Bayesian Active Learning for Finding Maximally-valued Exemplars

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

Sunday October 6, 2013

INFORMS Annual Meeting, Minneapolis Minnesota

We use optimal learning to address a problem in biochemistry

- Goal: find a short peptide that, when present, allows a certain pair of chemical reactions to occur.

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

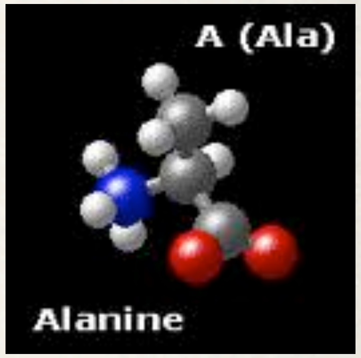
- Our collaborators (all at UCSD): Mike Burkart, Nathan Gianneschi, Mike Gilson, Nick Kosa, Mike Rothmann, Lori Tallorin.

Biology primer: What is a peptide?

- A peptide is a sequence of amino acids. Most of our peptides will be between 5 and 35 amino acids long.
- An amino acid is a molecule. There are 20 of them in nature, and we represent them by capital letters.

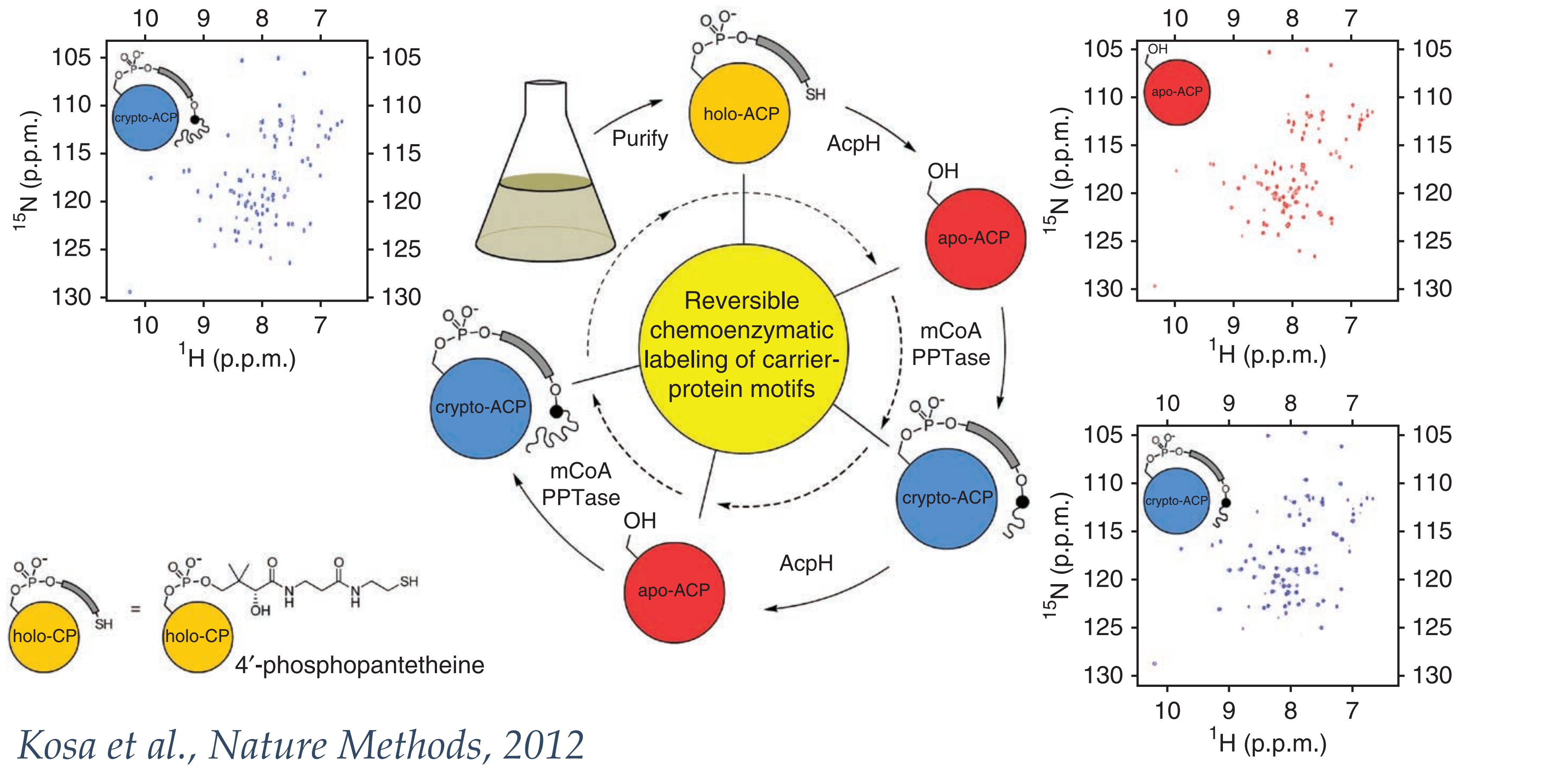
A peptide of length 9:
DSLEFSKIA

Amino acids:
A C D F G H I K
L M N P Q R S
T V W Y



Finding this peptide will support lots of cool biochemistry applications

- Finding a short peptide with this property will allow our collaborators to add & subtract functionality from a protein by embedding this peptide inside it.
- This could be used to create novel sensors, therapeutics, and to study protein interactions.



Kosa et al., Nature Methods, 2012

It is hard to find short hits; Using math will make 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.

Our Methodological Contribution

- We provide two methodological contributions:
  - 1. We build a statistical model that predicts, given a peptide and training data, the probability that this peptide is a “hit”.
  - 2. Based on this statistical model, and a value of information analysis, we recommend a set of peptides to test next that will best support the goal of finding a short hit.
- Our contribution is similar to work in computer science on active learning, which considers the training of statistical classifiers, and other related problems.

Overview

- Introduction
- **Statistical Methodology**
- Value of Information Methodology

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 the positional information about where amino acids occur 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,

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

- 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

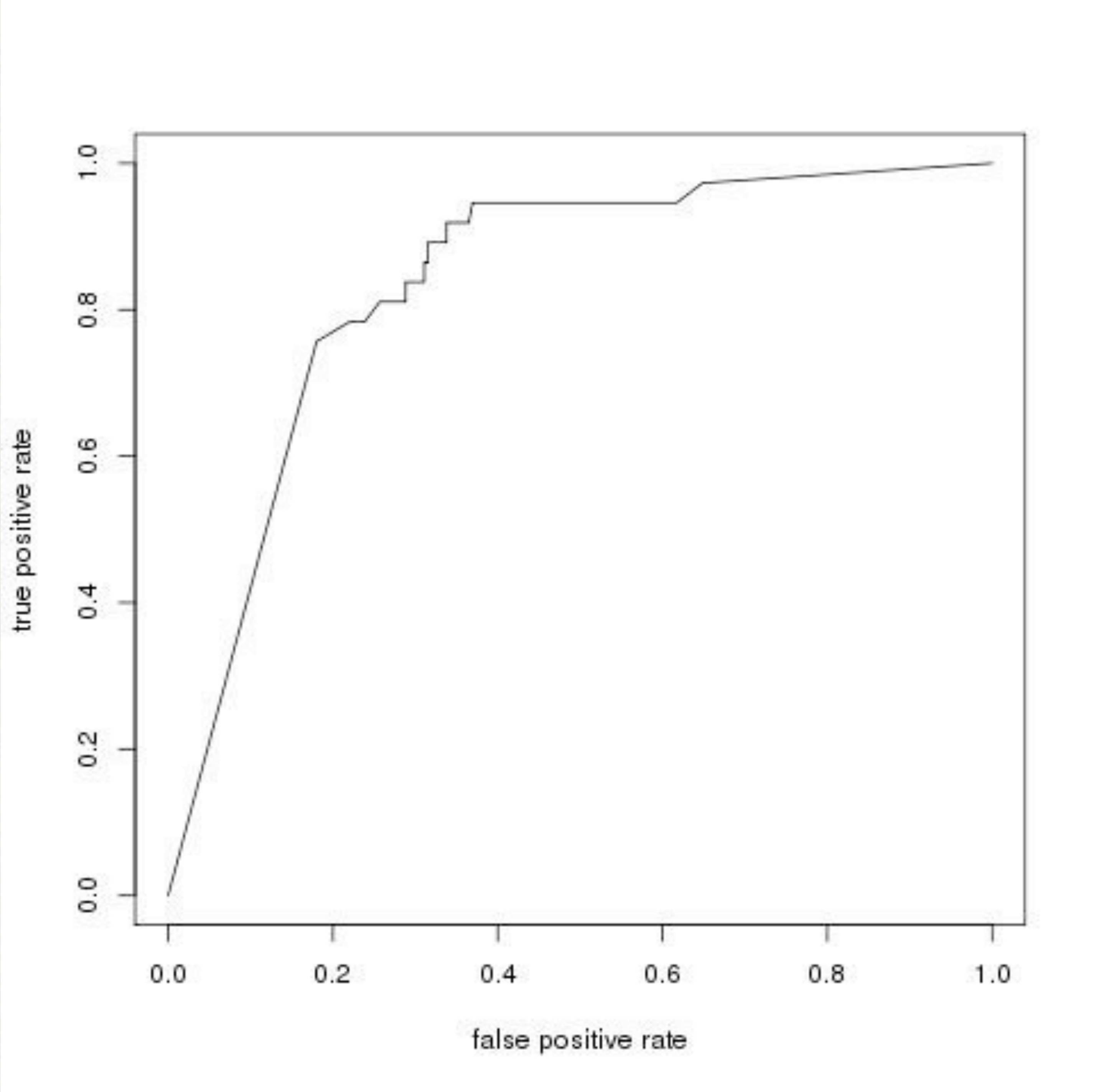
- We put independent Dirichlet prior distributions on each column of the latent matrices $\theta^{(\mathrm{hit})}$ and $\theta^{(\mathrm{miss})}$.

- 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 $x^1,\ldots,x^n$, $y(x^1),\ldots,y(x^n)$, the posterior on the thetas 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 done 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.

Overview

- Introduction
- Statistical Methodology
- **Value of Information Methodology**

Our value of information analysis relies on our statistical model

- The previous slides provide a method that takes training data as input, and produces a probability distribution as output.
  - Input (training data):
    - peptides (exemplars) $x^1,\ldots,x^n$
    - binary labels $y(x^1),\ldots,y(x^n)$ ($y(x)=1$ means "hit", 0 means "miss")
  - Output: a probability distribution over $\{y(x) : x \text{ in } S\}$, where S is any set of untested exemplars.

We seek a hit with small f(x)

- Let f(x) measure the quality of peptide x, with smaller f(x) being better. We take f(x) to be the length. We seek an x for which y(x)=1 (a hit), with f(x) is as small as possible.

- For any proposed set S of peptides to test, let

$$f^*(S) = \min_{x \in S : y(x)=1} f(x)$$

- If we test S, then f*(S) is the quality of the best peptide found. (Let the minimum over the empty set be infinity.)

- Our current discussion could also be extended to other settings, e.g., to drug discovery, where x would be a small molecule, and f(x) would be the “drugability”of that molecule (toxicity, size, solubility).

We value a set of peptides to test according to its ability to provide a short hit

- Let b be a given target value:
    - b could be the length of the shortest known hit;
    - Or, b could be some threshold on length we must meet for the hit to be useful.
- We consider two measures of the value of information provided by testing a set of peptides:
    - Probability of Improvement: $\mathrm{P}^*(S) = P(f^*(S) < b)$
    - Expected Improvement: $\mathrm{EI}(S) = E[(b - f^*(S))^+]$
- Given one of these two measures of the value of information, we then wish to find, and then test, the set S that maximizes this value.

The best set to test can be found by solving a combinatorial optimization problem

- Take g(S) equal to either EI(S) or P*(S).
- Our goal is then to solve: $\max_{S \subseteq E : |S| \leq k} g(S)$
- Here, k is the number of peptides we can test in a batch (about 500), and E is the set of all peptides with length less than b.

- This is a challenging combinatorial optimization problem: 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.

Consider the greedy algorithm

- Let the “greedy algorithm” be the following:
    - Set S to be the empty set.
    - While $|S| \leq k$
        - Let $e^* = \arg\max_{e \in E \setminus S} g(S \cup \{e\})$
        - Let $S \cup \{e^*\}$

The greedy algorithm has an approximation guarantee

Lemma: Both $P^*(S)$ and $\mathrm{EI}(S)$ are monotone submodular functions of S.

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

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

- The proof of the proposition follows directly 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.

Implementing the greedy algorithm is challenging

- To implement the greedy algorithm, we must solve

$$e^* = \arg \max_{e \in E \setminus S} g(S \cup \{e\})$$

- We estimate $g(S \cup \{e\})$ by sampling from our posterior distribution.

- But |E|=20^b-2, making this a very difficult simulation optimization problem.

We can implement the greedy algorithm efficiently for P*

- If we use the probability of improvement criterion (g=P*), then 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.

Using these methods gives a more diverse recommendation than simply ranking by P(hit).

- This approximation to optimizing P*(S) adds peptides to S according to

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

- Compare this to the naive method that simply ranks peptides shorter than b according to probability of being a hit, and takes the top k. This can be computed by adding to S incrementally the peptide:

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

- Using the more sophisticated method provides a more diverse set of peptides to test, better guarding against the possibility of all peptides failing to be a hit.

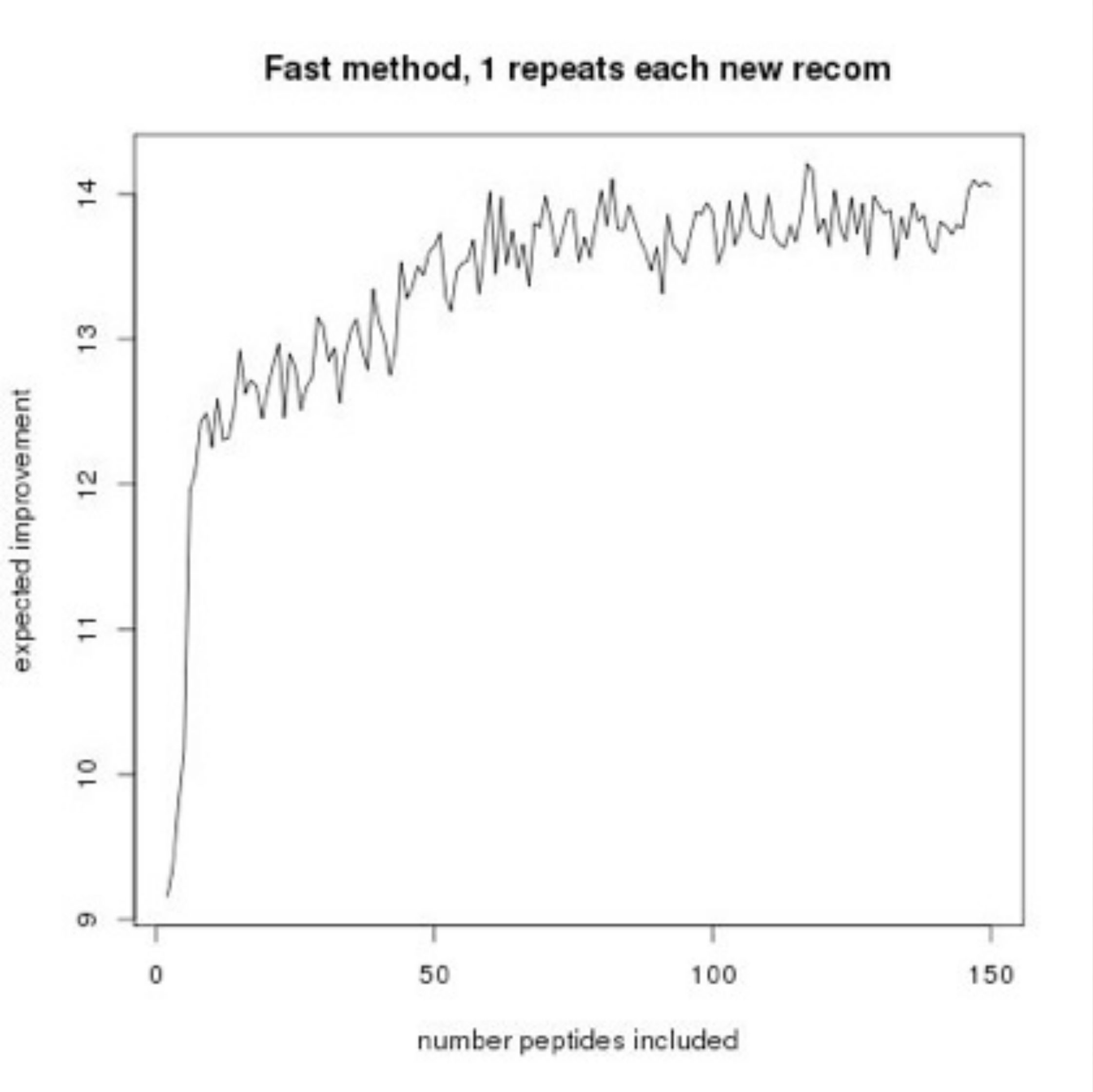
We modify greedy optimization of P* to improve EI

- Being shorter usually reduces the probability of being a hit.

- Thus, a downside of optimizing probability of improvement is that most peptides will have length close to b-1.

- We want to have a broader variety of lengths, to do better on expected improvement.

- To address this, we pre-select a random sequence of lengths $a^1,\ldots,a^k$ strictly less than b, and require that the n^{th} peptide selected has length less than a^n.



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

Experimental results are pending

- We have used this method to suggest a set of 550 peptides to test, using the method described, and a few other variants.

- Our collaborators at UCSD are currently setting up the experiment, and we expect results in a few weeks.

- We hope the experiments reveal some really short hits!

Conclusion

- We used OR techniques to help biochemists reduce the amount of experimental effort required to solve a problem, and to increase their probability of success.

- We used the statistical technique of Naive Bayes, and the value of information techniques of expected improvement and probability of improvement, and applied submodularity to get an approximation guarantee.

- This kind of method can be applied to other problem settings [e.g., drug discovery] where we have expensive-to-obtain binary labels, easy-to-obtain quality measures, and we seek an exemplar with a positive label and good quality.

Thanks!

- Any questions?