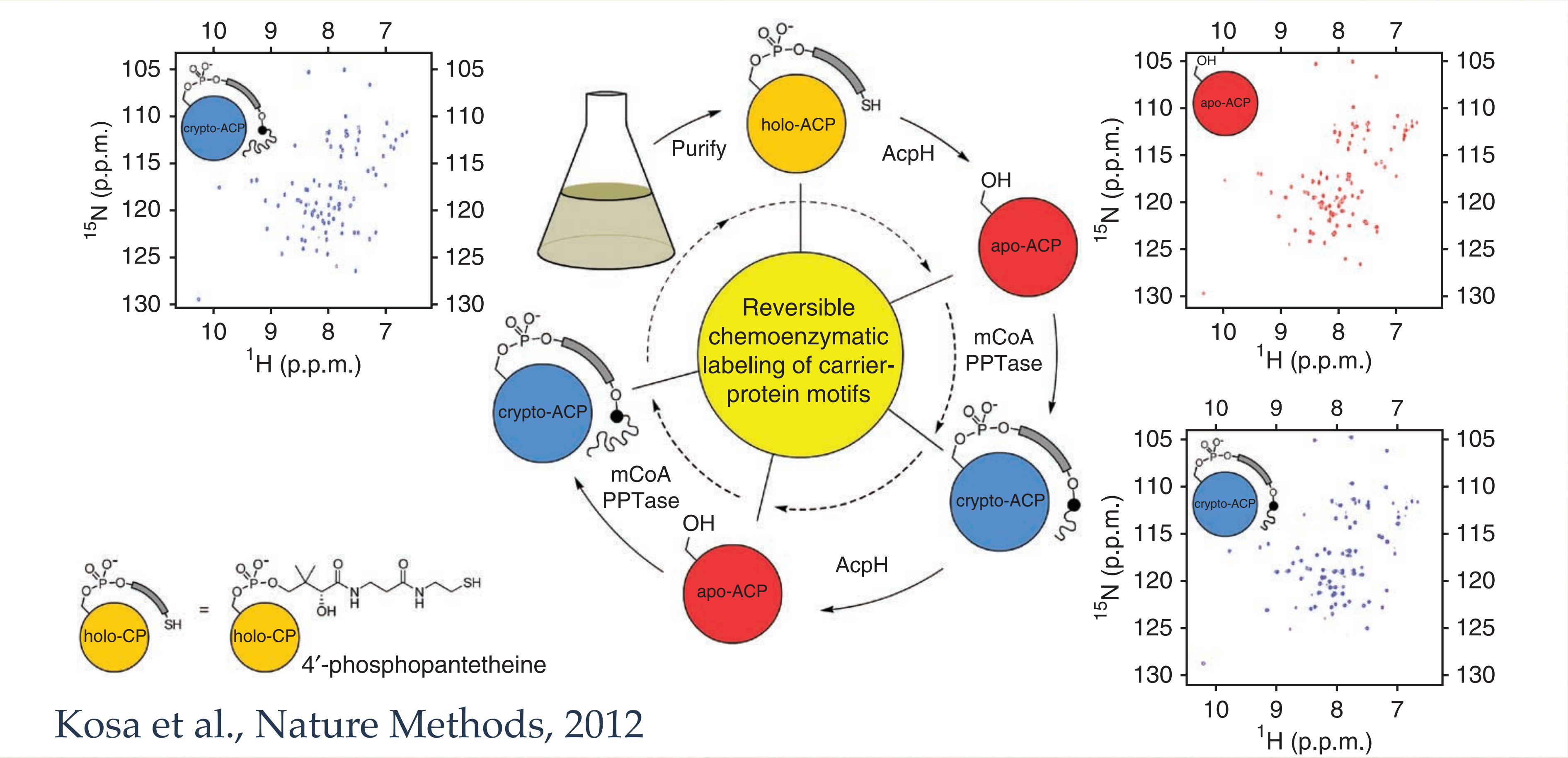


Misc. label, e.g., red dye

Finding this peptide will allow creating versatile building blocks for biomaterials

- Finding a short peptide with this property will allow our collaborators to add & subtract functionality from a protein with the peptide embedded inside it.

- This could be used to create novel sensors, study protein interactions, and will be used within a combinatorial chemistry search for CO2 reducing catalysts.



Kosa et al., Nature Methods, 2012

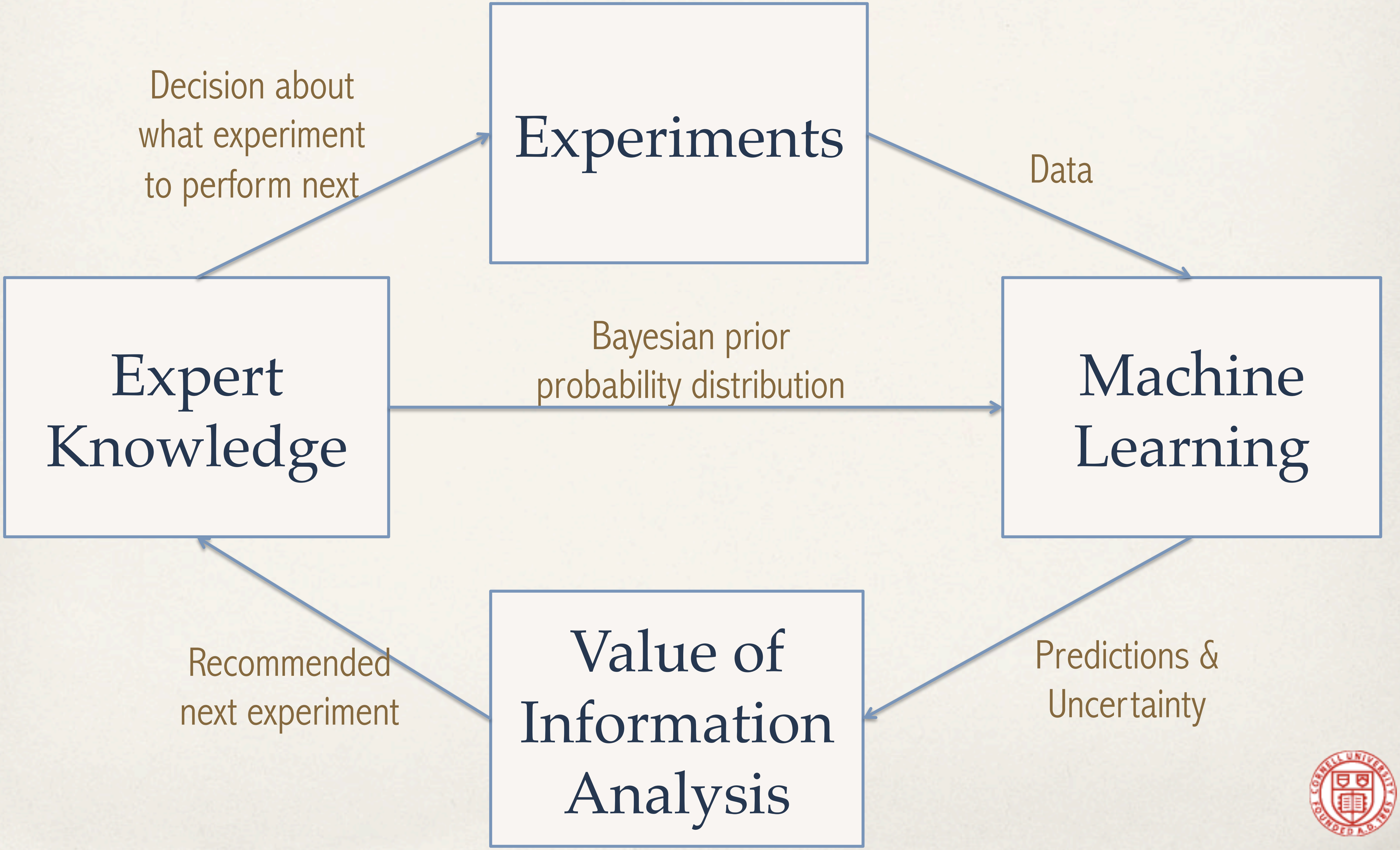
It is hard to find short hits; Math makes it easier.

- If a peptide allows both chemical reactions to occur, we say it is a "hit".
- Hits are rare: about 1 in 10^5 among shorter peptides.
- Testing peptides is expensive & time-consuming: it requires reserving time on an expensive capacity-limited time machine, about 1 week's worth of work by an experimentalist; and material costs.
- We test 500 peptides at time. 500 is much smaller than 10^5.
- To help us, we have some known hits, obtained from natural organisms. They are too long to be used directly.

We reduce the experimental effort required to find minimal substrates

- We provide a method for Peptide Optimization with Optimal Learning (**POOL**).
- Our method has two parts:
  - Predict which peptides are "hits".
  - Based on these predictions, recommend which peptides to test next.
- Our approach is similar to active learning in CS.

Peptide Optimization with Optimal Learning (POOL)



Decision about what experiment to perform next

Experiments

Data

Expert Knowledge

Bayesian prior probability distribution

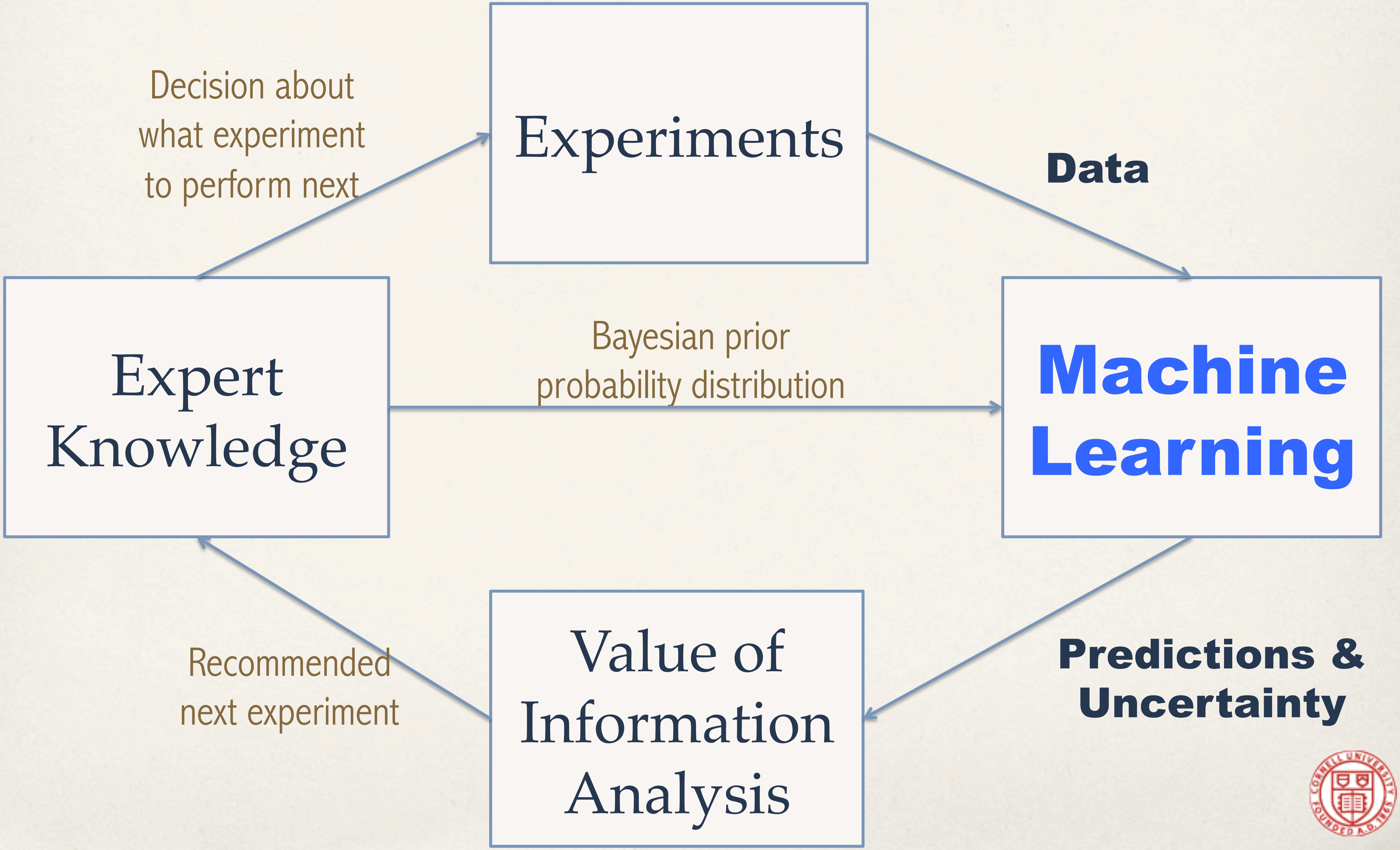
Machine Learning

Recommended next experiment

Value of Information Analysis

Predictions & Uncertainty

First, we consider prediction.



Experiments
Data
Machine Learning
Predictions & Uncertainty
Value of Information Analysis
Recommended next experiment
Expert Knowledge
Decision about what experiment to perform next
Bayesian prior probability distribution

We use Naive Bayes

- Naive Bayes is a statistical model often used for text classification (e.g., spam filters).
  - It is called "naive" because it makes a key independence assumption.
  - Although it is naive, it often works really well.
- We apply a variant of Naive Bayes to our problem, which is customized to include amino acids' **location** within the peptide.

We use Naive Bayes

- We assume that reality is characterized by a pair of latent matrices, called $\theta^{(\text{hit})}$and $\theta^{(\text{miss})}$, where columns of each matrix correspond to different positions within the peptide, and rows correspond to different types of amino acids.

- These latent matrices are unknown, but can be estimated from data.

- We further suppose that, for a peptide x,

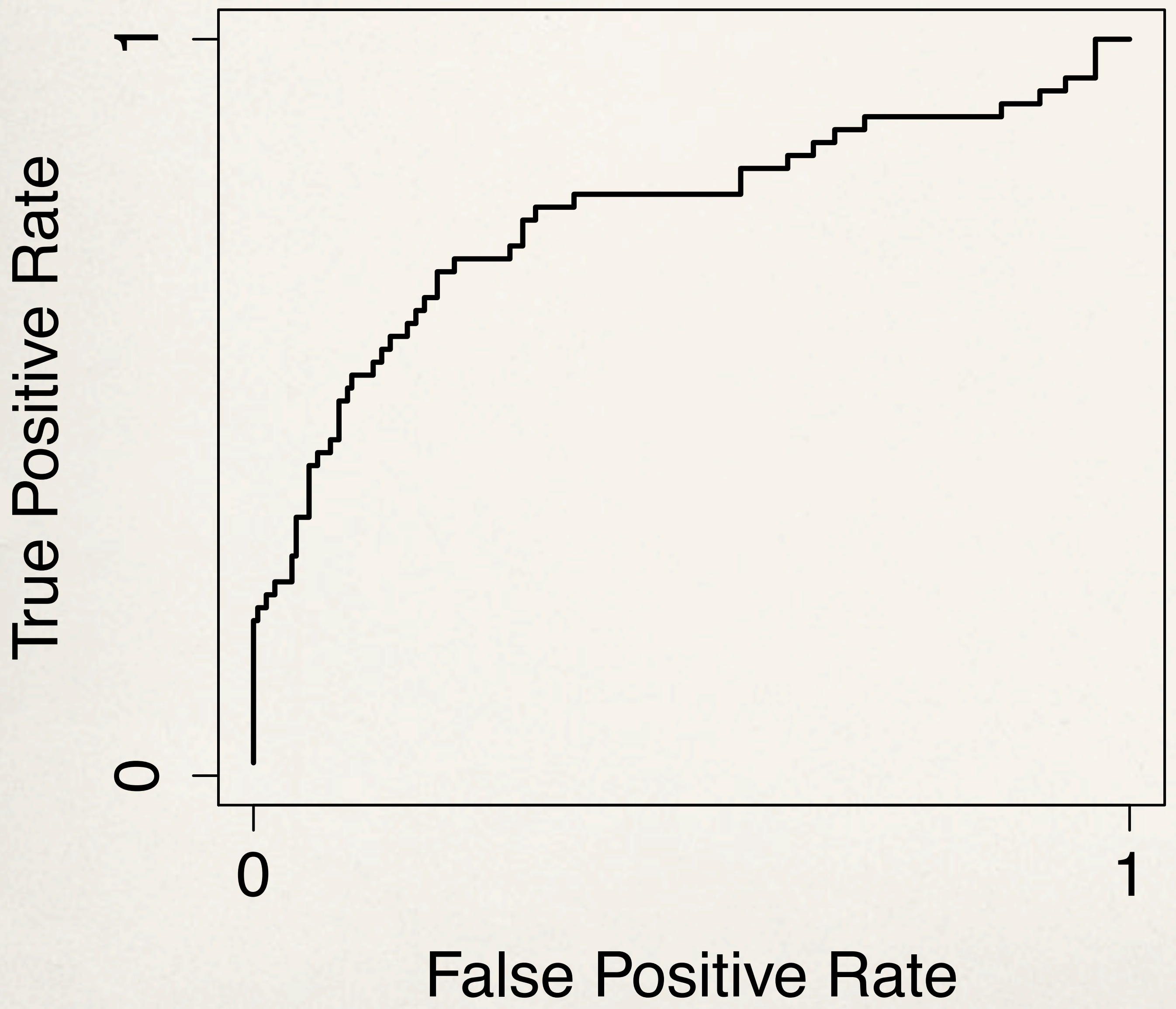
$$P(y(x)=1|x,\theta^{\text{hit}},\theta^{\text{miss}})=\frac{P(\text{hit})\prod_i \theta^{(\text{hit})}_{i,x_i}}{P(\text{hit})\prod_i \theta^{(\text{hit})}_{i,x_i}+P(\text{miss})\prod_i \theta^{(\text{miss})}_{i,x_i}}$$

- Here, x is a peptide, x_i is the type of the amino acid at position i, y(x) indicates whether x is a hit (1) or not (0), and P(hit) and P(miss) are prior estimates of the fraction of hits and misses in the population.

We use Bayesian Naive Bayes

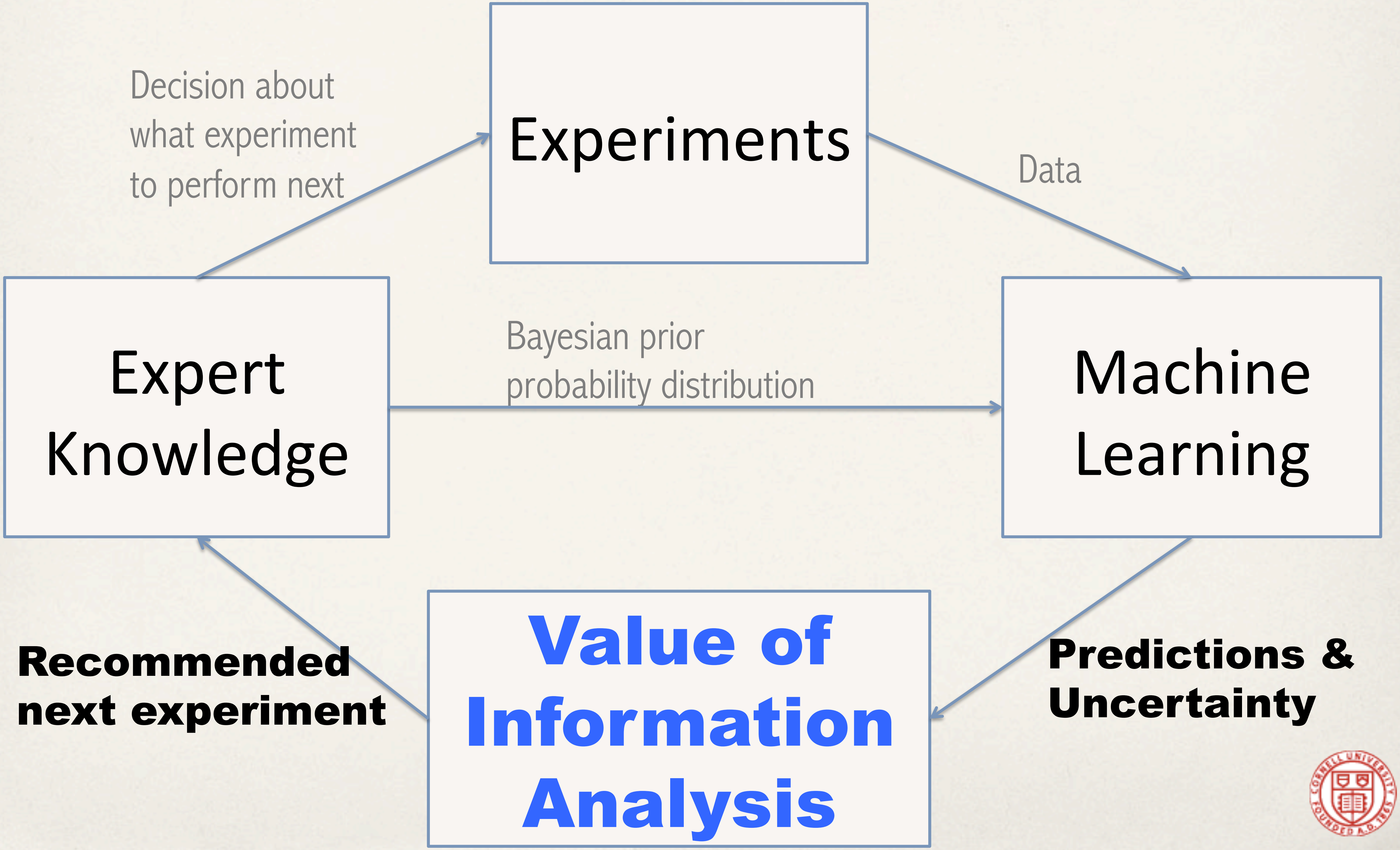
- We put independent Dirichlet prior distributions on each column of the latent matrices $\theta^{(\mathrm{hit})}$ and $\theta^{(\mathrm{miss})}$.
- Our choices for the parameters of this prior are based on a biological understanding of the problem, discussions with our collaborators, and cross validation.
- Given training data $x^1,...,x^n$, $y(x^1),...,y(x^n)$, the posterior on the θ's is also Dirichlet, and independent across i and j.
- To estimate the posterior probability of a hit, we can sample the thetas from the posterior, or calculate a single MAP estimate. The MAP estimate ignores uncertainty, but can be computed analytically.

This ROC curve suggests Naive Bayes performs reasonably well



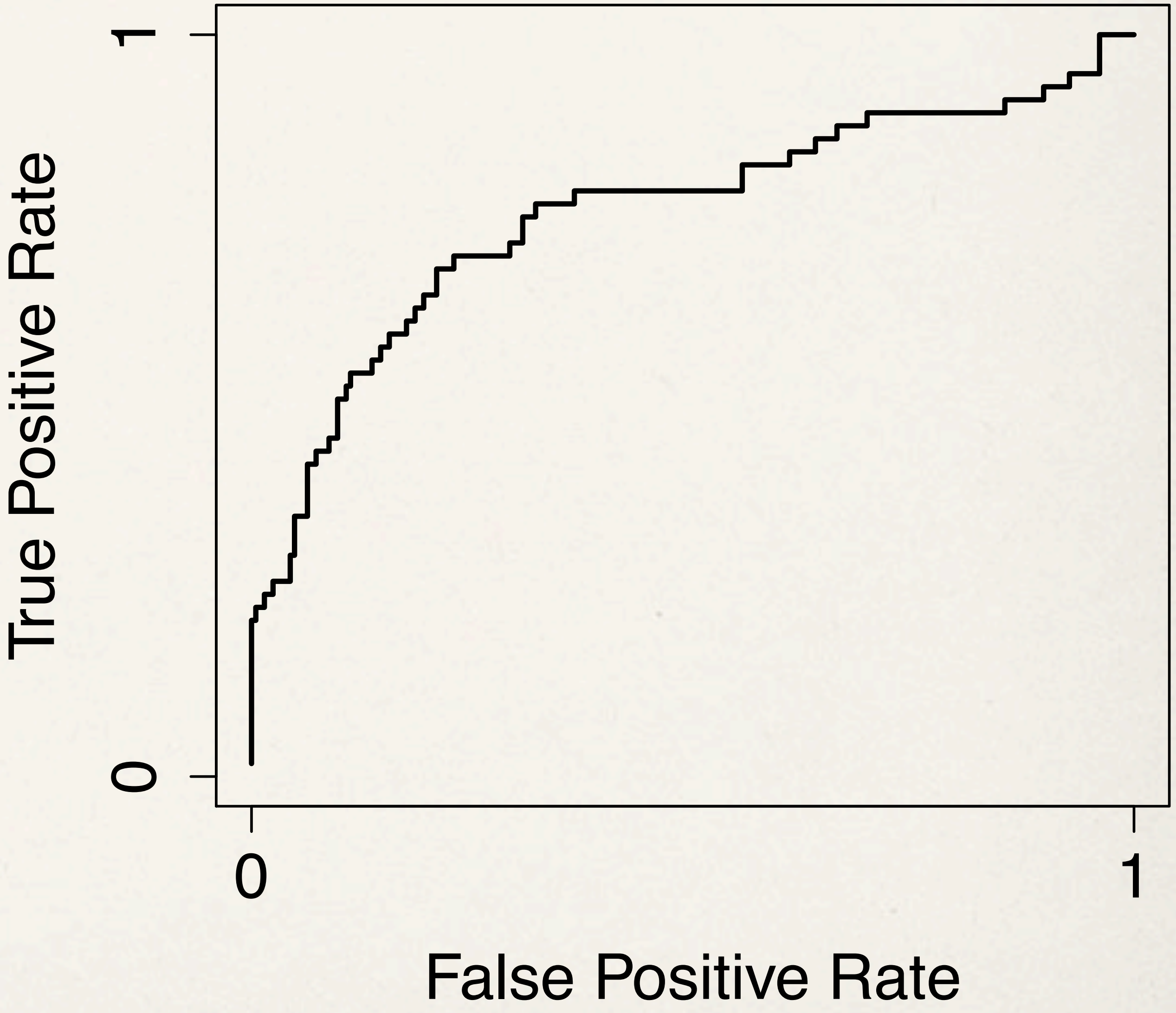
- We have training data for approximately 300 peptides (most are misses.)
- True positive rate = % of hits labeled as hits.
- False positive rate = % of misses labeled as hits.
- Rates were estimated via leave-one-out cross-validation.

Now, we consider the choice of experiment


Experiments
Data
Machine Learning
Predictions & Uncertainty
Value of Information Analysis
Recommended next experiment
Expert Knowledge
Decision about what experiment to perform next
Bayesian prior probability distribution

Given imperfect predictions, what should we test next?

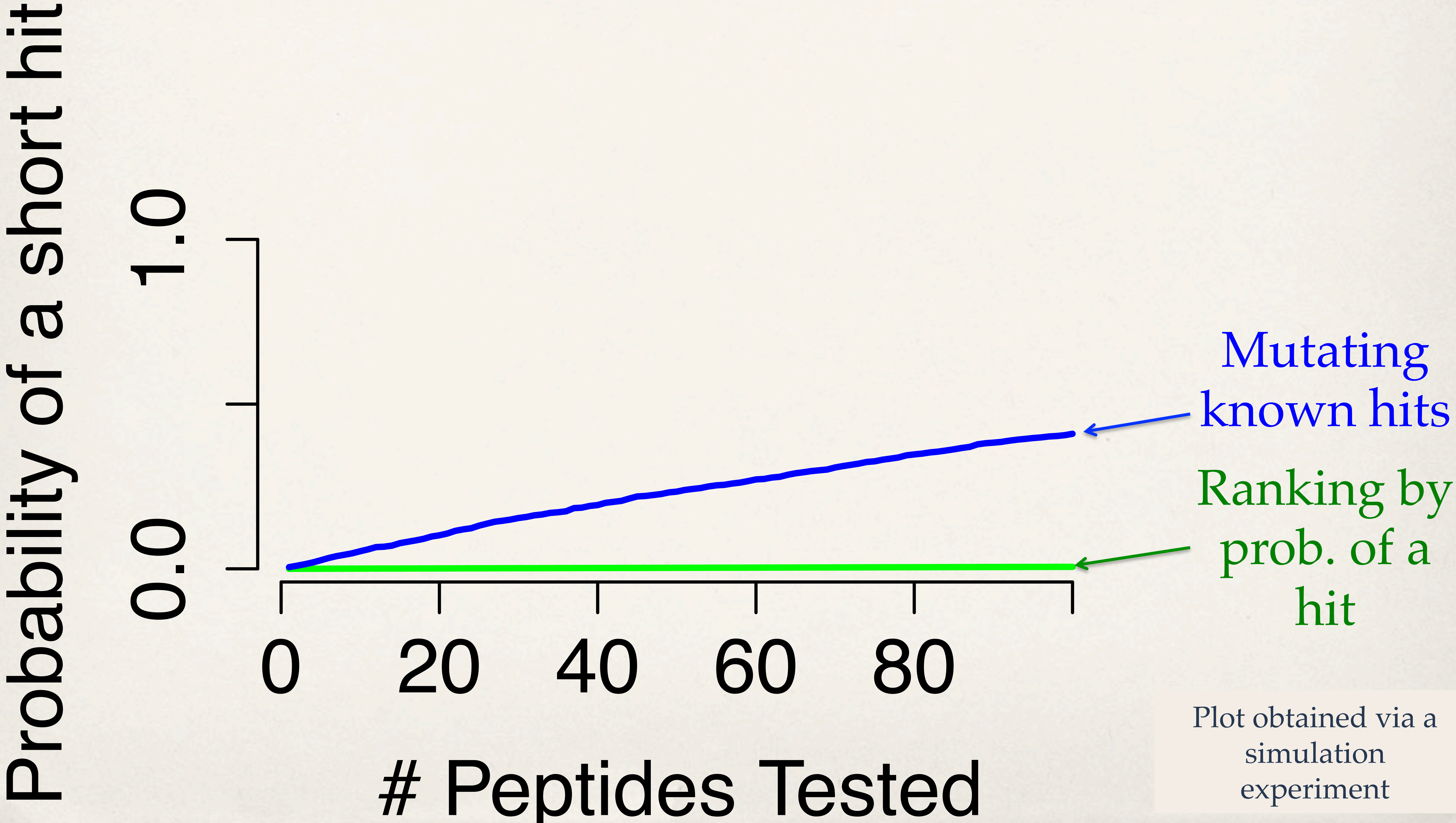
- If predictions were perfect, we could just test the shortest peptide predicted to be a hit.
- Our predictions are not perfect.
- How should we decide what to test next?



Ranking by probability of a hit does not work well

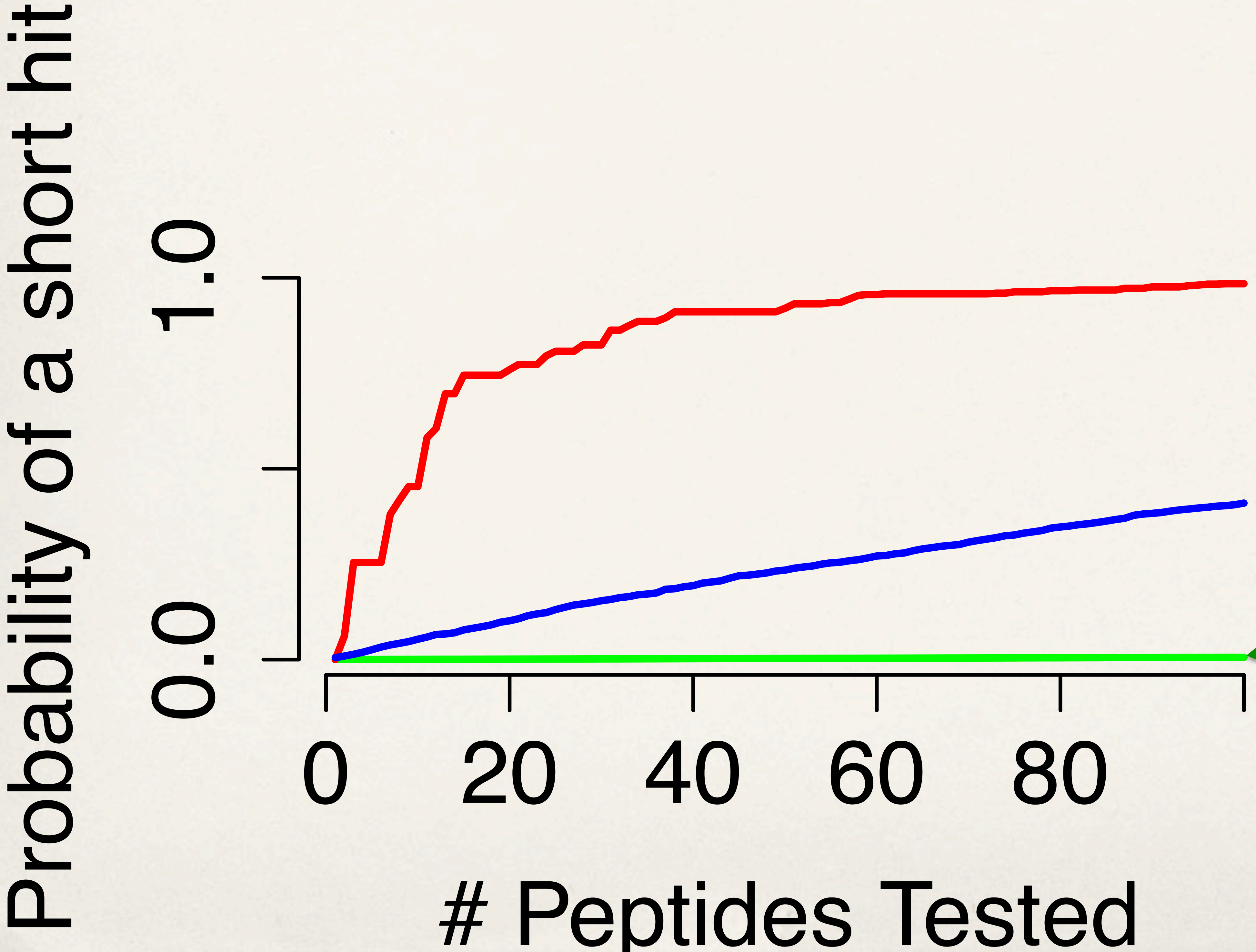
- One simple strategy is:
  - Select those peptides with length < 12.
  - Rank them by predicted probability of a hit
  - Test the top 300.
- The tested peptides are very similar. If the first tested peptide is not a hit, the other ones probably aren't either.

Ranking by probability of a hit does not work well



Plot obtained via a simulation experiment

Using value of information (VOI) works better



POOL, using value of info.

Mutating known hits

Ranking by prob. of a hit

Plot obtained via a simulation experiment

Value of Info. chooses the experiment that maximizes the probability we reach our goal

- Our goal is to find short hits.
- More specifically, our goal is:
  - Find at least one hit shorter than a target length b.
- We should run an experiment that maximizes the probability of reaching this goal.

The best experiment is the solution to a combinatorial optimization problem

- This can be formulated as this combinatorial optimization problem:

$$\max_{S\subseteq E:|S|\leq k} \mathrm{P}(\text{at least one short hit in } S)$$

- Notation:
  - E is the set of all peptides.
  - S is the set of peptides to test.
  - k is the number of peptides we can test in one experiment. Typically, k is between 200 and 500.
  - A “short hit” is a hit whose length is less than b.

We can't solve this exactly, so we approximate the solution using a greedy algorithm

- This combinatorial optimization problem is very challenging : The size of the set $\{S \subseteq E : |S| \leq k\}$ is |E| choose k. If b=15 and k=500, this is 10^{19} choose 500.

- Instead, we build up the set S of peptides to test in stages.

- In each stage, we find the single peptide to add that maximizes the probability of reaching our goal:

$$\max_{e \in E \setminus S} \mathrm{P}(\text{at least one short hit in } S \cup \{e\})$$

- We then add e to S and repeat, until S has k=500 peptides.

The greedy algorithm has an approximation guarantee

Lemma: $P^*(S)$ is a monotone submodular functions of S.

Proposition: Let $\mathrm{OPT} = \max_{S \subseteq E : |S| \leq k} P^*(S)$, and let GREEDY be the value of the solution obtained by the greedy algorithm. Then

$$\frac{\mathrm{OPT} - \mathrm{GREEDY}}{\mathrm{OPT}} \leq 1 - 1/e$$

- In the above, P*(S) = P(at least one short hit in S).

- The proposition follows from the lemma & a result from Nemhauser, Wolsey, Fisher '78.

- This result is similar in spirit to results obtained in Y. Chen & A. Krause, "Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization," ICML 2013.

We can implement the greedy algorithm efficiently

- The greedy optimization step can be shown to be equivalent to

$$\arg\max_{e \in E \setminus S} P(y(e) = 1 | y(x) = 0 \ \forall x \in S)$$

- We can compute this probability by treating all peptides in S as misses, and re-training our model. If we then use a MAP estimate, this probability decomposes over the amino acids, and can be optimized efficiently.

Here is the intuition why this approach works better than "rank by prob. hit"

- Finding the the single peptide to add that maximizes the probability of reaching our goal:

$$\max_{e \in E \setminus S} \mathrm{P}(\text{at least one short hit in } S \cup \{e\})$$

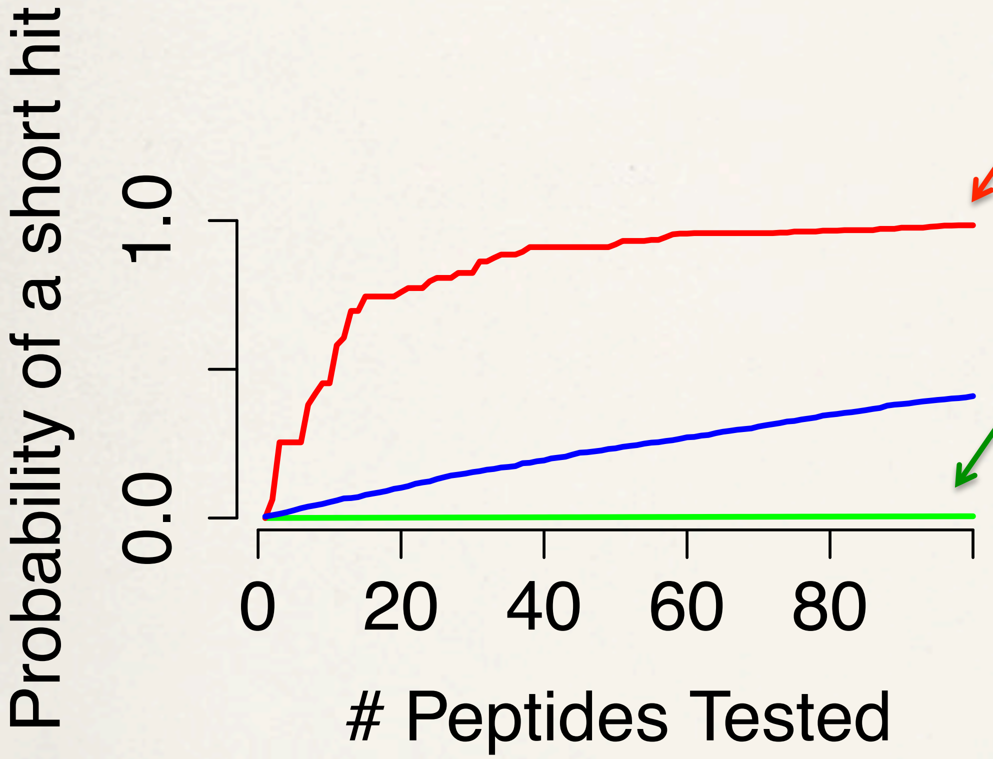
- Is equivalent to:

$$\max_{e \in E \setminus S} \mathrm{P}(e \text{ is a short hit} | \text{no short hits in } S)$$

- Compare this to the "rank by prob. hit" approach

$$\max_{e \in E \setminus S} \mathrm{P}(e \text{ is a short hit})$$

VOI works better because its peptides are more diverse



POOL, using value of info.

$$\max_{e\in E\setminus S} \mathrm{P}(e \text{ is a short hit} | \text{no short hits in } S)$$

Ranking by prob. of a hit

$$\max_{e\in E\setminus S} \mathrm{P}(e \text{ is a short hit})$$

- Peptides added using the value of information approach tend to be **different** from those already in S.
- Its recommendations are more **diverse**.

Using VOI to optimize P(≥1 short hit) has a shortcoming

- Under our Naïve Bayes model, it is usually possible to increase P(hit) by increasing the peptide's length.

- Thus, the experiments that maximize P(≥1 short hit) tend to have length b-1.

- However, a hit strictly shorter than b-1 would be even better.

- To allow us to find such strictly shorter peptides, we might consider an alternate goal: expected improvement.

Optimizing expected improvement would fix this

- Let f(x) be the length of peptide x.
- $f^*(S) = \min_{x \in S : y(x)=1} f(x)$ is the length of the shortest hit found.
- Define the **expected improvement** for testing S as:

$$\mathrm{EI}(S) = E[(b - f^*(S))^+]$$

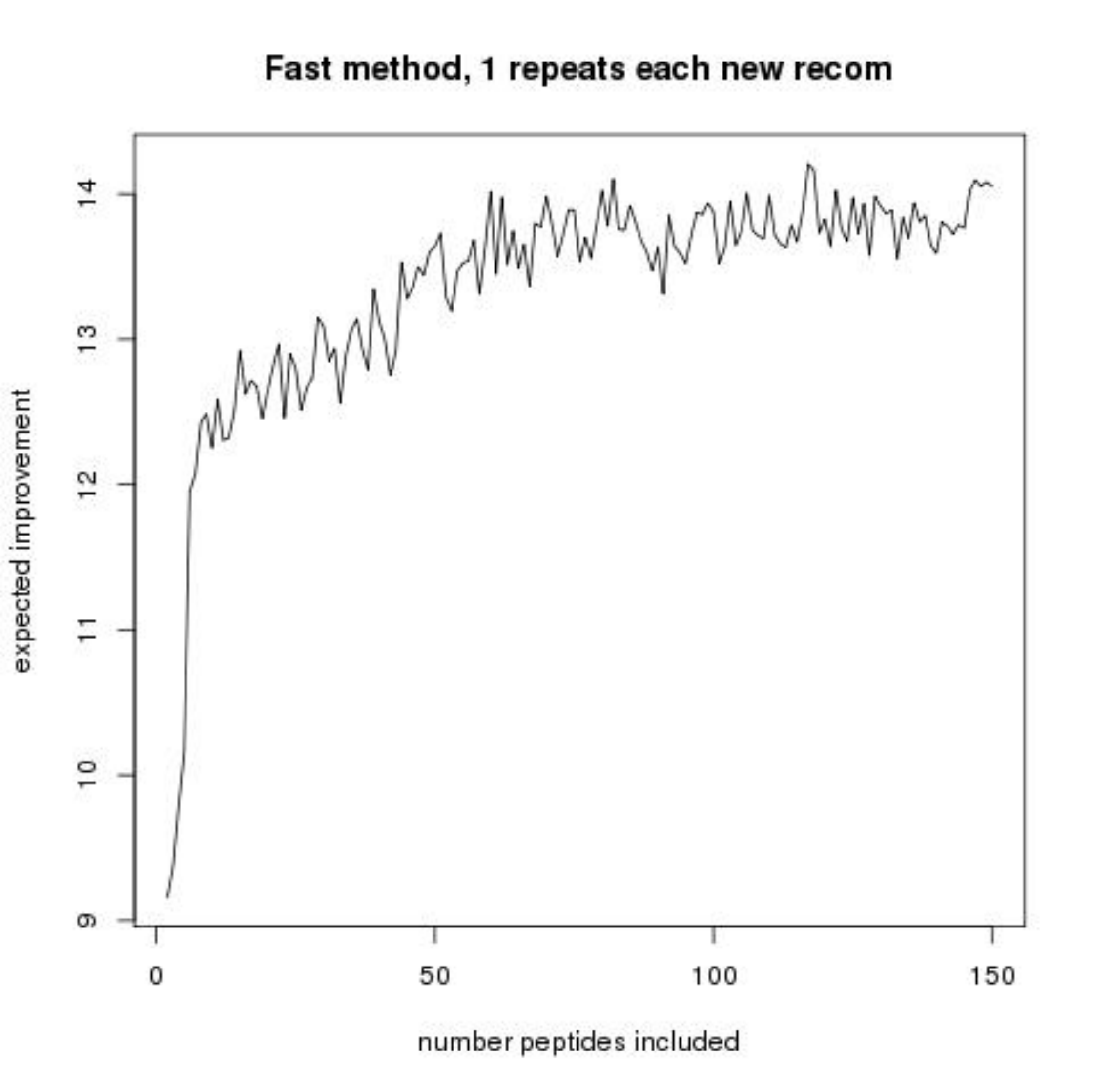
- An S that maximizes EI(S) could contain peptides shorter than b-1.

Efficiently optimizing expected improvement is ongoing work

- Solving $\max_{S \subseteq E : |S| \leq k} \mathrm{EI}(S)$ exactly is very challenging.
- EI(S) is also a monotone submodular function, and so the greedy algorithm also has an approximation guarantee.
- However, actually finding the single peptide to add that maximizes the expected improvement is itself extremely difficult.
- We are currently using an integer program to do this, but results are pending.

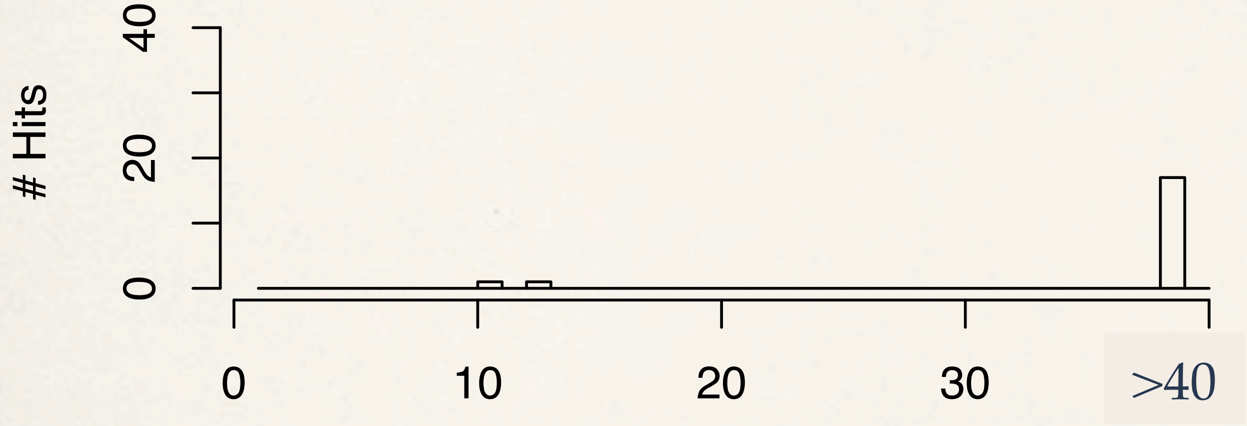
We are greedily optimizing P(≥1 short hit) with one tweak to make real recommendations

- We have used the following approach in recommending experiments to our collaborators.
- We pre-select a random sequence of lengths $a^1,\dots,a^k$ strictly less than b, and require that the nth peptide selected has length less than a^n.
- We then apply the greedy probability of improvement algorithm.
- This improves expected improvement, without hurting P(≥1 short hit).



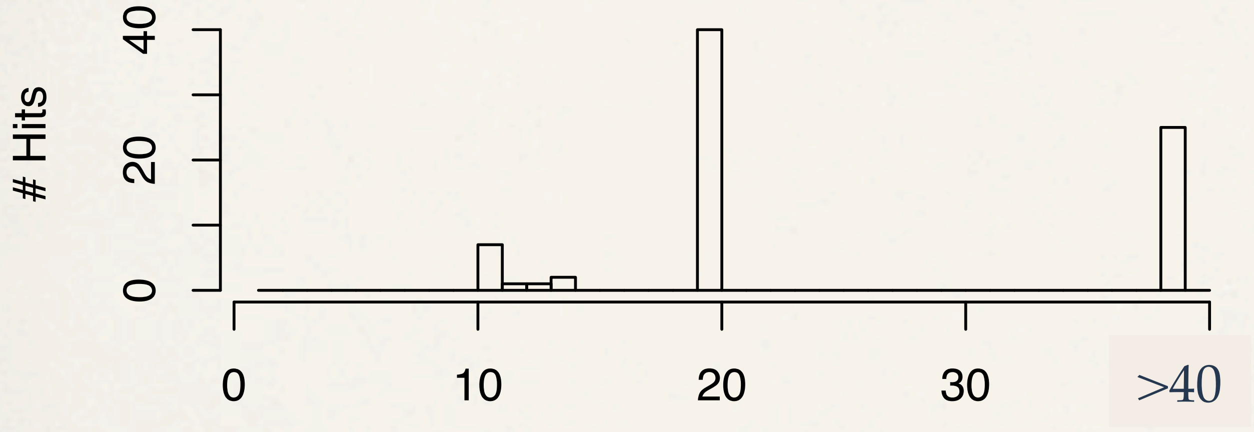
Expected improvement as a function of |S|, estimated via Monte Carlo.

We have found novel short peptides using this method



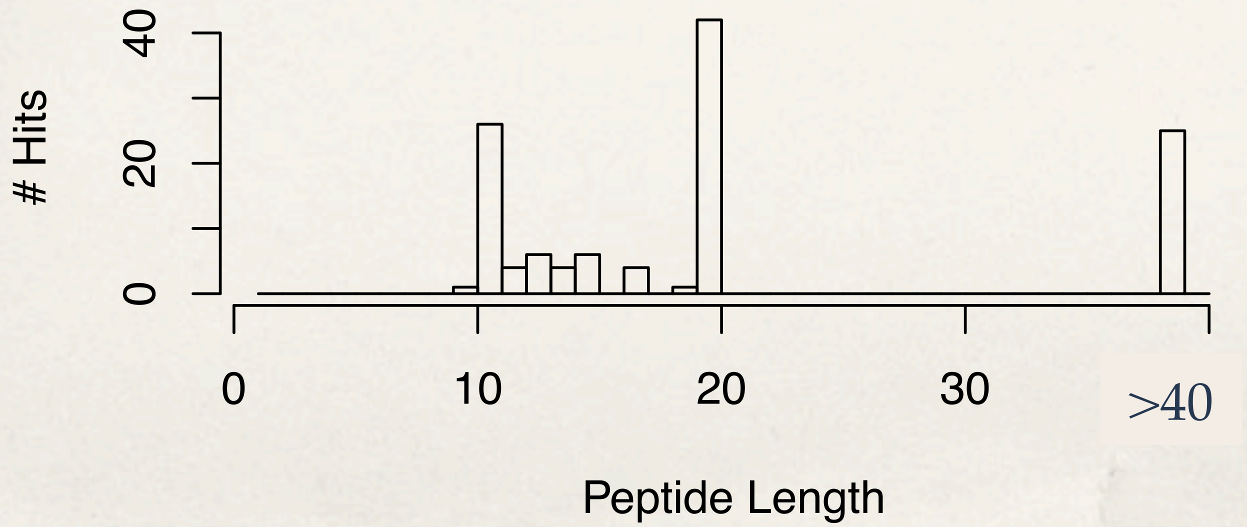
Training Set

Length of shortest hit: 11



After 1 round of POOL

Length of shortest hit: 11



After 2 rounds of POOL

Length of shortest hit: 10

Conclusion

- We have developed an optimal learning method for finding minimal peptide substrates.

- This method has found hits shorter than the shortest previously known.

- Optimal learning methods:

  1. Reduce the experimental effort required to reach a goal.

  2. Increase the chance of achieving a goal within a given experimental budget.

Thank you!

- Any questions?