AFRL-AFOSR-VA-TR-2016-0018

Optimal Learning for Efficient Experimentation in Nanotechnology, Biochemistry

Warren Powell
TRUSTEES OF PRINCETON UNIVERSITY

12/22/2015
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTB2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 06-01-2016	**2. REPORT TYPE** Final Performance	**3. DATES COVERED** *(From - To)* 01-07-2012 to 30-09-2015

4. TITLE AND SUBTITLE
Optimal Learning for Efficient Experimentation in Nanotechnology and Biochemistry

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-12-1-0200

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Warren Powell

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
TRUSTEES OF PRINCETON UNIVERSITY
1 NASSAU HALL
PRINCETON, NJ 08544-0001 US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTB2

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We have extended our work in optimal learning for continuous parameters, which previously was limited to searching among a small set of possible sample values, to a more general resampling approach that allows us to quickly find the best parameter values globally. We also developed a version of Peptide Optimization with Optimal Learning (POOL) that can be used with quantitative responses, rather than the binary responses assumed by the first version of the method. This greatly expands the set of applications to which POOL can be applied. We also developed a statistical method for characterizing the bias in phage display, and making predictions of activity based on phage display data. This method can be used on its own, and can also provide input to POOL. Finally, we have developed a series of tutorial materials, first in the form of a series of powerpoint presentations, and second as a book chapter, both geared toward materials scientists.

15. SUBJECT TERMS
Biochemistry

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF**	**19a. NAME OF RESPONSIBLE PERSON** Warren Powell
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Unclassified	Unclassified	Unclassified	UU	**PAGES**	**19b. TELEPHONE NUMBER** *(Include area code)* 609-258-5373

DISTRIBUTION A: Distribution approved for public release.

Detailed report:

Optimal Learning for Efficient Experimentation in Nanotechnology and Biochemistry

Principal Investigators:

Warren B. Powell
Princeton University
Department of Operations Research and Financial Engineering
powell@princeton.edu

Peter Frazier
Cornell University
Department of Operations Research and Information Engineering
pf98@orie.edu

Prepared for:

Fariba Fahroo
Hugo De Long

Air Force Office of Scientific Research
Grant: FA9550-12-1-0200

December, 2015

Executive summary

Goal of the research

The goal of this research has been to guide scientists in the running of experiments using the tools of mathematics, spanning combinatorics, statistical machine learning and probabilistic modeling. Our work started with a foundation laid by the Principal Investigators and others under the broad umbrella of "optimal learning" which provided a principled way to guide the scientific process. At its core, this method consists of combining initial belief models with a model of what we learn from the experimental process to design policies to guide the sequencing of experiments.

Technical accomplishments

The research started with a useful set of tools, but we quickly found that the problems faced by scientists were more complex than the relatively simple models we had been working with initially. One of our most powerful tools involves finding the expected value of information from different experiments that can be used to guide scientists (this might be displayed as a heat map). However, we found that computing the value of information for the more complex belief models that we encountered working with scientists required new methodologies.

Our research has created advances along several lines:

- Computing the value of information for nonlinear belief models (tuning temperatures, pressures, concentrations), high dimensional sparse additive models (designing accessibility probes for RNA molecules), multiattribute logistic regression (to maximizes successes), and peptide sequence optimization.
- Calculating the risk of a series of experiments.
- Statistical prediction of peptide and RNA sequence activity.
- Sequencing experiments in the search for peptides with target properties, to maximize the probability of success within a given experimental budget.

Outreach/transitions

Our work has proceeded primarily through interactions with scientists around the country, all funded within Hugh De Long's program. These have included included: Nathan Gianneschi and Mike Burkart at UCSD to build systems of peptides that can be orthogonally labeled and unlabeled by protein-modifying enzymes; Chad Mirkin, Stacey Barnaby, and Jessica Rouge at Northwestern to build Bayesian statistical models that predict the stability of small interfering RNA; Paulette Clancy at Cornell University to build optimization methods that can find local minima of energy surfaces, and predict crystal structures: Paras Prasad (Buffalo), Tiff Walsh (Deakin), and Marc Knecht (Miami) to find peptides that bind specifically to inorganic materials: Lydia Contreras to use sparse-additive belief models to guide the design of probes; Benji Marusama at AFRL in the design of an optimal learning system to guide a robotic scientist.

1. Introduction

Our interactions with the different teams of materials scientists have given us a genuine appreciation of the complexity of the problems being addressed by this community. These interactions have allowed us to identify different problem classes in terms of their mathematical structure (different problem settings can be mathematically equivalent). However, working with the scientists exposed us to the different steps of the scientific process, introducing us to the human dimension of the problem of making experimental decisions.

In this section we highlight the anticipated benefits from the research, which spans new technologies (models and algorithms for solving specific problems), followed by a list of different dimensions of the experimental process that have emerged from the work. We end with a list of different challenges that we encountered.

Sections 2 and 3 provide research narratives for the work being done at Princeton and Cornell, respectively. We have roughly divided the activities between the two universities along broad methodological lines. Princeton began by focusing primarily on optimizing problems with continuous parameters, while Cornell was initially motivated by discrete problems such as learning the behaviors of peptides. Both teams have evolved their own research agendas from this initial starting point, coordinating when areas for potential overlap would arise. We have found that the general problem area is quite broad – there is more than enough to keep not only our two schools quite busy, but also potential research programs that our students might get started if they stay in the area.

1.1. Anticipated benefits from the research

- Improve the scientific methodology by formalizing the process of designing experiments.
- Our Bayesian model requires that scientists articulate their beliefs from their domain knowledge
- We provide guidance to the choice of experiments, often by finding the value of information from an experiment that can be used by the scientist to make tradeoffs in the choice of the next experiment(s) to run.
- We provide a scientific framework for testing and comparing policies for designing and conducting experiments
- Our methods can be used to assess the risk of a series of experiments.

1.2. Dimensions of the optimal learning research

As we worked with the scientists, we identified different stages of the experimental process. Understanding these steps helped us understand where we can add the most value. The steps we have identified include:

- Choosing a line of investigation and assessing risk. Generally we were not involved in this initial stage.

- Identifying the experimental choices (decisions), which may consist of
    - Discrete choices such as materials, compounds, mixtures, choice of peptide sequence.
    - Setting of continuous parameters such as temperatures, pressures, concentrations, …
    - Choice of experimental steps
- Creating belief models which represents in a formal way the prior knowledge of the scientists
- Running experiments and quantifying the results of an experiment
- Updating belief models and re-assessing choices.
- Making the decision to proceed or stop a line of investigation.
- Education – While the focus of our research was using mathematical tools to help guide scientists, we also accepted that part of our role was an educational one. A self-guided tutorial was designed (see http://optimallearning.princeton.edu/tutorialsciences.htm) with the goal of helping scientists learn more about the process of making effective decisions.

1.3. Some challenges:

Below is a summary of some of the challenges we have encountered in the process of pursuing this line of research.

- Our success depends on changing how a scientist makes a decision. This is more than a technical problem - it requires understanding how the scientific process takes place. Possibly our biggest challenge was catching scientists at a point where we could add the most value.
- Scientists are looking for useful results, not necessarily with new methodology. However, we tended to find that each new problem introduced new mathematical twists which would keep our students quite busy. This has made the research both interesting and challenging from the perspective of our methodological community, but it could introduce delays in our ability to meet the needs of scientists.
- Software – Scientists are interested in numbers, not theorems. Implementing and testing algorithms is an essential part of our methodological research, but designing production code that can be used by scientists is not. At this stage of the research we are dependent on using graduate students to both develop and test the algorithms, and then use the software to help the scientists.
- The restrictions on most non-U.S. nationalities at military research facilities (in particular AFRL) has complicated the process of staffing the project with students who could continue the work started by Kris Reyes. Fortunately this has not been an issue with the academic teams.
- The problem of distance – understanding how people think is best done in-person. We do our best using the internet, but working in-person with the scientists is valuable.

- Did we add value? We can run simulations comparing our policies to competing policies, but it is not possible to run a true competition between our policy and what an informed scientist would do on his/her own.
- Publications – We found that we had to explore new avenues for publishing the research where the materials science application was a central component. The methodological journals tend to have a comfort level with certain classes of applications, which generally did not include hard sciences. By contrast, the hard science journals are not friendly to mathematics. We found that journals such as SIAM J. on Scientific Computing and SIAM J. on Uncertainty Quantification were willing to handle papers with a mixture of hard science with mathematical contributions. Informs J. on Computing also seems willing to handle these papers (although we are still waiting on the reviews from our most recent paper). By contrast, the machine learning journals were less comfortable with hard-science applications unless we minimized the context.
- Placement – We are just now facing our first wave of students graduating and seeking jobs. However, the post-doc, Kris Reyes, left the team after finding that he was not attracting offers from materials science departments. It is quite possible that he could have found a position in an industrial engineering department if we had received more notice about his job search. We are finding that our students need to emphasize their methodological research covering a range of applications, rather than just the work in materials science.

2. Research narrative – Princeton

In this section we review the research activities conducted at Princeton.

2.1. Problem settings

We have been involved in the following projects:

1. Creating nanoemulsions for the McAlpine group – This problem involved tuning parameters such as the diameter of bubbles containing the material to be delivered
2. Designing a controller for the ARES robotic scientist at AFRL
3. Maximizing the reflectivity of a surface covered by nanoparticles for the Mirkin group
4. RNA accessibility I – Working with the Contreras group, we designed a method to find the value of information from hundreds (or thousands) of different probes that could be used by a scientist to sequence different experiments to learn the accessibility of an RNA molecule.
5. RNA accessibility II – In September 2015, we were challenged to help design a set of RNA probes that could be used in a single batch experiment to learn about the accessibility for the full set of 62 RNA sequences.

We next divide these projects into five problem settings based on the mathematical properties of the learning problems.

2.1.1. Nonlinear belief models I – Sampled belief models

We begin with the following two projects:

- Creating nanoemulsions for the McAlpine group
- Designing a controller for the ARES robotic scientist at AFRL

Both of these problems involved tuning a series of continuous parameters (diameters, concentrations of gold nanoparticles, temperatures, ratios) to fit the parameters of a belief model.

We have been working with the value of information from an experiment *x*, where *x* was a set of values of the settings of each parameter. For computational reasons, we discretized the continuous space of all settings of these parameters into a set $x_1, x_2, ..., x_K$ (there could be hundreds, even thousands, of these potential settings). To determine which experiment we should conduct next, we began by computing the knowledge gradient, which gives the value of information from an experiment. This is given by

$$v_x^{KG,n} = \mathbb{E}_\theta \mathbb{E}_{W|\theta} \left\{ \max_y F(y, K^{n+1}(x)) \right\} - \max_y F(y, K^n)$$

where

$x =$ Settings of a potential experiment we might run

$K^n =$ The current set of beliefs after n experiments have been run

θ=A (typically multidimensional) random variable representing the true value of the unknown parameters (with current estimate θ^n).

$W =$ A scalar random variable capturing the results of an experiment

$F(y, K^n) =$ The current design value (e.g. conductivity) given what we know now.

$K^{n+1}(x) =$ The updated knowledge after running experiment *x*.

Here, we use K^n to represent "what we know" after *n* experiment. This might be just the point estimate θ^n of a set of parameters, but it can also include estimates of the uncertainty (this could be the variance of θ^n if it is a scalar, or the covariance matrix if it is a vector). $K^{n+1}(x)$ is the uncertain updated state of knowledge if we run experiment *x*; this is a random variable because we have not yet run the experiment, and are uncertain about the outcome.

In other words, the knowledge gradient captures how well we are going to solve our design problem as a resulting of running experiment *x*. The problem is that we have not yet run the experiment, so its outcome is random. As a result, we have to take the expectation of the maximum of our metric $F(y, K^{n+1}(x))$.

Prior to starting this project, we had worked out how to compute $E\left\{\max_y F(y,K^{n+1}(x))\right\}$ when our belief K^n is linear in any unknown parameters which we denote by θ. There are two important classes of linear models:

- Lookup tables – Let θ_x be our belief about running experiment *x* (*x* might be the choice of a particular catalyst). When we use a lookup table representation, we have to store a value θ_x for each value of *x*. This might be required if *x* refers to a discrete choice such as a catalyst or type of molecular substituent. If *x* is a multidimensional vector, then discretizing *x* might produce millions of possibilities, which can become quite clumsy.
- Linear, parametric models – Assume that *x* is a continuous parameter such as a temperature or concentration. We might write our belief as

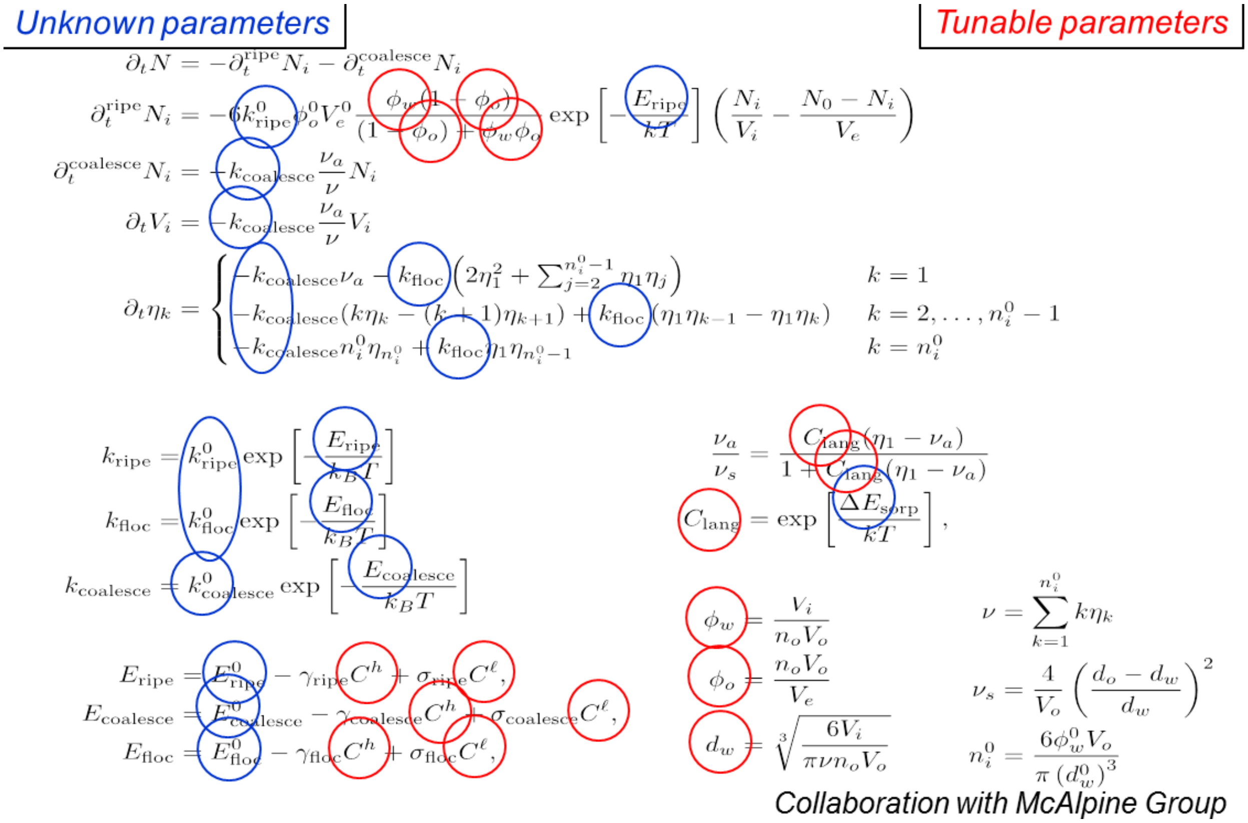
$$K(x) = \theta_0 + \theta_1 x + \theta_2 \ln x \;.$$

The parametric model might be nonlinear in *x*, but it is linear in θ. When this is the case, we have developed methods for calculating the knowledge gradient for these two broad classes of belief models.

When we began working with materials scientists, almost immediately we found there was a need to learn models that were nonlinear in the parameters. Such models might describe the diffusion of a chemical as a function of temperature or ratio of two concentrations. However, this significantly complicates calculating the knowledge gradient because of the problem of computing the expected value of the maximum of a nonlinear function. We note that the biggest challenge in calculating the knowledge gradient is computing the outer expectation over the multidimensional vector θ. For example, the figure below shows the nonlinear model developed for the nanoemulsion experiment being conducted by the McAlpine group. The blue circles highlight the tunable parameters which would make up the vector θ, while the tunable parameters would make up the vector *x*.

We experimented with several strategies for computing $\mathbb{E}\max_y F(y,K^{n+1}(x))$, which is part of the knowledge gradient. Initially we experimented with classical Monte Carlo sampling, but we found that we needed very large samples (because of the nonlinearities in the function $F(y,K^{n+1}(x))$) which made its computation very expensive (we might have thousands of values of *x*). We looked into using the structure of a very general class of statistical models known as *generalized linear models*, but were unable to make any progress there.

We then transitioned to a powerful strategy involving the use of a *sampled belief model*. Here, we redefine the expectation around a small sample of possible values of the parameter vector $\theta \in \hat{\Omega}$ where the set $\hat{\Omega} = \left\{\theta_1,...,\theta_K\right\}$. We would then let $q_k = \text{Prob}[\theta = \theta_k]$ be the probability that θ_k was the true value of θ_k. One value of this approach was that we could control the distribution of possible values of θ better than



Unknown parameters

Tunable parameters

$$\partial_t N = -\partial_t^{\text{ripe}} N_i - \partial_t^{\text{coalesce}} N_i$$

$$\partial_t^{\text{ripe}} N_i = -6k_{\text{ripe}}^0 \phi_o^0 V_e^0 \frac{\phi_w(1-\phi_o)}{(1-\phi_o)+\phi_w\phi_o} \exp\left[-\frac{E_{\text{ripe}}}{kT}\right]\left(\frac{N_i}{V_i} - \frac{N_0 - N_i}{V_e}\right)$$

$$\partial_t^{\text{coalesce}} N_i = -k_{\text{coalesce}} \frac{\nu_a}{\nu} N_i$$

$$\partial_t V_i = -k_{\text{coalesce}} \frac{\nu_a}{\nu} V_i$$

$$\partial_t \eta_k = \begin{cases} -k_{\text{coalesce}}\nu_a - k_{\text{floc}}\left(2\eta_1^2 + \sum_{j=2}^{n_i^0-1} \eta_1\eta_j\right) & k = 1 \\ -k_{\text{coalesce}}(k\eta_k - (k+1)\eta_{k+1}) + k_{\text{floc}}(\eta_1\eta_{k-1} - \eta_1\eta_k) & k = 2, \ldots, n_i^0 - 1 \\ -k_{\text{coalesce}} n_i^0 \eta_{n_i^0} + k_{\text{floc}} \eta_1 \eta_{n_i^0 - 1} & k = n_i^0 \end{cases}$$

$$k_{\text{ripe}} = k_{\text{ripe}}^0 \exp\left[-\frac{E_{\text{ripe}}}{k_B T}\right]$$

$$k_{\text{floc}} = k_{\text{floc}}^0 \exp\left[-\frac{E_{\text{floc}}}{k_B T}\right]$$

$$k_{\text{coalesce}} = k_{\text{coalesce}}^0 \exp\left[-\frac{E_{\text{coalesce}}}{k_B T}\right]$$

$$E_{\text{ripe}} = E_{\text{ripe}}^0 - \gamma_{\text{ripe}} C^h + \sigma_{\text{ripe}} C^\ell,$$

$$E_{\text{coalesce}} = E_{\text{coalesce}}^0 - \gamma_{\text{coalesce}} C^h + \sigma_{\text{coalesce}} C^\ell,$$

$$E_{\text{floc}} = E_{\text{floc}}^0 - \gamma_{\text{floc}} C^h + \sigma_{\text{floc}} C^\ell,$$

$$\frac{\nu_a}{\nu_s} = \frac{C_{\text{lang}}(\eta_1 - \nu_a)}{1 + C_{\text{lang}}(\eta_1 - \nu_a)}$$

$$C_{\text{lang}} = \exp\left[\frac{\Delta E_{\text{sorp}}}{kT}\right],$$

$$\phi_w = \frac{V_i}{n_o V_o} \qquad \nu = \sum_{k=1}^{n_i^0} k\eta_k$$

$$\phi_o = \frac{n_o V_o}{V_e} \qquad \nu_s = \frac{4}{V_o}\left(\frac{d_o - d_w}{d_w}\right)^2$$

$$d_w = \sqrt[3]{\frac{6V_i}{\pi \nu n_o V_o}} \qquad n_i^0 = \frac{6\phi_w^0 V_o}{\pi\,(d_w^0)^3}$$

Collaboration with McAlpine Group

we could than when we assumed that it followed a multivariate distribution (which is how all of our original work proceeded). The problem with using a multivariate normal distribution is that the normal distribution ranged from minus infinity to plus infinity, which invariably created unrealistic, extreme behaviors. In fact, a major advantage of a sampled belief model is that we could allow scientists (perhaps even a team) create a population of possible values.

We would start with an initial set of probabilities that was uniform over the sample. That is, if we have K possible values, we would set

$$q_k^0 = \frac{1}{K}.$$

Next, we would then assume that an experiment produced a noisy outcome from our nonlinear model, which we can write as

$$W_x^{n+1} = F(x = x^n \mid \theta = \theta^k) + \varepsilon^{n+1}.$$

Typically we would assume that the noise ε^{n+1} was normally distributed with mean 0 and a standard deviation that would be estimated by the scientist based on prior experience with experimental variability.

Next, we would use a simple application of Bayes' theorem to produce an updated estimate of the probabilities q^{n+1} given the prior probabilities q^n.

In our initial work, we would assume that one of the set of possible values $\theta_1,\ldots,\theta_K$ was the true parameter vector. We found that the knowledge gradient was able to quickly identify which of these was the truth.

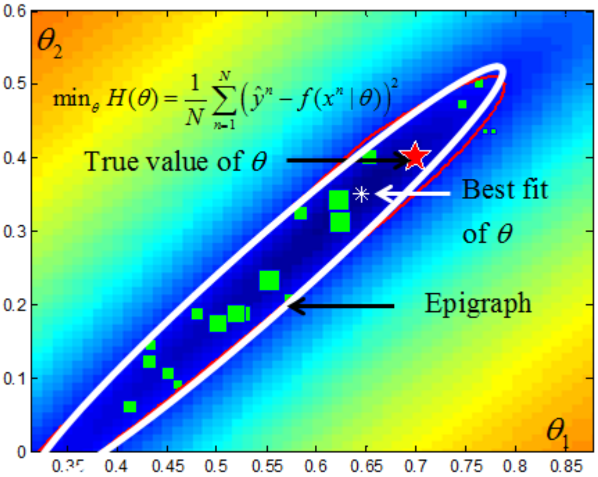
The combination of computational tractability, and the transparency in the specification of the set of possible values, made this quite attractive. We also believe that it did quite a good job guiding the experiments required to maximize $F(y,\theta)$. However, if θ had more than three dimensions, we found that it was generally the case that none of the sampled values of θ was close to the true value in all dimensions (this is the well-known curse of dimensionality). We could not even count on using the estimate

$$\bar{\theta}^n = \sum_{k=1}^{K} q_k^n \theta_k$$

as an estimate of the true value. This concern motivated the line of research we describe next.

2.1.2. Nonlinear belief models II - Resampling

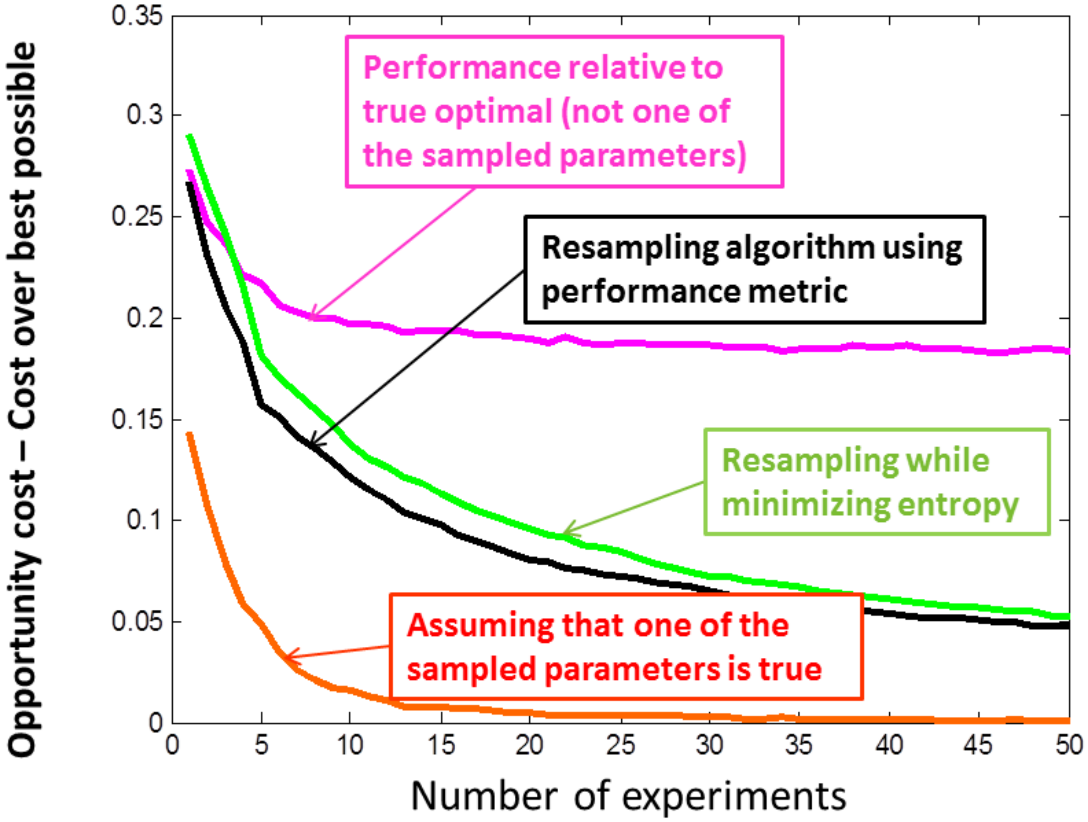
After several false starts, we stumbled into the idea of using *resampling*, where we would generate new values of θ_k that would be added to our sampled set $\hat{\Omega}$. We would do this by periodically using our series of experiments $(x^0,\hat{y}^1),\ldots,(x^{n-1},\hat{y}^n)$ (where x^n is the parameter settings made after *n* observations, and $\hat{y}^{n+1}$ is the outcome of the *n+1*st experiment). We would then use this data to solve the statistical problem:



$$\min_\theta H(\theta) = \frac{1}{N}\sum_{n=0}^{N-1}\left(\hat{y}^{n+1} - F(x^n\,|\,\theta)\right)^2 .$$

We could have taken the best value of θ that solved this equation, but we made better use of the very limited set of experiments by taking a sample of, say, 20 “good” values of θ that solved this problem. This is known as sampling from the *epigraph* of the function (the points in the white ellipse in the figure to the right). We solved this problem by creating a very large set of sampled θ (say, 10,000), finding the values that produced small values of the fitting function $H(\theta)$, and then taking a sample of these. We would use this expanded set of sampled values and then return to our uniform prior (now over a larger set), and rerun the Bayesian updating equations (without requiring any new physical experiments). We would then drop the values of θ_k with the smallest probabilities to obtain a new set of K parameters (we would have to rerun the Bayesian updating equations one more time with this reduced set).

This resampling strategy has been working very well in synthetic experiments where we sample from a problem where we control the truth. In the figure below, we plot the opportunity cost, which measures our ability to find the design that optimizes our metric (reflexivity, conductivity, strength, …) using our simulated known truth against the design we identify using data from our experiments. The lower, red line shows how well we thought we were doing when we assumed that one of our initial set of K parameters included the truth. The top purple line shows how well we were actually doing with a fixed sample when we recognized that this was just a sample, and that the truth was drawn from a much larger population.



The two intermediate lines are drawn from two variations of our resampling algorithm, where we experimented with two performance metrics. The first was the original metric (e.g. maximizing reflexivity, conductivity, strength, …) while the second used the entropy of the belief vector q^n which placed more emphasis on learning the true value of the parameters (we often found that this was a priority of the scientists). We found that using the actual performance metric would either work similarly or slightly better.

The finding that entropy worked reasonably well was itself a somewhat surprising result, since the opportunity cost focused on the original performance metric. Using just entropy would perform just as well as using the performance metric when all of the unknown parameters were important. There are problems, however, where some of the parameters are much more important than others. We think that it is in these problems (which we tested by creating irrelevant parameters and throwing them into the set) where using the performance metric proved to be a somewhat better guide.

This entire line of research provided a much deeper insight into the challenge of optimal learning using a parametric belief model. The work we originally started with Peter during his graduate student years focused initially on lookup table belief models, where there is a parameter θ_x for each possible experiment *x* (here, θ_x is the performance metric). With this belief model, it is very important to focus on the experiments that help us identify the settings that produce the best values of θ_x. By contrast, when using a low-dimensional parametric model, it important to learn the correct value of θ than it is to maximize $F(x|\theta)$, since learning the correct value of θ allows us to then maximize $F(x|\theta)$. This was an insight we did not fully appreciate until this year.

2.1.3. Maximizing the reflexivity of a surface

This project evolved out of discussions with the Mirkin group (during a two-day visit last year). The experimental problem involved two stages:

Stage 1: The scientists had to choose the size and shape for the nanoparticles that that would be spread over the surface.

Stage 2: They then had to run a series of different experiments that could be run in batch.

This setting introduced two novel twists. First, the experiments were nested: the decision on size and shape had to be made before performing a series of experiments at different densities. Second, the nested experiments (over different densities) could be run in batch, an experimental technique that actually arises with some frequency.

To solve this, we first had to determine the expected value of information we could expect from a single batch of experiments. This information then had to be used to inform the value of making tests of experiments over different sizes and shapes (obviously, these were categorical choices).

Batch experimentation means that we have to choose experiments without knowing the outcomes of other experiments. This problem would normally be solved using a classical design-of-experiments strategy such as Latin hypersquares, where a set of M experiments are chosen so as to maximize the spread of experiments over the search space. The limitation with these methods is that they do not allow the scientist to use his/her domain knowledge. For example, Mirkin's group had an approximate knowledge of the range of densities that were most promising. Our logic exploits a Bayesian prior so that the scientists can provide a reasonable guess.

A method that would not work in this setting is to compute the knowledge gradient for all possible densities that might be tested, and then picking the M best (if M is the size of our batch). Such an approach would tend to pick a set of densities that were close in value.

The problem is that picking, say, density 10 reduces the value of trying densities 9 and 11.

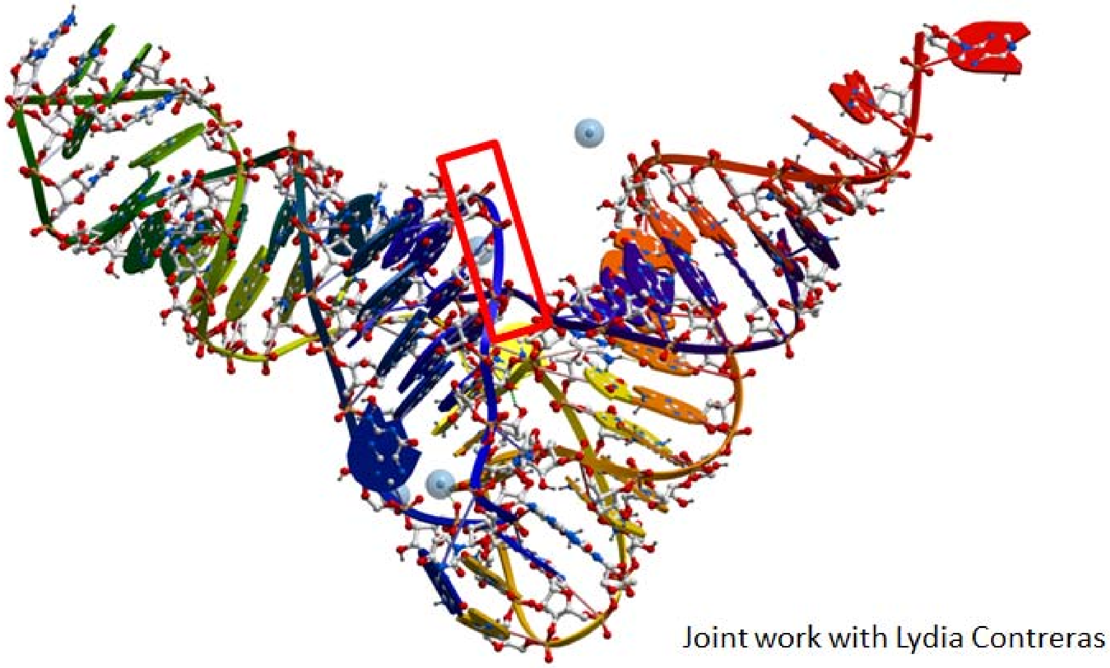
We overcame this by simulating the potential outcomes of each experiment, and then updating the knowledge gradients before picking the next. We repeated this M times to create a batch of M experiments that maximized a simulated value of information.

Finally, this value of information was imbedded in the evaluation of each size and shape of a nanoparticle.

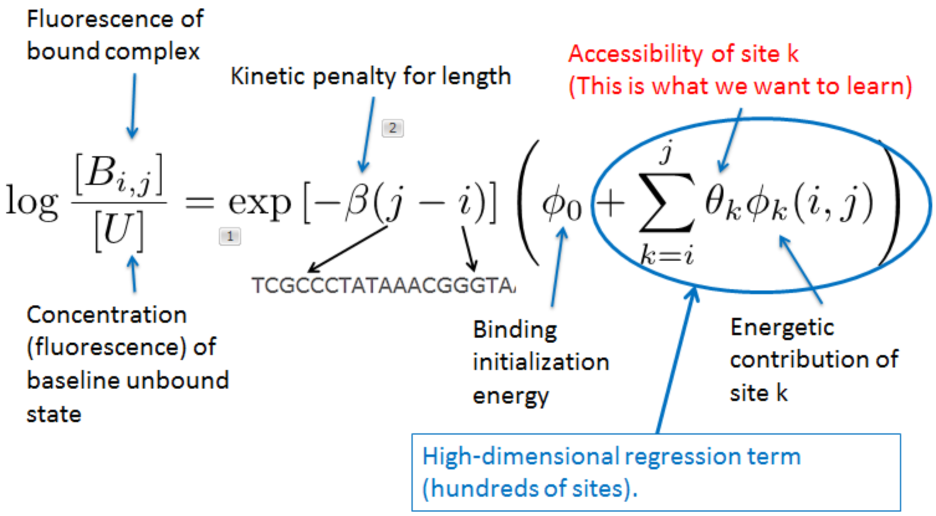
The simulated performance of this method was excellent, with a publication in the respected SIAM J. on Scientific Computing. By the time that this work was complete, the scientists had moved on to new questions. However, the method we worked out to solve this problem was used later in our work on RNA accessibility for Lydia Contreras.

2.1.4. RNA accessibility I

Lydia Contreras approached us with an interesting challenge – designing probes to learn the structure of an RNA molecule. The probe has to be designed to attach to a specific sequence of nucleotides. If the probe attaches, then we know that this particular sequence is accessible. Thus, a probe might be designed to attached to a particular region highlight (in red) below:



Creating and testing probes is time consuming, so there was considerable interest in a policy that would make this experimental process as efficient as possible. We approached the problem by first creating a belief model of the form:



$$\log \frac{[B_{i,j}]}{[U]} = \exp\left[-\beta(j-i)\right]\left(\phi_0 + \sum_{k=i}^{j} \theta_k \phi_k(i,j)\right)$$

Fluorescence of bound complex
Kinetic penalty for length
Accessibility of site k
(This is what we want to learn)
2
1
TCGCCCTATAAACGGGTA
Concentration (fluorescence) of baseline unbound state
Binding initialization energy
Energetic contribution of site k
High-dimensional regression term (hundreds of sites).

The methodological challenge was the fact that the summation in the equation above was over a large number of sites, where we had to estimate the accessibility coefficient θ_k for each site (a single strand of RNA might have from 100 to 400 sites). Further, most of these parameters would be zero.

This was a good setting for a *sparse additive belief model*, which describes high-dimensional linear models where most of the parameters are zero. Traditionally, this is easily handled by a method called Lasso which includes a penalty for allowing a parameter to be greater than zero. Increasing this penalty reduces the number of nonzero parameters, reducing the effect of spurious coefficients (nonzero coefficients with relatively meaningless values).

The difficult is that Lasso has to be run in batch, and therefore assumes that the observations already exist. In our experimental setting, we have to collect observations one at a time. We applied the thinking of the knowledge gradient, but this required solving a problem of the form

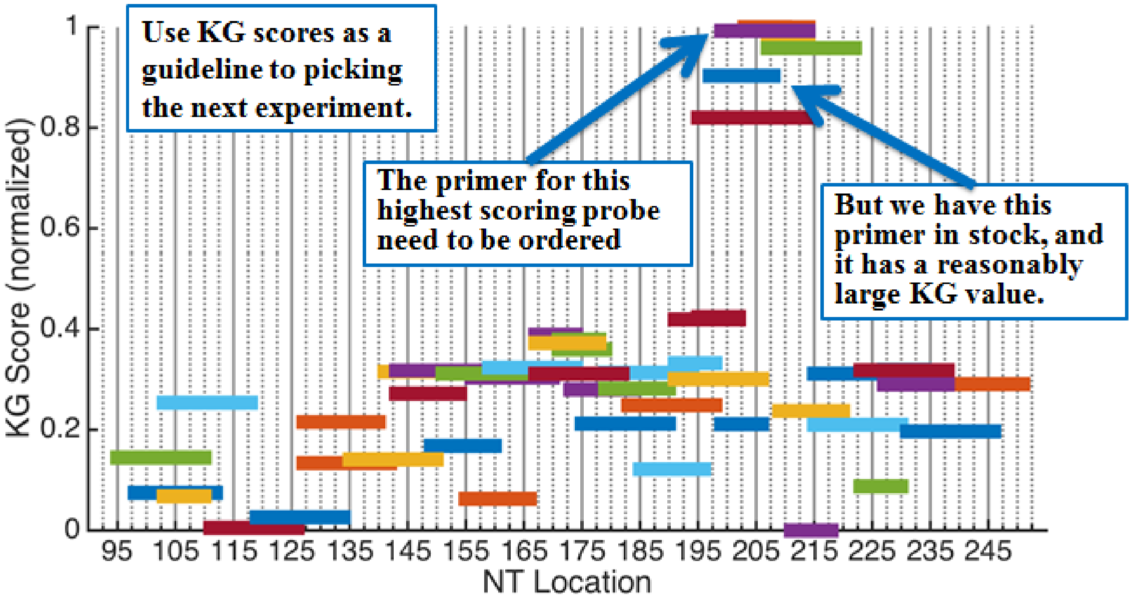
$$\begin{aligned} v_x^{\mathrm{KG},n} &= \mathbb{E}_{\boldsymbol{\alpha},\epsilon,\boldsymbol{\zeta}^{n+1},\boldsymbol{p}^{n+1}}\left(\max_{x'\in\mathcal{X}} \theta_{x'}^{n+1} | S^n, x^n = x\right) - \max_{x'\in\mathcal{X}} \theta_{x'}^n \\ &\approx \mathbb{E}_{\boldsymbol{p}^n}\mathbb{E}_{\boldsymbol{\zeta}^n|\boldsymbol{p}^n}\mathbb{E}_{\boldsymbol{\alpha},\epsilon|\boldsymbol{\zeta}^n,\boldsymbol{p}^n}\left(\max_{x'\in\mathcal{X}} \theta_{x'}^{n+1} | S^n, x^n = x, \boldsymbol{\zeta}^n, \boldsymbol{p}^n\right) - \max_{x'\in\mathcal{X}} \theta_{x'}^n \end{aligned}$$

Normally, the expectation is over two sets of variables: the prior on the parameters, followed by the random outcome of an experiment. Here, we have four sets of expectations: 1) the random variable indicating which sites have zero coefficients (represented by the 0/1 random variables ζ^n), 2) the probabilities p^{n+1} (which are themselves random at time n) of whether each $\zeta_i^{n+1} = 0/1$, 3) the values of the nonzero coefficients α^{n+1} , , and 4) the outcome of the experiments, given by ε .

This expectation was computationally intractable, so we approximated it with the second line, where we replaced the random variables p^{n+1} and ζ^{n+1}, with their current estimates p^n and ζ^n, allowing us to focus on the randomness of the update coefficients α^{n+1} and the experimental outcome ε^{n+1}. Even this approximation was quite difficult, since the updated estimates α^{n+1} required anticipating the solution of the Lasso optimization logic. In fact, we used a version of Lasso known as group-Lasso to handle the property that the coefficients could be clustered due to their relative proximity to each other.

The graduate student, Yan Li, undertook a very difficult implementation of a variant of Lasso known as recursive Lasso to handle the fact that we were not doing experiments in batch, but rather were updating estimates one observation at a time.

The system was implemented by computing the value of information for each possible probe. This was then displayed using the graphic below, where the horizontal axis showed the location on the RNA strand, and the vertical axis showed the value of information. This graphic allows a scientist to make subjective evaluations of which experiment to run next, since some probes are easier to construct than others (for example, because material might be already available in the lab).



The logic has been carefully tested using simulated data that allows us to assume a truth, and then evaluate how well we discover the truth.

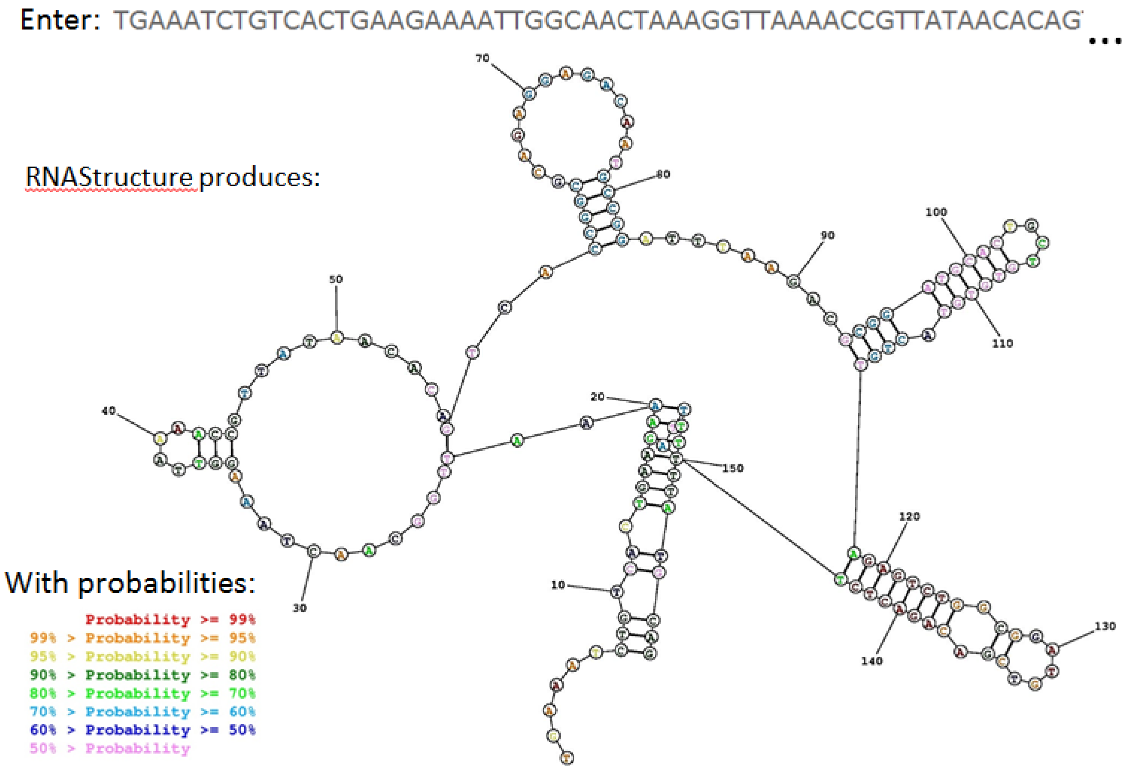
At this point in the project, our post-doc, Kris Reyes, took another position, leaving the work in the hands of the graduate student, Yan Li. However, by the time we were ready to move forward, Lydia and her team had switched gears, which we describe next.

2.1.5. RNA accessibility II

Round II of the RNA accessibility project involved making the transition to guiding sequential experiments, to one where all the work was going to be done as one large batch. From the perspective of information acquisition, this is an entirely different problem, since the only information we are given all comes in the form of the initial prior.

Lydia's graduate student, Jorge, introduced us to a numerical modeler called RNAStructure which takes as input a sequence of nucleotides (for a particular RNA), and outputs a two dimensional depiction as shown below. This two-dimensional graphic hints at the structure of the molecule. Also, each nucleotide was printed in a color that indicated the probability that the segment would be accessible.

Unfortunately, these results were not output in a machine readable form. Since we had to convert 61 RNA molecules (approximately 8000 nucleotides), we bought pizza for my entire lab and we spent the afternoon with everyone translating these figures into a machine readable form.



2.2. Objective functions

In the process of doing this work, we have encountered the following objective functions:

- Maximizing a performance metric such as conductivity, strength, or minimizing the number of flaws. This was the original objective function that we used to start the research.
- Minimizing the deviation of an estimated parameter from the true parameter. We made the transition to a methodology that struck a balance between maximizing a performance metric and minimizing the deviation from the true parameter (this work is contained in the resampling algorithm).
- Maximizing the probability of a discrete success (as in creating a double-walled carbon nanotube) – This objective is being used in ongoing research using a logistic regression belief model.
- Maximizing the fit of a release profile by minimizing the square of the deviation from a target release profile. This objective is being used in ongoing research to handle a Chi-squared objective (a draft paper should be ready this spring).
- Minimizing the risk that an experiment products a metric less than some target. This is work we presented last year that indicates that our logic for sequencing experiments can be used to simulate the experimental process. This logic can be used to help program directors assess the risk of undertaking a new set of experiments.

Recognizing this diversity of objectives has made us realize that we have to pay special attention to understanding what a scientist is trying to achieve.

2.3. New learning algorithms

Our work has produced a series of new learning algorithms, including:

- Maximizing the value of information for a sampled, discrete prior.
- Maximizing the value of information for sampled priors with resampling.
- Maximizing the value of information from a batch set of experiments (implemented for both the problem of testing continuous densities, as well as the probes used for RNA accessibility).
- Maximizing the value of information for nested experiments.
- Maximizing the value of information using a sparse, additive belief model.

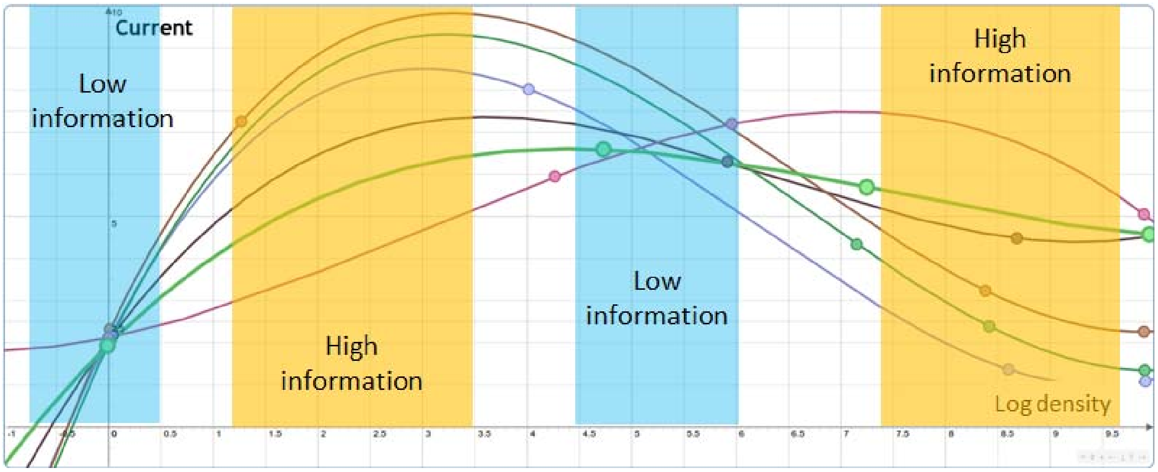
We are also working on two new methods:

- Maximizing the expected number of successes (e.g. the number of double-walled nanotubes produced by the ARES robot) using a logistic regression belief model.

- Maximizing the value of information when the belief model is represented using tree regression. This extends the sparse additive belief model so that it can handle nonlinear interactions between explanatory attributes.

2.4. Next steps

An issue that is on our radar screen is that our nonlinear belief models are typically simplifications of the actual problem. These models are likely going to be locally accurate. However, our mathematics assumes that they are globally accurate. As a result, a byproduct is that we may recommend performing extreme experiments, since this is where we tend to collect the most information. The figure below illustrates this, as it illustrates that there is generally the most variability near the edges of the experimental region (it is possible to create settings where the opposite is true, but the figure below is more typical in actual experiments).



There are two problems with running experiments near the edges:

- The edges tend to represent extreme values (e.g. very high or very low temperatures) which may be difficult experiments to actually run.
- Our low-order model is going to be less accurate near the edges, while the best results may be near the mid-point.

We are currently working out the theory for optimal learning for nonlinear belief models that are only locally accurate. In the process we have made to date, we are working on a method which adaptively tries to learn the optimum of the function (this is known as a "proximal point" in the algorithmic literature). Rather than sampling at this point (as other algorithms do), the knowledge gradient will sample in the neighborhood – not too close (you do not learn anything), but not too far (due to the errors in the model).

2.5. Mathematical results

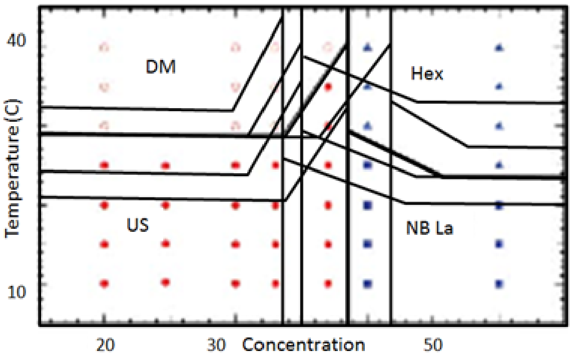
As we develop new methodologies, we also explore what we can from a theoretical perspective. These results tend to come in the following forms:

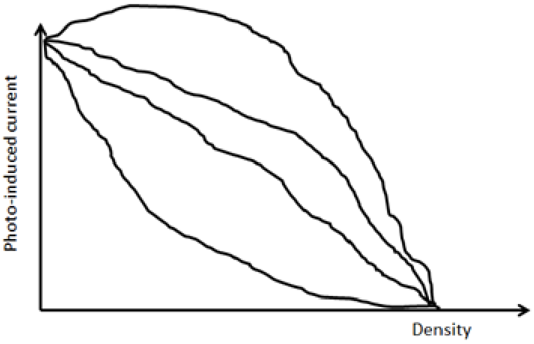
- Asymptotic convergence results – We like to demonstrate what happens if our experimental budget were to grow without bound. We currently have asymptotic results for all of our algorithms (the convergence for the discrete prior with and without resampling is in preparation).
- Bounds on finite convergence – These results tend to be much harder, but we were able to develop these bounds for the sparse-additive belief model, a result that would naturally generalize to the original results with a linear model.
- Other mathematical properties – These are typically structural results that provide insights into specific problems.

2.6. Belief models

Without question, the biggest learning experience was the value of using domain knowledge to develop belief models. For example, we learned quite a bit from Kris Reyes who contributed his ability to model the nonlinear dynamics of chemical processes using simple differential equations characterized by a few physical parameters (an example of this is illustrated above).

However, belief models tended to be unique to each setting. For example, a scientist at MIT was trying to determine which experiment to run to very expensive experiments (each required dedicated time at a LBNL facility). This problem involved testing two parameters – the combination of these two parameters would produce four different materials. The figure below (left) showed the scientist's uncertainty about the boundaries between the regions. The figure on the right represented a series of hand-drawn images showing the relationship between a photo-induced current and the density of nanoparticles on a surface. Simply constructing these diagrams helps to highlight the regions where a scientist should be running experiments.





3. Research Narrative – Cornell

At Cornell, we have developed Peptide Optimization with Optimal Learning (POOL), which is a new suite of mathematical methods for finding peptide sequences with desirable properties with minimal experimentation. We have deployed POOL in the following scientific projects with AFOSR-funded scientific collaborators:

- Finding peptides with specific enzymatic activity. Joint work with Nathan Gianneschi, Michael Burkart, and Michael Gilson, at UCSD.
- Finding peptides with specific binding against metals. Joint work with Paras Prasad (Buffalo), Marc Knecht (Miami), and Tiffany Walsh (Deakin), also involving Mark Swihart (Buffalo) and Aidong Zhang (Buffalo).

We have also worked on the following complementary projects in which the goal is to develop statistical models for inferring chemical activity from peptide sequence and historical training data:

- Development of statistical models for inferring peptide binding against carbon materials from phage display data (with Rajesh Naik and Christina Harsch at AFRL).
- Development of statistical models for inferring the stability of small interfering RNA (with Jessica Rouge, Stacey Barnaby, and Chad Mirkin at Northwestern).

3.1. Overview of POOL

In POOL, we discover peptides with desirable properties through this iterative loop:

1. We begin with data in the form of some (typically small) collection of peptide sequences with previously determined activity ("training data"), and potentially with prior information supplied by scientific collaborators.

2. We use this training data and prior information as input to a Bayesian statistical model that provides a joint probability distribution over the activity of unmeasured peptides. This probability distribution can be used to predict activity for previously unmeasured peptides; can also be used to calculate an uncertainty associated with these predictions; and can even be used to compute a correlation between the errors of two previously unmeasured peptides. In our work to date, the statistical model used has been Naive Bayes or Bayesian linear regression.

3. We use this probability distribution to recommend a peptide or set of peptides to test next. This recommendation is created by valuing experiments according to value of information analysis, and then by using combinatorial optimization techniques to find a peptide or set of peptides that provide near-optimal value of information.

4. Our scientific collaborators test the recommended peptides, add this to the training data, and repeat from step 1 until the experimental budget is exhausted or a peptide of sufficient quality is discovered.

This iterative process is illustrated below in Figure 1.

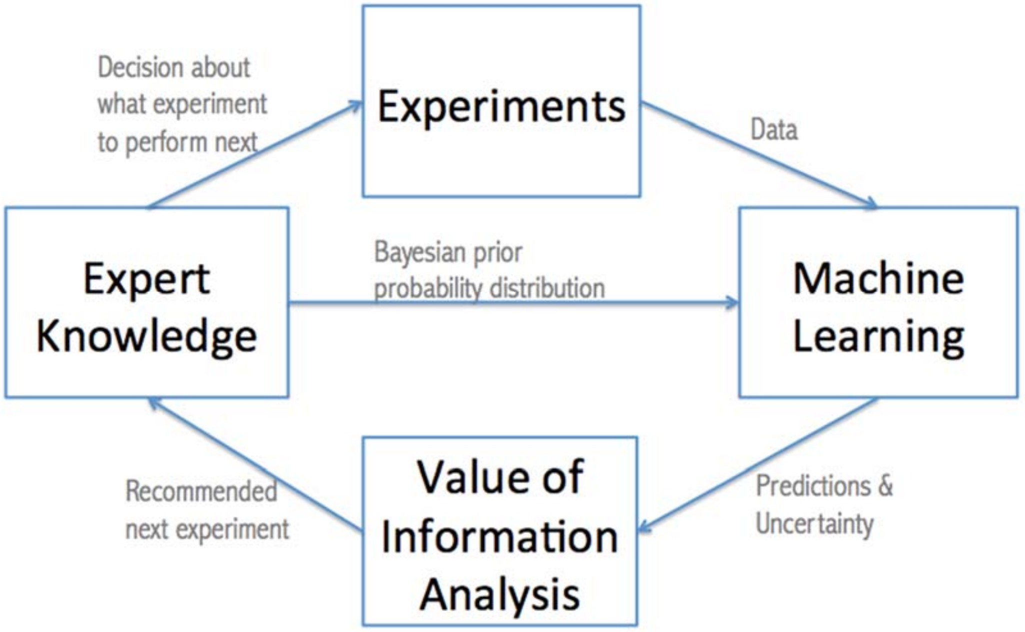

Experiments
Data
Machine Learning
Predictions & Uncertainty
Value of Information Analysis
Recommended next experiment
Expert Knowledge
Decision about what experiment to perform next
Bayesian prior probability distribution

Figure 1: POOL's iterative approach to finding peptides with desirable properties. Experiments are processed using a machine learning-based statistical model. This model is used within a value of information analysis to generate a recommendation of peptides to test. These peptides are evaluated by an expert (this step is optional, but nevertheless useful), and then tested in experiment. This loop is repeated several times until peptides of the desired quality are discovered.

3.2. POOL's capabilities and demonstrated uses

We have developed versions of POOL for several specific peptide discovery tasks, which evolved over the course of the project to address specific needs from our scientific collaborators.

- POOL v1.0 seeks peptides that are as short as possible, and that exhibit activity, where activity is binary ("hit" or "miss") and is measured by an assay that can test many peptides at a time. Activity can be measured by a single assay, or can be a composite of several different assays. POOL v1.0 requires examples in its training data of longer peptides or proteins that are hits.
- POOL v2.0 also seeks short peptides that exhibit activity, using an assay that can test many peptides simultaneously, assuming binary responses, but differs from POOL v1.0 in that it is designed for finding peptides with specific activity, measured by combining results from multiple independent assays. By performing statistical analysis separately on each assay type, POOL v2.0 offers to find short hits with fewer experiments than POOL v1.0 for composite activity measures. Also, while POOL v1.0 requires examples in its training data of long hits, POOL v2.0 requires examples only of peptides (long or short) that are active for each constituent assay, and not for the global specific activity measure of interest. This makes it significantly more general.

- POOL v3.0 can be used with assays that provide quantitative rather than binary responses, and can be used to search for peptides that provide a large response from a single assay, or for which the ratio of responses of one assay over another is large. By using quantitative responses rather than binary ones obtained by thresholding, we provide more information to statistical methods, improving their performance, and also avoiding the need to choose arbitrary thresholds

Below we describe in more detail the use of these versions of POOL in four distinct scientific use cases:

- Reversible peptide labeling: In this project, we used POOL v1.0 to search for peptides that were substrates for a pair of protein-modifying enzymes, Sfp and Acp hydrolase, where activity was measured through the use of a membrane-based assay. (Joint with the Giannechis team at UCSD)
- Orthogonal reversible peptide labeling: In this project, we used POOL v2.0 to search for peptides that are substrates for one of a pair of phosphopantetheinyl transferases (PPTases), Sfp and AcpS, but not the other, and also are substrates for AcpH, where activity was measured through the same membrane-based assay. (Joint with the Giannechis team at UCSD)
- Specific peptide binders: In this project, we use POOL v3.0 to search for peptides that bind strong to gold and weakly to silver, and for other peptides that bing strong to silver and weakly to gold. We measure activity one peptide at a time, using a quantitative QCM assay. (Joint with the Prasad team based at Buffalo)
- Peptides with specific matrix metalloproteinase (MMP) activity: In this project, we use POOL v3.0 to search for peptides that exhibit activity for one of a pair of MMP enzymes, but not the other, using a quantitative assay where we measure multiple peptides at a time. (Joint with the Giannechis team at UCSD)

Figure 2 provides a timeline showing these four scientific demonstrations of POOL, and the respective versions of the POOL methodology used.

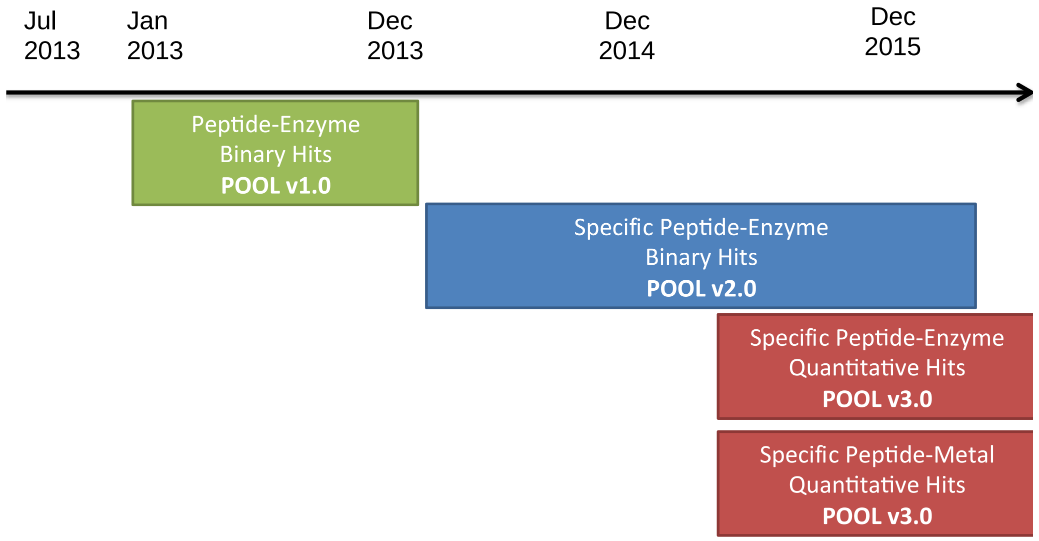


Figure 2: POOL's algorithmic development. Timeline describing the development of the three versions of POOL, and the scientific uses to which they have been put.

3.3. The character of POOL's recommendations

Before describing the mathematical foundations of POOL in detail, we first offer a more intuitive explanation of the character of POOL's recommendations, and how they differ from other past approaches using machine learning for chemical discovery.

While past approaches to the use of machine learning prediction for chemical discovery have focused on the accuracy of the machine learning method, POOL's value of information analysis builds in "mathematical safeguards" to offer robust performance in spite of inaccurate machine learning predictions.

When POOL recommends several peptides to test simultaneously, included will be some peptides from the region of sequence space that is predicted to perform best, and other regions that are likely to perform well if this region does not perform as well as expected. POOL hedges its bets in this way, providing a set of peptides to test that is both predicted to perform well, and that is robust to prediction errors. Building in robustness in this way typically produces diverse recommendations, but unlike *ad hoc* approaches to ensuring sequence diversity, this approach ensures that the diversity added is of the kind most supportive of the overarching peptide discovery goal.

Figure 3 illustrates the diversity of the peptides recommended through POOL in the reversible labeling project. In this visualization, peptides have been projected into a two-dimensional space in a way that preserves the distance between pairs of peptides, calculated using a modified version of edit distance, using a dimension reduction technique. Thus, in this diagram, the distance between two points is approximately proportional to the modified edit distance between the corresponding pair of peptides.

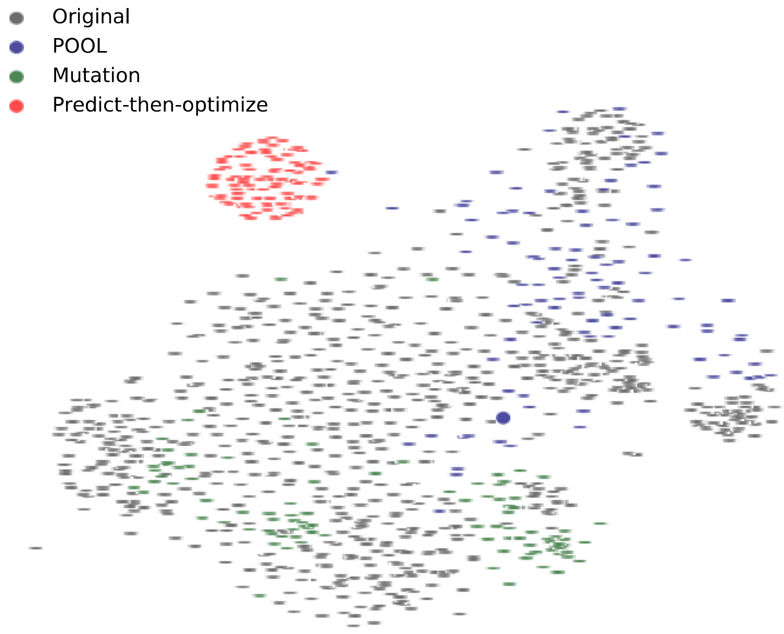


Figure 3: Visualization of Peptide Optimization with Optimal Learning (POOL). Each point represents a peptide, present either as training data, or recommended by POOL or one of two benchmark methods: Mutation, and which takes known hits and mutates them randomly; and Predict-then-optimize, which ranks peptides according to the same machine learning prediction method used by POOL. We see that POOL provides a set of peptides that includes at least one peptide from the region of the search space predicted to perform well, but that also explores regions of the sequence space that will perform well if this prediction is erroneous.

This diagram visualizes recommendations from POOL (purple) calculated using training data (grey) available in one round of the reversible labeling project, in which our goal was to find short peptides that were substrates for one PPTase enzyme but not the other, and also a substrate for AcpH. It also visualizes recommendations made using two other benchmark methods: Mutation, which takes known hits and mutates them; and Predict-the-optimize, which uses the same prediction method used by POOL, but simply ranks the peptides according to their probability of being a hit, and tests them in decreasing order of this probability.

We see that Mutation provides small clumps of recommendations, in the vicinity of known hits, while Predict-then-optimize provides a single clump of recommendations, in a region of the space likely to contain a hit. POOL's first recommendation is near this clump of recommendations from Predict-then-optimize, but its subsequent recommendations explore the space, providing a diverse set of peptides to test that is much more likely to provide at least one hit.

Figure 4 shows results from a simulation study in which we compare these benchmark methods against POOL, in the task of finding a single peptide that exhibits specific activity (i.e., is a "hit"). We use training data and the Naïve Bayes statistical method to compute a probability distribution over whether each untested peptide is a hit or not, and then simulate data using this probability distribution, hiding it from the methods to be evaluated. Then, for each sample of simulated peptides, and for a given number of peptides tested, we calculate whether the method would have found a short hit. By averaging across samples of simulated peptides, we are able to calculate the probability that a method is able to find a short hit, within a given experimental budget. The figure shows that POOL is able to obtain a substantial improvement over both benchmark methods.

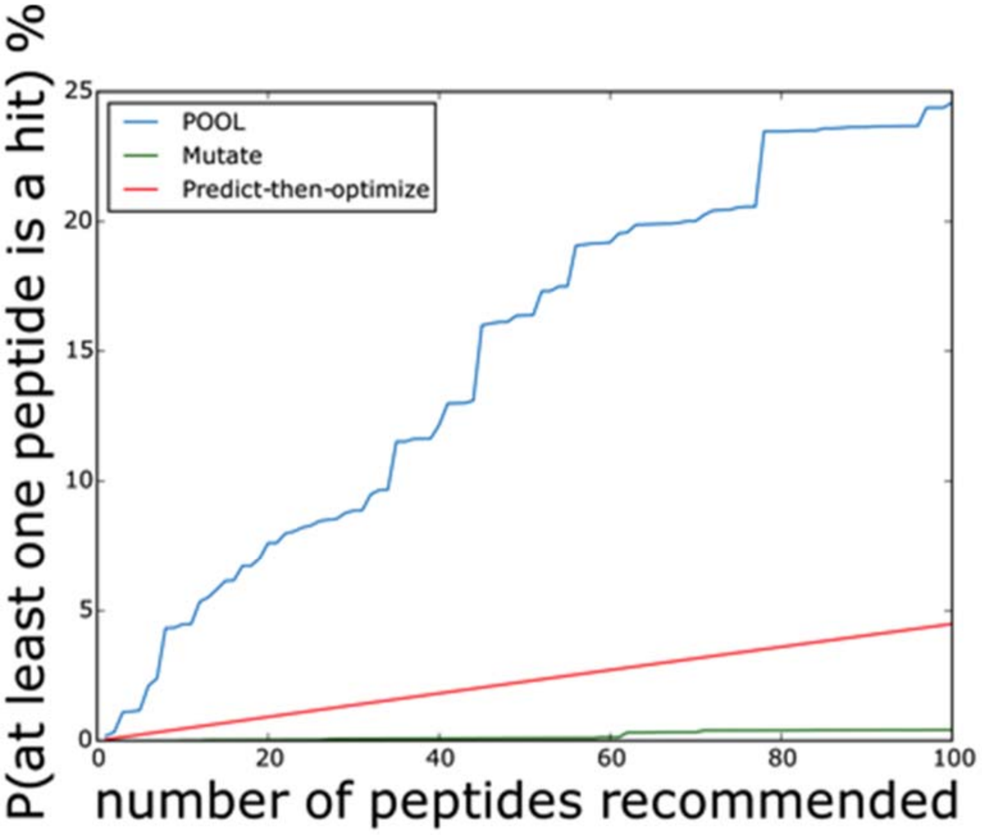


Figure 4: Simulation study comparing the performance of POOL and two benchmark methods, in terms of their ability to find at least one short peptide with specific activity in a reversible labeling project, using the same training data illustrated by Figure 3.

3.4. Mathematical Foundations of POOL

At the heart of POOL lies first a probabilistic machine learning model, which is a variant of Naïve Bayes in POOL v1.0 and v2.0, and is Bayesian linear regression in POOL v3.0, and a value of information analysis. To give the main ideas behind POOL in a mathematically precise way, we give a detailed description of POOL v1.0 below.

3.4.1. Statistical Analysis

In POOL v1.0, we represent peptides using a reduced amino acid alphabet, as a sequence $x = (x_1,\ldots,x_k)$ of elements from this alphabet. We let y(x) represent whether peptide x is a hit (y(x)=1) or not (y(x)=0), and following the Naïve Bayes approach we assume that there are two unknown matrices $\theta^{(hit)}$ and $\theta^{(miss)}$ that provide the probability of a hit according to the following formula.

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

Here, P(hit) is the known prior probability that a peptide chosen uniformly at random from sequence space is a hit, and was chosen in consultation with our scientific collaborators. P(miss) is the corresponding probability that a peptide is not a hit, and is given by, P(miss) = 1 – P(hit).

In the formula above, $\theta^{(hit)}$ and $\theta^{(miss)}$ are unknown, and are estimated using Bayesian inference, in which we place a prior probability distribution created by placing an independent Dirichlet distribution on each column. With this choice of prior distribution, the posterior distribution on $\theta^{(hit)}$ and $\theta^{(miss)}$ retains the same functional form, and can be sampled efficiently. Thus, P(y(x)=1 | x) can be obtained by sampling many $\theta^{(hit)}$ and $\theta^{(miss)}$ matrices from the posterior, computing P(y(x)=1 | x, $\theta^{(hit)}$, $\theta^{(miss)}$) for each, and then averaging this quantity across samples.

Given a collection of peptides, a joint distribution over the binary vector given by whether each peptide is a hit or not can be computed similarly. While property of being a hit, y(x)=1 given $\theta^{(hit)}$ and $\theta^{(miss)}$ is conditionally independent across peptides, they are correlated under the unconditional (marginal) distribution, because the common use of the same sampled $\theta^{(hit)}$ and $\theta^{(miss)}$ matrices induces correlation.

3.4.2. Value of Information Analysis

Using the statistical model described above, we may compute a probability distribution given all available training data over the vector (y(x) : x is in S), for any set of peptides S. This then allows us to compute the quantity P(at least one short hit in S) as

P(at least one short hit in S) = P(y(x) = 1 and length(x) < b for at least one x in S).

We then seek to find the set of peptides to test S that maximizes the probability of success, where success is measured as finding a hit in the set of peptides tested whose length is less than (or equal to) b. This problem can be written mathematically as

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

where E is the set of all peptides, and k is the number of peptides, e.g., 500, that can be tested in a single round of experimentation.

This is a challenging optimization problem, and so we use an approximate solution based around a greedy approach, in which we iteratively add peptides to S that most increase the objective function, P(at least one short hit in S), until we reach our limit on the size of S. Although this approach does not necessarily provide the optimal recommendation, its quality as compared with the optimal solution has a mathematical guarantee on quality, given by the following theorem.

Proposition: Let 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

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

The peptide added under the greedy strategy also has appealing intuition: it is the one that is most likely to be a short hit, given that all peptides previously added to S are not hits. This is the mechanism referenced above by which POOL provides diverse recommendations, and builds in mathematical safeguards against the event that the initial peptides tested are misses.

3.5. POOL's demonstrated uses

We now describe three scientific demonstrations of POOL, which illustrate POOL's functionality, and demonstrate its general ability to support and accelerate scientific discovery.

3.5.1. Reversible peptide labeling systems

In this project, joint with the Gianneschi / Burkart / Gilson team at UCSD, we sought to find peptides that are substrates for a pair of protein-modifying enzymes: Sfp, which is a phosphopantetheinyl transferase (PPTase); and Acp hydrolase (AcpH).

For peptides that are a substrate for both enzymes, pictured below in Figure 5, the first enzyme (Sfp) catalyzes a reaction that attachés a phosphopantytheine arm (PPant-arm) to a conversed serine residue within the peptide. This PPant-arm may have attached to it an arbitrary label, which might be a fluorescent dye, or could be a surface, or a bead, or some other object providing chemical functionality. This attachment functionalizes the peptide, or the larger protein in which the peptide is embedded. The second enzyme then removes the PPant-arm, and the functionality that it provides, returning the peptide to its original form. This is illustrated in Figure 5.

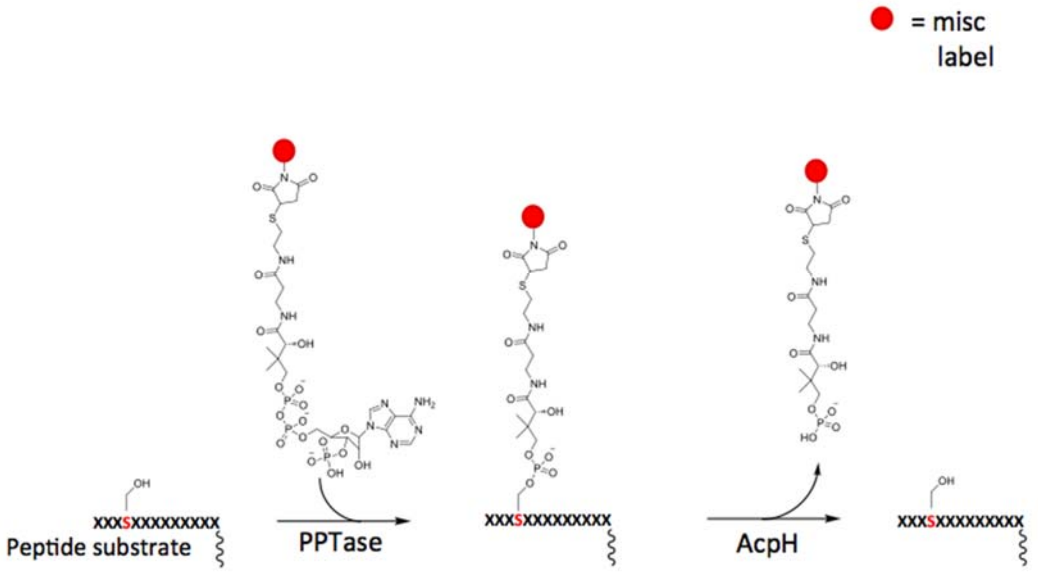
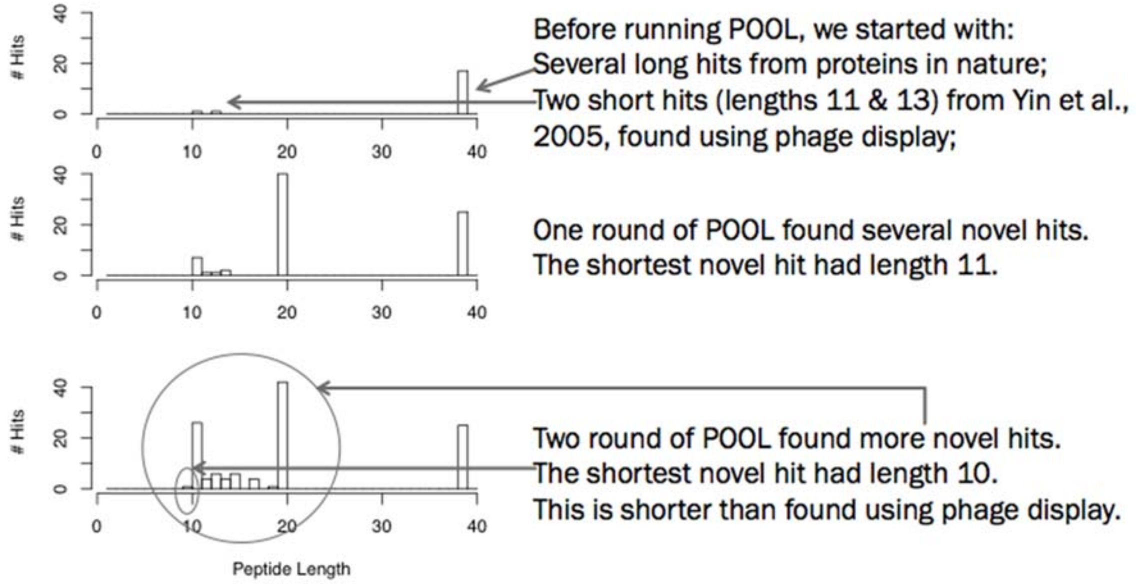


Figure 5: Illustration of the chemical reactions catalyzed by pair of enzymes utilized in the reversible labeling system, and the orthogonal reversible labeling system. In the first reaction, catalyzed by a PPTase (either Sfp or AcpS), a phosphopantytheine arm (PPant-arm) is added to a conversed serine residue within the peptide that is a substrate for this reaction (the red "S" in the figure).

In this first demonstration of POOL's use, we sought to find a peptide that was short enough to not disturb the functionality of proteins in which it would be embedded, but that would be a substrate for both of these chemical reactions. To support this effort, we had a number of longer peptides obtained from organisms in nature that were known to be substrates for both enzymes, and some other peptides that were substrates for Sfp, but not for AcpH. We also had two shorter peptides that were substrates for both, one of length 11 and one of length 13, discovered using phage display.

We applied POOL v1.0 to this task, using it to find hits shorter than were previously known. Figure 6 shows the number of hits found in each round, and their length. After one round, we found a number of short hits of length equal to the shortest found using phage display, or somewhat larger. After two rounds, we found more novel hits, and one whose length was 10 amino acids, shorter than found using phage display.



Before running POOL, we started with:
Several long hits from proteins in nature;
Two short hits (lengths 11 & 13) from Yin et al., 2005, found using phage display;

One round of POOL found several novel hits.
The shortest novel hit had length 11.

Two round of POOL found more novel hits.
The shortest novel hit had length 10.
This is shorter than found using phage display.

Figure 6: The progress of POOL v1.0 in finding peptide substrates for Sfp and AcpS. After two rounds of POOL, we were able to find a peptide hit shorter than found using phage display, and were able to find a number of other novel hits.

3.5.2. Orthogonal reversible peptide labeling systems

Building on the success of POOL v1.0 in finding peptides that were substrates for both Sfp and AcpH, we used POOL v2.0 to find peptides that would support two orthogonal reversible peptides, one using Sfp and AcpH, and the other using a different PPTase, AcpS, together with AcpH. This allows the addition of two different types of functionality to different peptide substrates, and proteins in which they are embedded, providing greater control and flexibility in the design and manipulation of peptide-based systems.

To achieve this, we needed to find peptides that were substrates for AcpS and AcpH, but not Sfp (AcpS-specific labeling with unlabeling), and for Sfp and AcpH but not AcpS (Sfp-specific labeling with unlabeling).

POOL v2.0 was critical to the success of this discovery process, because we did not have examples to start of peptides that provided specific labeling of either type with unlabeling, thus failing to meet the precondition for POOL v1.0. Instead, we only had examples of peptides that exhibited activity with each individual enzyme, which met the conditions for POOL v2.0.

Figure 7 shows the progress of POOL v2.0 in finding specific hits. After 4 rounds, a number of specific hits of each type were found, including several short peptides that exhibited AcpS specific labeling with unlabeling, despite the fact that no peptides with this activity profile were known at the start of the experiment, regardless of length.

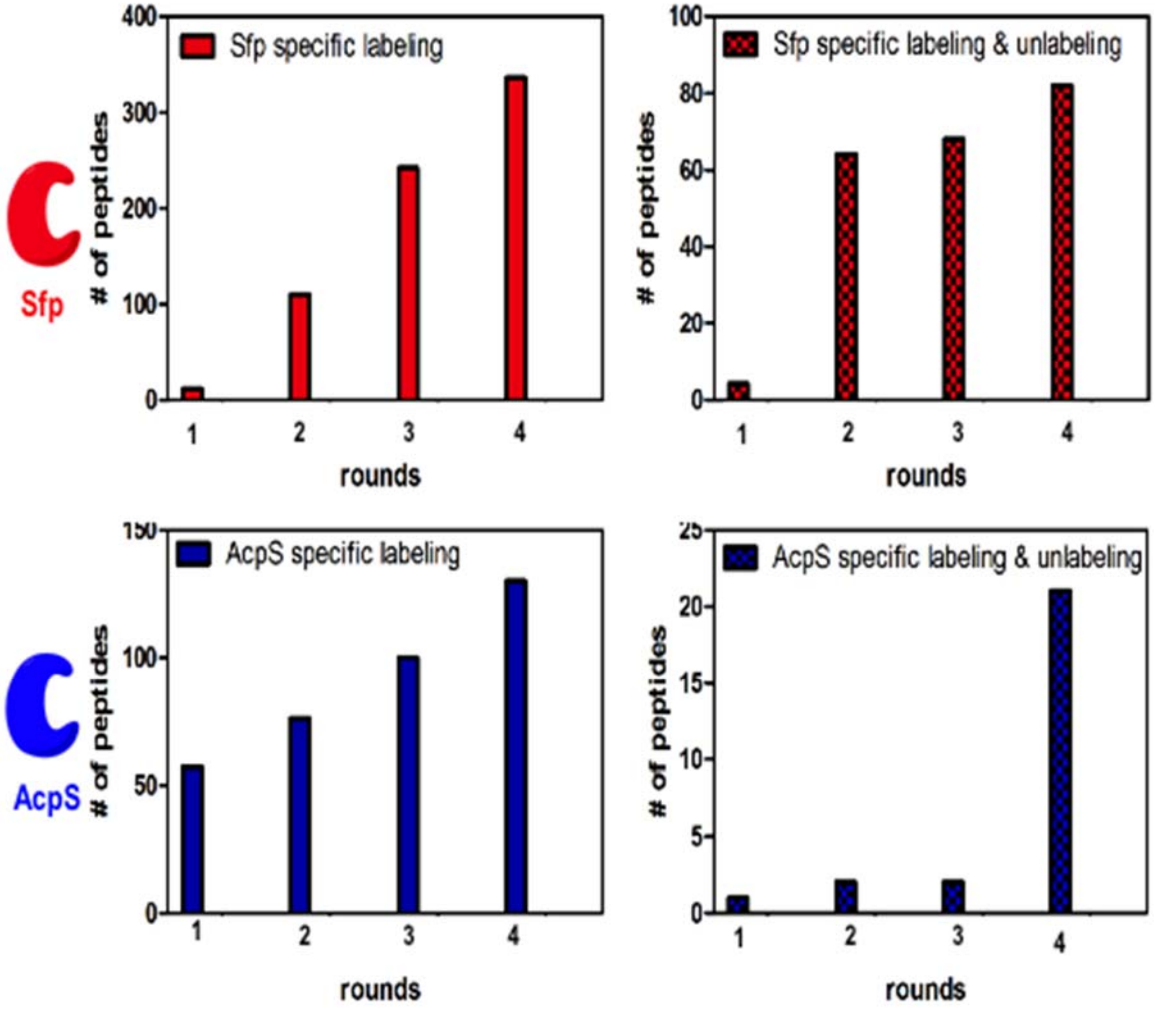


Figure 7: Discovery of novel peptide substrates over time using POOL v2.0 in the orthogonal reversible peptide labeling project. We picture progress in discovering each of the four types of hits sought (upper left, peptides that were labeled by Sfp and not AcpS; lower left, peptides that were labeled by AcpS and not Sfp; upper right, peptides that were labeled by Sfp, not by AcpS, and unlabeled by AcpH; and lower right, peptides that were labeled by AcpS, not by Sfp, and unlabeled by AcpH). For each type of hit, the total number of hits found versus the number of rounds of POOL is shown. We see that for each type of hit, POOL is able to increase the number of hits found over time.

Figure 8 shows a demonstration of reversible labeling, in which specifically labeled and unlabeled peptides were used to print letters on slides ("UCSD" using one of the specifically labeled peptides, and "AFOSR" with the other). Enzymes Sfp, AcpS and AcpH were then applied to demonstrate labeling and partial unlabeling: in the first step, Sfp was applied to label the first peptide (UCSD) with fluorescent dye, without affecting the second peptide. In the second step, AcpH was applied to unlabel this peptide. In the third step, AcpS was applied to label the second peptide (AFOSR) with a different fluorescent dye, without affecting the first peptide.

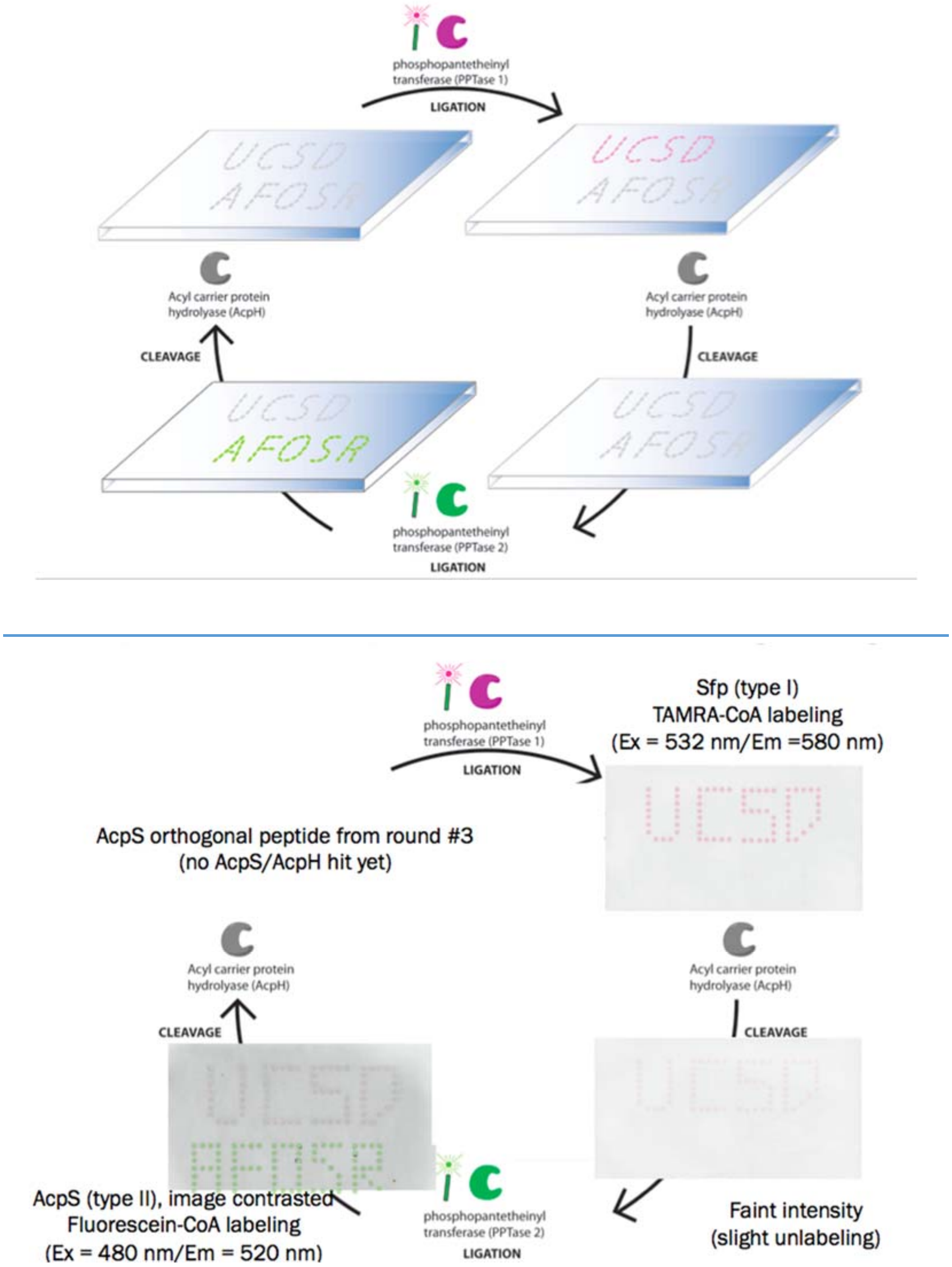

phosphopantetheinyl transferase (PPTase 1)
LIGATION
UCSD
AFOSR
Acyl carrier protein hydrolyase (AcpH)
CLEAVAGE
Acyl carrier protein hydrolyase (AcpH)
CLEAVAGE
UCSD
AFOSR
phosphopantetheinyl transferase (PPTase 2)
LIGATION
phosphopantetheinyl transferase (PPTase 1)
LIGATION
Sfp (type I)
TAMRA-CoA labeling
(Ex = 532 nm/Em =580 nm)
AcpS orthogonal peptide from round #3
(no AcpS/AcpH hit yet)
Acyl carrier protein hydrolyase (AcpH)
CLEAVAGE
Acyl carrier protein hydrolyase (AcpH)
CLEAVAGE
AcpS (type II), image contrasted
Fluorescein-CoA labeling
(Ex = 480 nm/Em = 520 nm)
phosphopantetheinyl transferase (PPTase 2)
LIGATION
Faint intensity
(slight unlabeling)

Figure 8: Demonstration of orthogonal reversible labeling using POOL v2.0. The upper diagram shows an idealized schematic of the experiment, while the bottom diagram shows images of the experimental results. In the first step, the letters "UCSD" printed using one peptide discovered using POOL v2.0 are labeled using Sfp without labeling the other letters. In the second step, these letters are unlabeled using AcpH. In the third step, the letters "AFOSR" printed using another peptide discovered using POOL v2.0 are labeled by AcpS.

3.5.3. Peptide with specific metal binding activity

In this ongoing joint work with Paras Prasad (Buffalo), Marc Knecht (Miami), and Tiffany Walsh (Deakin), we are using POOL v3.0 to suggest peptides to test in the search for peptides that are strong binders to one metal, and weak binders to another.

The discovery of these peptides will support the Prasad team's goal of creation of PARE-based macromolecules, in which nanoparticles of different types (e.g., gold and silver) will be functionalized by a peptide sequence comprising two specifically-binding peptides (blue and red in Figure 9) linked together by another peptide sequence (green) that can be controlled, e.g., through temperature or pH. This will allow the creation of reconfigurable assemblies of nanoparticles that exhibit novel optical, electronic, and photonic properties.

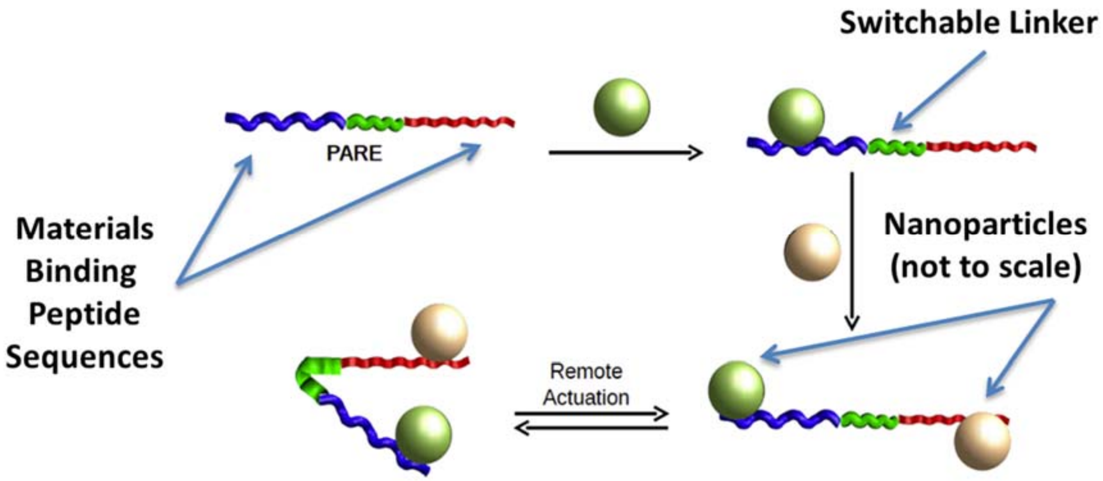

Switchable Linker
PARE
Materials Binding Peptide Sequences
Nanoparticles (not to scale)
Remote Actuation

Figure 9: Visualization of a PARE, in which nanoparticles are connected by switchable linkers to create reconfigurable assemblies of nanoparticles.

In this project, peptides are tested individually, rather than in batches (as they were in the reversible labeling projects), and the number that can be tested is much smaller than in the reversible labeling projects (10s instead of 1000s). This makes the peptide discovery problem more challenging. To overcome this challenge, POOL v3.0 uses quantitative responses to obtain more information from each measurement.

Although experiments are ongoing, and our scientific collaborators have not yet ascertained whether POOL will be able to successfully discover specific binders that achieve their scientific goals (we have tested two peptides thus far recommended by POOL), we have used simulation to study the performance of POOL, and to provide guidance on the risks of this project as a function of the experimental effort expended. This risk analysis is pictured in Figure 10.

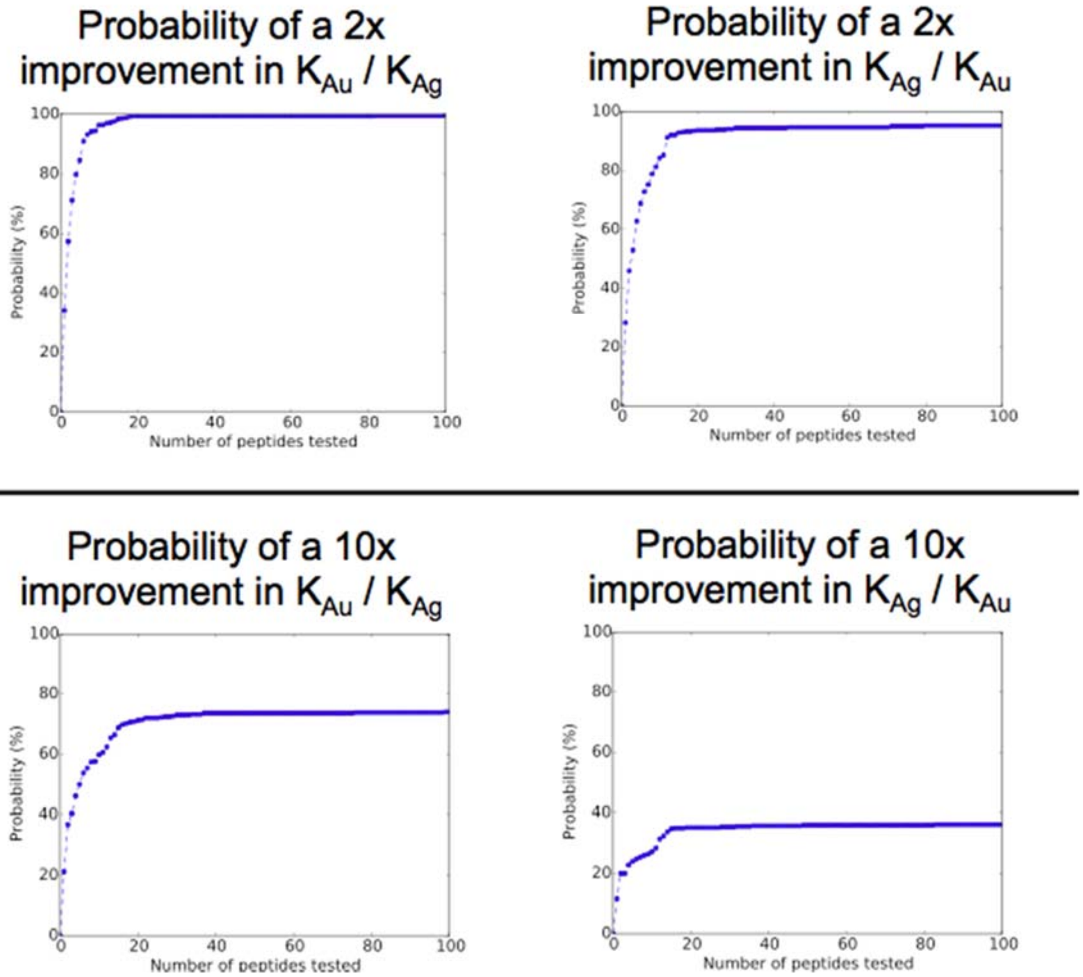


Figure 10: Predicted probability of success versus experimental effort expended, in the metal binding project in collaboration with the Prasad team based at Buffalo. Here, success is expressed as finding peptides whose ratio of binding coefficients (either gold to silver, or silver to gold) is improved over the best current specific binders by a given threshold. Through the generation of these plots, POOL can provide guidance to experimentalists regarding the overall probability of success in a given endeavor.

3.6. The future of POOL

Going forward, we are building on the success of the development of POOL in three ways:

- First, we are continuing to work with AFOSR-funded scientists to use POOL to support their scientific aims. In addition to ongoing collaborations with the Gianneschi and Prasad teams, the Mirkin lab, and AFRL, we made contact at the most recent 2015 AFOSR Natural Materials and Systems program review with Mark Blenner, Rein Ulijn, Carol Hall, and Carole Perry who are interested in using POOL in their own AFOSR-funded research.

- Second, we are continuing to improve the mathematical methodology underlying POOL, improving the accuracy of our statistical approach, the quality of our optimization and value of information analysis, and the generality of our approach.
- Third, as POOL becomes established as a scientific technique, we are exploring ways to make its application more standardized, either through software that would be installed and used by scientists, or through a web interface that would avoid the need for a software installation.

4. Software

We started our research with the hope that we could develop a general purpose, web-based package. Originally called "Dr. Watson" (or just "Watson"), we evolved to "hOLMES" (OLMES =- optimal learning for material experiments). However, as we worked with different scientific teams, we found that a general purpose package was much harder than we thought. The difficulty was that each problem seemed to exhibit unique structural qualities that required custom systems. Further, we came to appreciate that creating a general purpose software interface was simply well beyond what we could handle (especially while dealing with the custom problems, which also proved to be much more interesting from a methodological perspective).

We have, however, created a new, general purpose testing environment for optimal learning called MOLTE (Modular, Optimal Learning Testing Environment) which can be used by the methodological community to compare different learning algorithms. MOLTE makes it possible for researchers (in the mathematical learning community) to introduce new methods, as well as new problem settings, each of which are captured in their own Matlab-based ".m" file. The software, along with a detailed users manual, can be downloaded from

http://castlelab.princeton.edu/software.htm#molte

This environment should improve the relatively poor state of experimental work in the learning community. However, we have not yet generalized the ability to handle the more complex belief models that we encountered in different materials science settings. We believe that this can be handled by an extension where belief models are also represented in their own matlab modules which would have to be provided by the user.

We have also transitioned Bayesian optimization algorithms to industry, through the joint development of the Metrics Optimization Engine (MOE, https://github.com/yelp/moe) together with the tech company Yelp, and Frazier's former PhD student Scott Clark. MOE is an open source Bayesian global optimization engine for real-world metric optimization, where a "metric" is understood to be any performance measure. While the place where it has seen the most use is within the tech industry, by Yelp and by Netflix, the class of Bayesian optimization problems solved includes optimization of functions with low-dimensional vector-valued inputs (e.g., temperature and pressure), and is also of use in chemical discovery applications. This work has also spawned a startup company, Sigopt, http://sigopt.com/.

5. Education

We have accepted that one dimension of our work is an educational one. While we can develop tools to help guide scientists, such as showing the value of information, we also felt that we could add value to the scientific process by providing scientists with a principled approach to sequential design of experiments. This process consists of the following steps:

1. Belief construction – Before running any experiments, a scientist should capture what he/she already believes based on past experience and knowledge of the underlying physics and chemistry.
2. Articulating experimental choices – These are the *decisions* a scientist has to make. Interestingly, we have encountered situations where the scientist had not clearly articulated all the potential experimental choices. This can be overwhelming – in some cases these are overwhelmingly large.
3. Understand what you will (or might) learn from an experiment. Generally these are the laboratory measurements that will be made.
4. Belief updating – Understand how the results of your experiment will be used to update your belief.
5. Objectives – Articulate what you want to achieve from an experiment. This might be a combination of learning about the physics of the problem (e.g. learning unknown parameters), as well as trying to optimize some metric (maximizing the conductivity or strength of a material, or minimizing the deviation from a target release pattern).

These five components represent the fundamental elements of any sequential decision problem.

We have developed a series of PowerPoint presentations that were designed as a self-guided tutorial. These are available at

http://optimallearning.princeton.edu/tutorialsciences.htm

We have also written a tutorial article, which is to appear in an edited volume on informatics methods for materials scientists, with a preliminary version available here:

http://arxiv.org/pdf/1506.01349.pdf