

Submitted to *INFORMS Journal on Computing*  
manuscript (Please, provide the manuscript number!)

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

# A Knowledge Gradient Policy for Sequencing Experiments to Identify the Structure of RNA Molecules Using a Sparse Additive Belief Model

Yan Li, Kristofer G. Reyes

Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey 08544,  
{yanli@princeton.edu,kreyes@gmail.com}

Jorge Vazquez-Anderson

Department of Chemical Engineering, University of Texas at Austin, Austin, Texas 78712, jorgevazquez@utexas.edu

Yingfei Wang

Department of Computer Science, Princeton University, Princeton, New Jersey 08544, yingfei@cs.princeton.edu

Lydia M. Contreras

Department of Chemical Engineering, University of Texas at Austin, Austin, Texas 78712, lcontrer@che.utexas.edu

Warren B. Powell

Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey 08544,  
powell@princeton.edu

We present a sparse knowledge gradient (SpKG) algorithm for adaptively selecting the targeted regions within a large RNA molecule to identify which regions are most amenable to interactions with other molecules. Experimentally, such regions can be inferred from fluorescence measurements obtained by binding a complementary probe with fluorescence markers to the targeted regions. We use a biophysical model which shows that the fluorescence ratio under the log scale has a sparse linear relationship with the coefficients describing the accessibility of each nucleotide, since not all sites are accessible (due to the folding of the molecule). The SpKG algorithm uniquely combines the Bayesian ranking and selection problem with the frequentist  $\ell_1$  regularized regression approach Lasso. We use this algorithm to identify the sparsity pattern of the linear model as well as sequentially decide the best regions to test before experimental budget is exhausted. Besides, we also develop two other new algorithms: batch SpKG algorithm, which generates more suggestions sequentially to run parallel experiments; and batch SpKG with a procedure which we call length mutagenesis. It dynamically adds in new alternatives, in the form of types of probes, are created by inserting, deleting or mutating nucleotides within existing probes. In simulation, we demonstrate these algorithms on the Group I intron (a mid-size RNA molecule), showing that they efficiently learn the correct sparsity pattern, identify the most accessible region, and outperform several other policies.

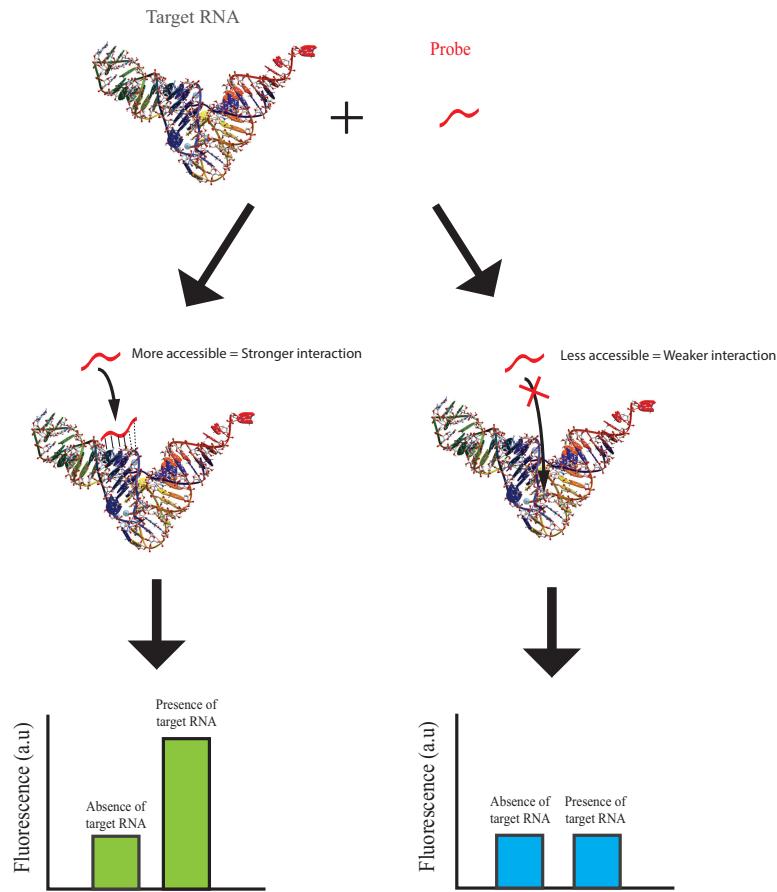
*Key words:* simulation; design of experiments; decision analysis; sequential; statistics: Bayesian

*History:*

## 1. Introduction

In recent years RNA has been rediscovered as a potent drug target with important implications to biotechnology and human health (Chan et al. 2006, Bennett and Swayze 2010, DeVos and Miller 2013, Vazquez-Anderson and Contreras 2013, Haning et al. 2015). Learning the structure of RNA molecules has become important in health research to improve the understanding of the interactions between RNA molecules and drugs. In addition, RNA regulates essential cellular processes through specific interactions with other biomolecules (e.g. proteins, other RNA or DNA molecules, etc). Disruption of an otherwise natural molecular interaction can potentially cause diseases. RNA can fold into intricate tridimensional structures making some regions accessible to interact with other molecules, while other regions remain inaccessible. Although biochemical technology has taken huge leaps by making RNA sequences readily available (Gelderman and Contreras 2013), significant progress is still required to understand RNA structure. The scientists on the team have made it possible to determine accessible regions using a fluorescence-based system (Sowa et al. 2015), where an RNA strand, hereafter referred to as a probe, interacts with a specific complementary region within a target RNA generating fluorescence (see Figure 1). This fluorescence directly correlates to the accessibility of a given region within a target RNA molecule. Although the *in vivo* RNA Structural Sensing System (iRS<sup>3</sup>), as named by Sowa et al. (2015), is a valuable tool as it provides the accessibility of a segment of interest within an RNA molecule in living cells, synthesizing and running the experiments in the absence of any *a priori* information of the RNA can be expensive and time-consuming. With the purpose of expanding its use to characterize a full molecule, we undertake the endeavor of optimizing the experimental settings of the iRS<sup>3</sup>. This paper seeks to use the knowledge gradient policy, adapted to a high-dimensional, sparse linear model to maximize the information gained from each experiment.

Our work addresses the problem of sequentially guiding experiments to identify the accessibility patterns of an RNA molecule known as the “Tetrahymena Group I intron” (gI intron), which has been widely used as an RNA folding model, and whose complex structure has been extensively characterized (Cech et al. 1981, Kruger et al. 1982, Cech



**Figure 1** Illustration of the *in vivo* RNA Structural Sensing System (iRS<sup>3</sup>)

*Notes.* A complementary probe with fluorescence marker is synthesized to bind to the targeted region of an RNA molecule. If the probe targets an accessible region (left), an interaction will occur leading to a strong fluorescence signal. Otherwise, if no major fluorescence signal is detected in the presence of the target RNA, it is interpreted as no interaction between the target RNA molecule and the probe (right).

et al. 1994, Kieft and Tinoco 1997, Golden et al. 1998, Russell et al. 2002, Koduvayur and Woodson 2004, Wan et al. 2010, Vazquez-Anderson and Contreras 2013). Determining these accessibility patterns is difficult to do *in silico*, as they depend on the complicated folding of the molecule known as the tertiary structure (Scherr et al. 2000, Mückstein et al. 2006). For details on *in silico* approaches, see Vazquez-Anderson and Contreras (2013). Experimentally, such accessibility patterns can be inferred from fluorescence measurements obtained from the iRS3 by using various complementary probes designed *a priori* to target a region within the intron (Sowa et al. 2015). However, the number of variations of the probe increases quadratically in the number of nucleotides; therefore, the number of candidate

probes is usually extremely large. A critical problem is therefore deciding which targeted regions should be tested, especially given the time and cost to perform each experiment.

The problem of identifying the accessibility pattern of an RNA molecule can be modeled mathematically as a ranking and selection (R&S) problem. By limiting the length of the probes, we are confronted with a collection of targeted regions of the RNA, which we call alternatives in R&S. In this R&S problem, we have a budget of measurements that we need to allocate sequentially to test the alternatives. As more information is collected, the belief distribution is changed or updated by conditioning on all the observations we have up to this point. Our goal is to maximize our ability to gain valuable information and the reward until the budget is exhausted. Although there are many papers on R&S problems, only a few make use of the model structure of the underlying belief model. The classical model for R&S is the lookup table belief model, which does not assume or exploit any structure. In identifying the accessibility pattern of an RNA molecule, where we may have tens or hundreds of thousands of alternatives, the lookup table strategy is computationally intractable. It may also be inappropriate when our goal is also to learn about the underlying model structure.

In this paper, we use a thermo-kinetic model which represents the log fluorescence level as a linear model of the weight coefficients representing the accessibility of each nucleotide. This coefficient vector is of the same dimension as that of the target molecule and thus can be high-dimensional. However, it is typically the case that only a small portion of these coefficients contain explanatory power, because not all sites are accessible due to the folding of the molecule. In such cases, a sparse linear model can offer considerably more flexibility than a linear model without sparsity structure. Therefore, we develop a sparse knowledge gradient (SpKG) policy for sequential experimental design (Li et al. 2015). This policy combines Bayesian R&S with a frequentist learning approach for recursive Lasso (Least Absolute Shrinkage and Selection Operator) (Tibshirani 1996, Garrigues and El Ghaoui 2008, Chen and Hero 2012), which is a well known  $\ell_1$  regularized version of least squares.

In this paper, we perform thorough testing of the SpKG algorithm using the setting of RNA accessibility identification. However, the SpKG algorithm is broadly applicable to learning problems with high-dimensional belief models, a problem domain that has been attracting considerable attention. Furthermore, SpKG can also be applied to problems with a known sparse group structure, as might happen when variables exhibit a natural

clustering. For example, applications in health might exhibit clusters of variables related to specific medical conditions. In addition, it can also handle interaction terms with smoothing spline ANOVA models (see Li et al. 2015).

It is worth noting that the SpKG algorithm is a unique and novel hybrid of Bayesian R&S with the frequentist  $\ell_1$  regularized regression known as Lasso. Both have been extensively studied in different fields (Friedman et al. 2001, Powell and Ryzhov 2012). For example, as a regularized version of least squares, Lasso minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. As a result of the penalty term, many of the coefficients will be exactly zero for large problems. Since Lasso was first proposed by Tibshirani (1996), there has been a considerable amount of work exploring the use of Lasso in different settings. It is useful in many settings due to its tendency to prefer solutions with fewer nonzero parameters, effectively reducing the problem dimension. For this reason, Lasso and its variants, such as elastic net regularization, are fundamental to many high-dimensional regression models.

Most of the previous work is in the classical batch setting, where we are given a dataset from the beginning, with no control over how the observations are chosen. However, our application requires guiding experiments in an online fashion. It leads to the acquisition of new information about the environment which may improve future decisions. For this purpose, we take advantage of a homotopy algorithm for using Lasso in a recursive setting, which was proposed by Garrigues and El Ghaoui (2008). This algorithm introduces an optimization problem to compute the exact update of the Lasso estimator when one additional observation is achieved. Additionally, Chen and Hero (2012) has extended this algorithm with an  $\ell_{1,\infty}$  group Lasso penalty. In this work, they consider a more general group sparsity system, which is composed of a few known nonoverlapping clusters of nonzero coefficients. The coefficients among each group have some correlation and are either all selected or not. Our work takes Lasso estimator updated from the homotopy algorithm as a sample from the true distribution of the coefficients. Then we use this to update both the conditional normal distribution of the coefficients and the Beta-Bernoulli conjugate distribution of the probability distribution of whether each coefficient is selected or not.

Discrete stochastic search problems have been approached in the literature from two directions: offline learning, widely known as ranking and selection (R&S) (Swisher et al. 2003), and online learning, often referred to as the multiarmed bandit problem (Gittins

et al. 2011). Either problem can be approached using frequentist or Bayesian approaches. For example, optimal computing budget allocation, or OCBA (Chen 2010, Chen et al. 2012), is a frequentist approach developed within the simulation optimization community for finding optimal designs that are tested using Monte Carlo simulation. Upper confidence bounding (UCB) policies, widely studied in the machine learning community for online bandit problems, are often approached as distribution-free strategies that enjoys bounds on the number of times that the wrong alternative might be tested (Auer et al. 2002, Bubeck and Cesa-Bianchi 2012). Frequentist approaches tend to require that each alternative be tested at least once.

A substantial literature has grown up around the general strategy of maximizing the value of information. Gupta and Miescke (1996) first proposed this idea for the offline ranking and selection problem, an idea that has been pursued under the name of the knowledge gradient (KG) (Frazier et al. 2008, 2009, Powell and Ryzhov 2012). Approximations of this idea have been proposed under names including sequential kriging optimization (SKO) (Huang et al. 2006), and efficient global optimization (EGO) (Jones et al. 1998, Bull 2011). Ryzhov et al. (2012) shows that the knowledge gradient can be easily applied to both online and offline problems. Furthermore, a batch KG policy based on Monte Carlo is proposed in the decision problems where the experimentalists may be able to run several parallel experiments in batches by maximizing the value of information for an entire batch (Wang et al. 2015).

Value of information policies are particularly well suited to problems where experiments are time consuming and expensive. The idea is particularly powerful when we can exploit belief models that capture some of the underlying structure of the problem (for example, linear belief models). This is the setting we face in this paper.

Identifying and validating RNA structures has been a problem of interest for the molecular biology community, especially since the catalytic properties of RNA were discovered back in 1982 (Cech et al. 1981, Gold et al. 1995). Unveiling the structure of an RNA molecule is critical to the understanding and exploitation of the interactions established with other RNA molecules. In this context structural accessibility becomes a central object of study. To this purpose the same experimental methods mentioned above have been used to understand this phenomenon, and a number of computational algorithms have

been developed to identify and characterize RNA interactions (Pain et al. 2015). A semi-empirical thermodynamic model common to a lot of the computational algorithms available to study RNA structure is the nearest neighbor method (SantaLucia 1998). A recent example of an algorithm to explain RNA-RNA interactions is the one proposed by Rodrigo et al. (2013). This paper has proposed a thermo-kinetic model including both Gibbs free energy and a kinetic function considering an intermediate stage (known as the seeding interaction). Another important strategy to predict and understand RNA structure is the use of the partition function and other stochastic methods. In this paper, we use a novel semi-empirical (since it uses experimental DMS footprinting data) thermo-kinetic model based on the nearest neighbor model parameters (Xia et al. 1998). We design a sequential experimentation policy based on maximizing the value of information (the knowledge gradient) using a Bayesian belief model. We showcase this strategy in the context of the setting of characterizing the structure of an RNA molecule using fluorescent probes.

This paper makes the following contributions. (1) We provide a sparse additive belief model for the fluorescence level produced by a probe applied to an RNA molecule. Here the log fluorescence level is a linear combination of the weight coefficients describing the accessibility of each nucleotide. The coefficient vector is sparse because not all sites contribute to the thermodynamic binding process. (2) We derive the batch SpKG policy which generates several suggestions sequentially to run parallel experiments. (3) We then introduce a new *length mutagenesis* procedure where new alternatives, in the form of types of probes, are created by inserting, deleting, or mutating nucleotides within existing probes. Then by each experiment we enlarge the alternative library by adding the one with the highest value of information from the larger library generated by the length mutagenesis procedure. (4) We demonstrate the effectiveness of the SpKG policy, with length mutagenesis and batch learning, in the setting of selecting probes to maximize fluorescence (as an indication of identifying accessible region) for RNA molecules with hundreds of potential sites.

The remainder of the paper is organized as follows. Section 2 describes the ranking and selection problem and the biophysical model. In section 3, we introduce the KG policy with a (nonsparse) linear belief model, and then describe the sparse linear model proposed by Li et al. (2015). Then we extend the SpKG algorithm to handle batch experimentation, as well as the new length mutagenesis procedure. Section 4 reports on the application of the

procedure to the in vitro DMS footprinting data with the RNA molecule Group I intron. Section 5 concludes the paper.

## 2. Model

We begin by considering a Bayesian R&S model where we have  $M$  alternatives. Let  $\mathcal{X}$  be a finite set consisting of the  $M$  alternatives and  $\mu_x : x \in \mathcal{X} \mapsto \mathbb{R}$  be a mapping from each alternative to its value. We have a budget of  $N$  measurements, and we wish to sequentially decide which alternative to measure so that we can find the best alternative when our budget is exhausted. Let  $\boldsymbol{\mu} = [\mu_1, \dots, \mu_M]^T$  (Table 1 provides a summary of the notation used in the paper). We assume that  $\boldsymbol{\mu}$  follows a multivariate normal distribution:

$$\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}). \quad (1)$$

**Table 1 Table of Notation**

Variable	Description
$M$	Number of alternatives/testing probes
$\mathcal{X}$	Set of alternatives
$\mu_x$	Unknown mean of alternative $x$
$N$	Number of measurements budget
$\boldsymbol{\mu}$	Column vector $(\mu_1, \dots, \mu_M)^T$
$x^i/x^i$	Sampling decision at time $i$ (vector or scalar index)
$y^{n+1}/y_x^{n+1}$	Sampling observation from measuring alternative $x^n$
$\epsilon_x^{n+1}$	Measurement error of alternative $x^n$
$\sigma_x$	Known standard deviation of alternative $x$
$\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n$	Mean and covariance of prior distribution on $\boldsymbol{\mu}$ at time $n$
$\Pi$	Set of all possible policies
$p$	Number of features/nucleotides
$\phi_k(i, j)$	Basis function for learning the local energetic value
$\Phi$	Linear transformation matrix
$\alpha_k$	Weight accessibility coefficient of nucleotide at site $k$
$\zeta_j$	Random indicator variable of $\alpha_j$
$\boldsymbol{\vartheta}^n, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, n}$	Mean and covariance of posterior distribution on $\boldsymbol{\alpha}$ after $n$ measurements
$(\xi_j^n, \eta_j^n)$	Set of shape parameters of beta distribution on $p_j^n$
$\hat{\boldsymbol{\vartheta}}^n$	Lasso estimate at time $n$
$(\hat{\boldsymbol{\vartheta}}_{\mathcal{S}}^n, \hat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta}, n})$	Mean and covariance matrix estimator from Lasso solution at time $n$
$S^n$	State variable, defined as the pair $(\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$
$v_x^{\text{KG}, n}$	Knowledge gradient value for alternative $x$ at time $n$
$L$	Number of possible sample realizations of $\zeta^n$
$p_j^n$	Parameter of Bernoulli distribution on $\zeta_j^n$
$K$	Number of batch measurement budget
$B$	Number of batch experiments at each time
$Q$	Number of Monte Carlo simulations
$(\boldsymbol{\theta}^{k,b}, \boldsymbol{\Sigma}^{k,b})$	Mean and covariance of posterior distribution on $\boldsymbol{\mu}$ after $k$ batches and additional $b$ measurements
$(\boldsymbol{\vartheta}^{k,b}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, k,b})$	Mean and covariance of posterior distribution on $\boldsymbol{\alpha}$ after $k$ batches and additional $b$ measurements

Consider a sequence of  $N$  sampling decisions,  $x^0, x^1, \dots, x^{N-1}$ . At time  $n$ , the measurement decision  $x^n$  selects an alternative from set  $\mathcal{X}$  to sample, and we observe

$$y_x^{n+1} = \mu_x + \epsilon_x^{n+1},$$

where  $\epsilon_x^{n+1} \sim \mathcal{N}(0, \sigma_x^2)$ , and  $\sigma_x$  is known. At the beginning, we may think of  $\boldsymbol{\mu}$  as a realization of the distribution given in (1), while the experimenter is only given some prior  $\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\theta}^0, \boldsymbol{\Sigma}^0)$ . Throughout the experiment, the experimenter is given the opportunity to better learn what value  $\boldsymbol{\mu}$  has taken through the sequential sampling decisions.

For convenience, let  $\mathcal{F}^n$  be the  $\sigma$ -algebra generated by the samples observed up to time  $n$ . Note that we have chosen our indexing so that random variables measurable with respect to the filtration  $\mathcal{F}^n$  are indexed by  $n$  in the superscript. Following this notation, let  $\boldsymbol{\theta}^n = \mathbb{E}(\boldsymbol{\mu}|\mathcal{F}^n)$ , and  $\boldsymbol{\Sigma}^n = \text{Var}(\boldsymbol{\mu}|\mathcal{F}^n)$ . This means the posterior distribution on  $\boldsymbol{\mu}$  is also multivariate normal with mean  $\boldsymbol{\theta}^n$  and covariance matrix  $\boldsymbol{\Sigma}^n$ . Let  $\Pi$  be the set of all  $\mathcal{F}^n$  measurable policies. That is  $\Pi := \{(x^0, \dots, x^{N-1}) : x^n \in \mathcal{F}^n\}$ . Our problem is to find the policy that solves

$$\sup_{\pi \in \Pi} \mathbb{E}^\pi \left[ \max_{x \in \mathcal{X}} \theta_x^N \right].$$

## 2.1. The Biophysical Model

In the RNA accessibility identification problem, let  $T$  be a molecule comprised of RNA nucleotides, called the target molecule. Denote the target molecule sequence as

$$T = (t_1, \dots, t_p),$$

where  $t_i \in \{A, C, G, U\}$  are the individual nucleotides. RNA molecules range from a few to thousands of nucleotides, but the typical length is several hundred. For the specific group I intron RNA molecule we work with in this study, it contains  $p = 414$  nucleotides. Depending on this sequence, the target molecule will fold upon itself in a thermodynamically favorable manner. The precise, three dimensional structure of this molecule upon folding is called the molecule's tertiary structure. Particularly, identifying regions of a molecule most amenable to interactions with other molecules is important to understanding how such interactions are mediated. Such regions depend on the molecule's tertiary structure. Those regions that are well protected in a mechanistic sense are less likely to interact with other molecules than those regions that are exposed. We refer to the regions more likely to interact with external

molecules as accessible regions. Identifying such regions is accomplished by sequentially and adaptively selecting the sites of the target molecule to bind a complementary RNA probe reporter. The RNA probe reporter includes a sequence, which is typically 8 to 16 nucleotides in length.

There is no precise definition of absolute accessibility of a region. However, we can think of the accessibility of a region relative to the accessibility of another region. This distinction is important experimentally. Therefore, to determine the accessibility of a region, the probe reporter also includes a fluorescent marker. The presence of the fluorescence at the end of an experiment, which the experimenter can measure optically, indicates whether the probe has successfully bound to the target region. The intensity of this fluorescence is an indication to how well this binding has occurred.

As described above, one can think of an “alternative” as a specific region within the target molecule or a complementary probe and its “value” as the amount of binding or the fluorescence level synonymously. We now describe briefly the biophysical model that connects both through a linear model.

Our main assumption is that the fluorescence measurements are a combination of mechanistic accessibility (kinetics) and change in Gibbs free energy between bound and unbound state (thermodynamics). To model this, we consider the accessibility profile

$$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p),$$

of the target molecule, where  $\alpha_k$  is a weight describing the relative accessibility of nucleotide  $k$  in the target molecule. This profile is generally unknown and can be estimated through some experimental data. Then the fluorescence intensity to target region  $[i, j]$  can be modeled as

$$\mu_{\text{bind}}(i, j) := \log \frac{[B]}{[U]} = \phi_0(i, j) + \sum_{k=1}^p \alpha_k \phi_k(i, j), \quad (2)$$

where  $\mu_{\text{bind}}(i, j)$  represents the amount of binding to the target region  $[i, j]$ ,  $[U]$  and  $[B]$  denote the fluorescence intensity of the bound and unbound state, respectively.  $\phi_0(i, j)$  is a base energy gradient value for attempting to bind to region  $[i, j]$ , and  $\phi_k(i, j)$  is the local energetic contribution of the  $k$ -th nucleotide position. Given the target molecule and under some assumptions, the values  $\phi_k(i, j)$  are known. It is the energy as measured for Watson-Crick Helices (Xia et al. 1998).

As mentioned above, the accessibility profile vector  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_p]^T$  is sparse, which means that many accessible values are zero or near zero. Mechanistically, this is a reasonable assumption, as we expect the tertiary structure of any sufficiently large molecule to be well-folded, meaning the proportion of exposed, mechanistically accessible regions to protected, inaccessible regions to scale like surface area to volume. Experimentally, the prior estimate from the experimental footprinting data also shows such property as one can see later in Section 4.1. Therefore, this model as shown in (2) is a sparse additive model.

## 2.2. The Bayesian Sparse Additive Model

The above model shows that the amount of binding  $\boldsymbol{\mu}$  is linear in the weight coefficients  $\boldsymbol{\alpha}$  representing the accessibility of each nucleotide. Here one can view a “feature” or an “attribute” as the accessibility of each nucleotide. Furthermore, we let  $[i^{(m)}, j^{(m)}]$  represent the region for the  $m$ -th alternative and  $\mu_m$  represent  $\mu_{\text{bind}}(i^{(m)}, j^{(m)})$ , then we can write (2) into the following affine system

$$\begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_M \end{pmatrix} = \begin{pmatrix} \phi_1(i^{(1)}, j^{(1)}) & \phi_2(i^{(1)}, j^{(1)}) & \cdots & \phi_p(i^{(1)}, j^{(1)}) \\ \phi_1(i^{(2)}, j^{(2)}) & \phi_2(i^{(2)}, j^{(2)}) & \cdots & \phi_p(i^{(2)}, j^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(i^{(M)}, j^{(M)}) & \phi_2(i^{(M)}, j^{(M)}) & \cdots & \phi_p(i^{(M)}, j^{(M)}) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{pmatrix} + \begin{pmatrix} \phi_0(i^{(1)}, j^{(1)}) \\ \phi_0(i^{(2)}, j^{(2)}) \\ \vdots \\ \phi_0(i^{(M)}, j^{(M)}) \end{pmatrix}.$$

Here if we write the basis matrix as  $\Phi \in \mathbb{R}^{M \times p}$  and the intercept vector as  $\Phi_0$ , the above linear equations can be written in the matrix form:  $\boldsymbol{\mu} = \Phi \boldsymbol{\alpha} + \Phi_0$ . Since  $\Phi_0$  is known, we can assume  $\Phi_0 = \mathbf{0}$  without loss of generality. Thus we have

$$\boldsymbol{\mu} = \Phi \boldsymbol{\alpha}, \tag{3}$$

where  $\boldsymbol{\mu}$  and  $\boldsymbol{\alpha}$  are random variables. Here  $\boldsymbol{\alpha}$  is sparse in the sense that most of its components are zero. In the Bayesian setting, whether each feature is zero or not is also random. Specifically, let  $\zeta = [\zeta_1, \dots, \zeta_p] \in \mathbb{R}^p$  be the indicator random variable of  $\boldsymbol{\alpha}$ , that is

$$\zeta_j = \begin{cases} 1 & \text{if } \alpha_j \neq 0 \\ 0 & \text{if } \alpha_j = 0 \end{cases}, \quad \text{for } j = 1, \dots, p.$$

Additionally, conditioning on  $\zeta$ , we assume that  $\boldsymbol{\alpha}$  follows a multivariate normal distribution with mean  $\boldsymbol{\vartheta}$  and covariance  $\Sigma^\vartheta$ , that is

$$\boldsymbol{\alpha} | \zeta \sim \mathcal{N}(\boldsymbol{\vartheta}, \Sigma^\vartheta).$$

Without loss of generality, conditioning on  $\zeta$ , we can permute the elements of  $\alpha$  and partition  $\alpha$  into the nonzero part and the zero part, so  $\alpha^T = [(\alpha_S)^T, \mathbf{0}]$ . Besides, conditioning on  $\zeta$ , the components of  $\vartheta$  and  $\Sigma^\vartheta$  are only nonzero where indexed by  $\mathcal{S}$ . Furthermore, conditioning on  $\zeta$ , we get that  $\mu$  follows a multinormal distribution through the linear transformation, that is

$$\mu \sim \mathcal{N}(\Phi\vartheta, \Phi\Sigma^\vartheta\Phi^T).$$

Combining this with (1), we have

$$\theta = \Phi\vartheta,$$

$$\Sigma = \Phi\Sigma^\vartheta\Phi^T.$$

The linear model in (3) allows us to maintain the belief model in the parameter space rather than a look up table belief model in the alternative space. In the case that the parameter structure is sparse, we use a frequentist learning approach (Lasso) which uses a least squares regression with an  $\ell_1$  regularization penalty, to update the belief model. In order to do this recursively, we introduce Beta-Bernoulli conjugate priors on each component of  $\zeta$ . Specifically, at time  $n$ , we have the following Bayesian model, for  $j, j' = 1, \dots, p$ ,

$$\alpha | \zeta^n = \mathbf{1} \sim \mathcal{N}(\vartheta^n, \Sigma^{\vartheta,n}), \quad (4)$$

$$\zeta_j^n | p_j^n \sim \text{Bernoulli}(p_j^n), \quad (5)$$

$$\zeta_j^n \perp\!\!\!\perp \zeta_{j'}^n, \quad \text{for } j \neq j', \quad (6)$$

$$p_j^n | \xi_j^n, \eta_j^n \sim \text{Beta}(\xi_j^n, \eta_j^n), \quad (7)$$

where  $p_j^n$  is the probability of the  $j$ -th feature being in the model, and  $(\xi_j^n, \eta_j^n)$  are the shape parameters for the beta distribution of  $p_j^n$ . We assume that  $\zeta_j^n$  and  $\zeta_{j'}^n$  are independent for different groups  $j$  and  $j'$ . Then after we get the new measurement  $(x^n, y^{n+1})$ , we recursively update the current Lasso estimate from  $\widehat{\vartheta}^n$  to  $\widehat{\vartheta}^{n+1}$  by the algorithm in Garrigues and El Ghaoui (2008) and sample a covariance matrix  $\widehat{\Sigma}^{\vartheta,n+1}$  from the first order optimality condition. For notational simplicity, we let  $\widehat{\vartheta}_{\mathcal{S}}^{n+1}$  and  $\widehat{\Sigma}_{\mathcal{S}}^{\vartheta,n+1}$  denote the nonzero parts, leaving the  $\mathcal{S}$  superscripted by time  $n + 1$  implicit. If we regard the Lasso estimate as a sample from the conditional multinormal distribution, we can use the following heuristic

updating scheme for a Beta-Bernoulli model and a normal-normal model. The updating equations are given by:

$$\boldsymbol{\Sigma}_{\mathcal{S}}^{\vartheta,n+1} = \left[ (\boldsymbol{\Sigma}_{\mathcal{S}}^{\vartheta,n})^{-1} + (\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\vartheta,n+1})^{-1} \right]^{-1}, \quad (8)$$

$$\boldsymbol{\vartheta}_{\mathcal{S}}^{n+1} = \boldsymbol{\Sigma}_{\mathcal{S}}^{\vartheta,n+1} \left[ (\boldsymbol{\Sigma}_{\mathcal{S}}^{\vartheta,n})^{-1} \boldsymbol{\vartheta}_{\mathcal{S}}^n + (\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\vartheta,n+1})^{-1} \widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1} \right], \quad (9)$$

$$\xi_j^{n+1} = \xi_j^n + 1, \quad \eta_j^{n+1} = \eta_j^n, \quad \text{for } j \in \mathcal{S}, \quad (10)$$

$$\xi_j^{n+1} = \xi_j^n, \quad \eta_j^{n+1} = \eta_j^n + 1, \quad \text{for } j \notin \mathcal{S}. \quad (11)$$

To illustrate, one can think of the hyperparameters  $(\xi_j, \eta_j)$  as the frequencies of “in” and “out” for each attribute and are updated through the Lasso estimates. Therefore,  $\xi_j^n / (\xi_j^n + \eta_j^n)$  can be viewed as approaching the probability of the  $j$ -th feature being nonzero as  $n$  becomes large. If the Lasso estimators can correctly recover the sparsity pattern asymptotically, then our approach should also identify the accessible nucleotides as the sampling budget goes to infinity. Theoretically, we have shown that our posterior mean estimate  $\boldsymbol{\vartheta}^n$  converges to the truth  $\boldsymbol{\vartheta}$  asymptotically under some conditions (Li et al. 2015).

### 3. The SpKG Algorithms

Before introducing the SpKG algorithm, we first briefly review the knowledge gradient policy with both a lookup table belief model and a nonsparse, linear belief model. The knowledge gradient policy for correlated beliefs (KGCB), as introduced in Frazier et al. (2009) is a fully sequential policy for learning correlated alternatives. At each time  $n$ , it makes the decision to measure the alternative with the largest expected incremental value, which is defined as

$$v_x^{\text{KG},n} = \mathbb{E}(\max_{x' \in \mathcal{X}} \theta_{x'}^{n+1} | S^n, x^n = x) - \max_{x' \in \mathcal{X}} \theta_{x'}^n$$

and

$$x^{\text{KG},n} = \arg \max_{x \in \mathcal{X}} v_x^{\text{KG},n},$$

where the knowledge state  $S^n$  is defined as  $S^n := (\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$ . The KG policy can be viewed as a gradient ascent algorithm. Maintaining a multivariate normal belief on the alternative space, we can update our belief by the Bayes rule and the Sherman-Morrison formula. Taking  $x^n = x$  to simplify subscripts, the updating equations are

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \frac{y_x^{n+1} - \theta_x^n}{\sigma_x^2 + \Sigma_{xx}^n} \boldsymbol{\Sigma}^n \mathbf{e}_x, \quad (12)$$

$$\boldsymbol{\Sigma}^{n+1} = \boldsymbol{\Sigma}^n - \frac{\boldsymbol{\Sigma}^n \mathbf{e}_x \mathbf{e}_x^T \boldsymbol{\Sigma}^n}{\sigma_x^2 + \Sigma_{xx}^n}, \quad (13)$$

where  $\mathbf{e}_x$  is the standard basis vector with one indexed by  $x$  and zeros elsewhere. We can further define a vector-valued function  $\tilde{\boldsymbol{\sigma}}$  as

$$\tilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x) = \frac{\boldsymbol{\Sigma}^n \mathbf{e}_x}{\sqrt{\sigma_x^2 + \Sigma_{xx}^n}},$$

and define a random variable

$$Z^{n+1} = \frac{(y_x^{n+1} - \theta_x^n)}{\sqrt{\text{Var}[y_x^{n+1} - \theta_x^n | \mathcal{F}^n]}},$$

which is a one-dimensional standard normal random variable when conditioned on  $\mathcal{F}^n$  (Frazier et al. 2008). Then we can write (12) as

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \tilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x^n) Z^{n+1}.$$

This allows us to compute the KG value as

$$\begin{aligned} v_x^{\text{KG}, n} &= \mathbb{E}(\max_{x' \in \mathcal{X}} \theta_{x'}^n + \tilde{\boldsymbol{\sigma}}_{x'}(\boldsymbol{\Sigma}^n, x^n) Z^{n+1} | S^n, x^n = x) - \max_{x' \in \mathcal{X}} \theta_{x'}^n \\ &= h(\boldsymbol{\theta}^n, \tilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x)). \end{aligned} \quad (14)$$

Here  $h : \mathbb{R}^M \times \mathbb{R}^M \mapsto \mathbb{R}$  is defined by  $h(\mathbf{a}, \mathbf{b}) = \mathbb{E}[\max_i a_i + b_i Z] - \max_i a_i$ , where  $\mathbf{a}$  and  $\mathbf{b}$  are deterministic  $M$ -dimensional vectors, and  $Z$  is any one-dimensional standard normal random variable. Frazier et al. (2009) provides a method to compute  $h(\mathbf{a}, \mathbf{b})$  with complexity  $O(M^2 \log M)$ .

In the case of a linear model when  $\boldsymbol{\mu} = \boldsymbol{\Phi}\boldsymbol{\alpha}$  and we do not hold sparsity belief on  $\boldsymbol{\alpha}$ , there exists recursive least squares (RLS) updating equations for  $(\boldsymbol{\vartheta}^n, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, n})$  that are similar to the recursive updating in (12) and (13). Before providing the updating equations, we first introduce some additional notation. Let  $\boldsymbol{\phi}^n = [\phi_1^n, \dots, \phi_p^n]^T$  be the column vector describing the alternative that is measured at time  $n$  after the basis transformation  $\phi$ . Then, the following updating equations result from standard expressions for normal sampling of linear combinations of features (see Powell and Ryzhov 2012, p. 187):

$$\boldsymbol{\vartheta}^{n+1} = \boldsymbol{\vartheta}^n + \frac{\hat{\epsilon}^{n+1}}{\gamma^n} \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, n} \boldsymbol{\phi}^n, \quad (15)$$

$$\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, n+1} = \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, n} - \frac{1}{\gamma^n} (\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, n} \boldsymbol{\phi}^n (\boldsymbol{\phi}^n)^T \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, n}), \quad (16)$$

where  $\hat{\epsilon}^{n+1} = y^{n+1} - (\boldsymbol{\vartheta}^n)^T \boldsymbol{\phi}^n$ , and  $\gamma^n = \sigma_x^2 + (\boldsymbol{\phi}^n)^T \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, n} \boldsymbol{\phi}^n$ . For the KG computation, we can just plug the linear transformation  $\boldsymbol{\theta} = \boldsymbol{\Phi}\boldsymbol{\vartheta}$  and  $\boldsymbol{\Sigma} = \boldsymbol{\Phi}\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}\boldsymbol{\Phi}^T$  into equation (14). The linear model exponentially reduces the computational and storage requirements of the lookup table model. The essential idea is to maintain a belief on the attributes. A summary of the KG policy with a nonsparse, linear belief model is outlined in Algorithm 1.

---

**Algorithm 1** Knowledge gradient algorithm with nonsparse linear belief

---

**Require:**  $\vartheta^0, \Sigma^{\vartheta,0}, \Phi$ .

```

1: for  $n = 0$  to  $N - 1$  do
2:   for  $x = 1$  to  $M$  do
3:      $\mathbf{a} \leftarrow \Phi \vartheta^n$ 
4:      $\mathbf{b} \leftarrow \Sigma_{x,*}^n / \sqrt{\sigma_x^2 + \Sigma_{xx}^n}$ 
5:      $v \leftarrow h(\mathbf{a}, \mathbf{b})$ 
6:     if  $x = 1$  or  $v > v^*$  then
7:        $v^* \leftarrow v, x^* \leftarrow x$ 
8:     end if
9:   end for
10:   $x^n = x^*$ 
11:  Get a new measurement:  $(\mathbf{x}^n, y^{n+1})$ 
12:   $\vartheta^{n+1} \leftarrow \vartheta^n \% RLS \ update \ by \ (15) \ (16)$ 
13:   $\Sigma^{\vartheta,n+1} \leftarrow \Sigma^{\vartheta,n}$ 
14: end for
15: return  $\vartheta^N, \Sigma^{\vartheta,N}$ .

```

---

### 3.1. The SpKG Algorithm

It is worth noting that, for the sparse linear belief, we introduce the random variable  $\zeta_j$  to indicate if the  $j$ -th attribute is selected or not. As a result, the KG calculation in (14) needs to be modified so that the expectation is also taken over  $\zeta$ .

Specifically, at time  $n$ , the Bayesian model is as described in (4)-(7). The prior  $\zeta^n$  is a discrete random variable. Let  $\zeta^{n,1}, \dots, \zeta^{n,L}$  be all the possible realizations of  $\zeta^n$ , and  $\mathbb{P}(\zeta^n = \zeta^{n,l}) = p^{n,l}, l = 1, \dots, L$ . To compute the KG value, we need to approximate the distribution of  $(\zeta^{n+1}, \mathbf{p}^{n+1})$  by that of  $(\zeta^n, \mathbf{p}^n)$ . This is because the change of the sparsity belief depends on the next observation and the Lasso algorithm, and thus can be very complicated to model. Therefore, by the Law of Total Expectation, the KG value can be computed by weighting over all the possible sparsity structures (Li et al. 2015):

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

$$\begin{aligned}
&= \sum_{l=1}^L \mathbb{E}_{p^n}(p^{n,l}) h(\mathbf{a}^{n,l}, \mathbf{b}^{n,l}) \\
&= \sum_{l=1}^L \prod_{\{j: \zeta_j^{n,l}=1\}} \frac{\xi_j^n}{\xi_j^n + \eta_j^n} \prod_{\{j: \zeta_j^{n,l}=0\}} \frac{\eta_j^n}{\xi_j^n + \eta_j^n} h(\mathbf{a}^{n,l}, \mathbf{b}^{n,l}),
\end{aligned} \tag{17}$$

where

$$\begin{aligned}
\mathbf{a}^{n,l} &= \Phi_{*, \zeta^{n,l}} \boldsymbol{\vartheta}_{\zeta^{n,l}}, \\
\mathbf{b}^{n,l} &= \tilde{\sigma}(\Phi_{*, \zeta^{n,l}} \Sigma_{\zeta^{n,l}}^{\boldsymbol{\vartheta}, \boldsymbol{\vartheta}} (\Phi_{*, \zeta^{n,l}})^T, x).
\end{aligned}$$

Here  $h$  is the function defined in (14). The subscript  $*, \zeta^{n,l}$  means the submatrix is taken with all the rows and columns indexed by  $\zeta^{n,l}$ . The same notation is used throughout the paper. Since  $L$  can be as large as  $2^p$ , we can sort the weights and approximate the KG value approximately by only computing the ones with the largest probabilities. In that case, we approximately compute the KG value to avoid the curse of dimensionality. This is reasonable because we expect the sparsity pattern to converge as  $n$  becomes large. We summarize the SpKG in Algorithm 2.

---

**Algorithm 2** Sparse knowledge gradient algorithm

---

**Require:**  $\boldsymbol{\vartheta}^0, \Sigma^{\boldsymbol{\vartheta}, 0}, \{\xi_j^0, \eta_j^0\}_{j=1}^p, \Phi, N$ , regularization tunable parameters  $\{\lambda^i\}_{i=0}^N$ .

- 1: **for**  $n = 0$  to  $N - 1$  **do**
  - 2:   Compute KG by (17):  $x^n = \arg \max v_x^{\text{KG}, n}$  %compute  $h$  as in Algorithm 1
  - 3:   Lasso homotopy update:  $\hat{\boldsymbol{\vartheta}}^n, (\mathbf{x}^n, y^{n+1}) \in \mathbb{R}^m \times \mathbb{R}, \lambda^n, \lambda^{n+1} \rightarrow \hat{\boldsymbol{\vartheta}}^{n+1}$
  - 4:   Approximately simulate  $\hat{\Sigma}_S^{\boldsymbol{\vartheta}, n+1}$
  - 5:    $\boldsymbol{\vartheta}^{n+1} \leftarrow \boldsymbol{\vartheta}^n, \Sigma^{\boldsymbol{\vartheta}, n+1} \leftarrow \Sigma^{\boldsymbol{\vartheta}, n}, \{\xi_j^{n+1}, \eta_j^{n+1}\}_{j=1}^p \leftarrow \{\xi_j^n, \eta_j^n\}_{j=1}^p$  by (8)-(11)
  - 6: **end for**
  - 7: **return**  $\boldsymbol{\vartheta}^N, \Sigma^{\boldsymbol{\vartheta}, N}, \{\xi_j^N, \eta_j^N\}_{j=1}^p$ .
- 

### 3.2. The Batch SpKG Algorithm

The sequential knowledge gradient policy fails to account for the ability of experimentalists to run several experiments in parallel or in batches. Batch experiment means running a group of experiments at the same time. For example, the experimenter could divide a plate into squares, where each square is a different experiment, but the plate is immersed in

a chemical bath at the same time. Doing experiments in parallel has the connotation of literally running different experiments on different machines at the same time. For example, in this RNA problem, the experimenter can synthesize three probes in parallel to test their fluorescence intensities all in one run. Our batch SpKG algorithm can deal with both “batch” and “parallel” settings. To handle such experimental settings, Wang et al. (2015) proposes a Monte Carlo based batch knowledge gradient (BKG) approach to guide the batch experimental design by maximizing the value of information for an entire batch with the lookup table belief model. In this section, we first review the BKG policy with lookup table belief model and linear belief model, after which we then derive the new Batch SpKG policy.

To begin, we modify our notation to fit the batch measurements. Suppose we are given a batch measurement budget of  $K$  with  $B$  batch decisions at each time. Then the total number of measurements allowed is  $N = BK$ . Now at time step  $k$ , we choose to measure a batch of  $B$  alternatives  $\mathbf{x}^{k,0}, \mathbf{x}^{k,1}, \dots, \mathbf{x}^{k,B-1}$  simultaneously and get  $B$  observations  $y^{k+1,0}, \dots, y^{k+1,B-1}$ . We also use superscript  $(k, b)$  to index our beliefs. For example, the prior multivariate normal belief can be rewritten as  $(\boldsymbol{\theta}^{0,0}, \boldsymbol{\Sigma}^{0,0})$ . The superscript  $(k, b)$  is understood as meaning that we have done  $k$  batches and used  $\mathbf{x}^{k,0}, \dots, \mathbf{x}^{k,b-1}, y^{k+1,0}, \dots, y^{k+1,b-1}$  to update our belief. Similarly to equations (12) and (13), the new updating equations can be written recursively for a batch measurements as

$$\boldsymbol{\theta}^{k,b+1} = \boldsymbol{\theta}^{k,0} + \sum_{j=0}^b \frac{y^{k+1,j} - \theta_{x^{k,j}}^{k,j}}{\sigma_{x^{k,j}}^2 + \Sigma_{x^{k,j}x^{k,j}}^{k,j}} \boldsymbol{\Sigma}^{k,j} e_{x^{k,j}}, \quad (18)$$

$$\boldsymbol{\Sigma}^{k,b+1} = \boldsymbol{\Sigma}^{k,b} - \frac{\boldsymbol{\Sigma}^{k,b} e_{x^{k,b}} (e_{x^{k,b}})^T \boldsymbol{\Sigma}^{k,b}}{\sigma_{x^{k,j}}^2 + \Sigma_{x^{k,b}x^{k,b}}^{k,b}}, \quad (19)$$

where  $k = 0, 1, \dots, K-1$ ,  $b = 0, 1, \dots, B-1$ ,  $\boldsymbol{\theta}^{k+1,0} = \boldsymbol{\theta}^{k,B}$ , and  $\boldsymbol{\Sigma}^{k+1,0} = \boldsymbol{\Sigma}^{k,B}$ . It is worth emphasizing that in the batch setting the covariance matrix would be updated within a batch since it is determined by the measurement decisions and is independent of the observations, whereas the mean values  $\boldsymbol{\theta}^{k,b}$  are only updated after the observations are collected for the whole batch.

The batch knowledge gradient policy greedily adds in each alternative that maximizes the expected incremental value one at a time until  $B$  alternatives are chosen. The expected incremental value of measuring alternatives  $x_1, \dots, x_j$  at time step  $k$  is defined as

$$v_{x_1, \dots, x_j}^{\text{BKG}}(S^k) = \mathbb{E}[\max_x \theta_x^{k+1} - \max_x \theta_x^k | x^{k,0} = x_1, \dots, x^{k,j-1} = x_j, S^k].$$

The batch knowledge gradient policy has the decision function

$$x^{k,b} := X_b^{\text{BKG}}(S^k) = \arg \max_{x \in \mathcal{X}} v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BKG}}(S^k),$$

which can be rewritten as

$$X_b^{\text{BKG}}(S^k) = \arg \max_{x \in \mathcal{X}} \mathbb{E} \left[ \max_{x'} \left( \boldsymbol{\theta}^{k,0} + \sum_{j=0}^{b-1} \tilde{\sigma}(\boldsymbol{\Sigma}^{k,j}, x^{k,j}) Z^{k+1,j} + \tilde{\sigma}(\boldsymbol{\Sigma}^{k,b}, x) Z^{k+1,b} \right) \right], \quad (20)$$

according to equation (18) and (19). Notice that here  $x^{k,j}$ ,  $j \leq b$  are fixed when choosing  $x^{k,b}$ , and  $\boldsymbol{\Sigma}^{k,j}$  can be updated within a batch according to (19). Since an analytic expression for the expected maximization as in (20) is unknown, Monte Carlo sampling is used to approximate the expectation. The pseudo-code of the BKG for the  $k$ -th batch decision and the Monte Carlo algorithm are presented in Algorithm 3 and Algorithm 4.

---

**Algorithm 3** Batch knowledge gradient policy with lookup table belief for the  $k$ -th batch decision

---

**Require:**  $\boldsymbol{\theta}^{k,0}$ ,  $\boldsymbol{\Sigma}^{k,0}$ , and the number of sample  $Q$  for the Monte Carlo simulation

- 1: Use the KGCB policy to find  $x^{k,0}$
  - 2:  $\tilde{\sigma}^0 \leftarrow \tilde{\sigma}(\boldsymbol{\Sigma}^{k,0}, x^{k,0})$
  - 3: Update  $\boldsymbol{\Sigma}^{k,1}$  according to (19)
  - 4: **for**  $b = 1$  to  $B - 1$  **do**
  - 5:   Use Algorithm 4 below to calculate  $v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BKG}}$
  - 6:    $x^{k,b} = \arg \max_{x \in \mathcal{X}} v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BKG}}$
  - 7:    $\tilde{\sigma}^b \leftarrow \tilde{\sigma}(\boldsymbol{\Sigma}^{k,b}, x^{k,b})$
  - 8:   Update  $\boldsymbol{\Sigma}^{k,b+1}$  according to (19)
  - 9: **end for**
  - 10: **return** batch decisions  $x^{k,0}, x^{k,1}, \dots, x^{k,B-1}$
- 

Besides, the logic of the batch knowledge gradient policy can be generalized to a linear belief model. In this case, instead of recursively updating  $\boldsymbol{\theta}^{k,b}$  and  $\boldsymbol{\Sigma}^{k,b}$  directly as in (18) and (19), we recursively update  $\boldsymbol{\vartheta}^{k,b}$  and  $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, k, b}$  for a batch of observations through RLS and use the linear transformation  $\boldsymbol{\theta} = \boldsymbol{\Phi} \boldsymbol{\vartheta}$  and  $\boldsymbol{\Sigma} = \boldsymbol{\Phi} \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}} \boldsymbol{\Phi}^T$ :

$$\boldsymbol{\vartheta}^{k+1,b+1} = \boldsymbol{\vartheta}^{k,0} + \sum_{j=0}^b \frac{\hat{\epsilon}^{k+1,j}}{\gamma^{k,j}} \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, k, j} \boldsymbol{\phi}^{k,j}, \quad (21)$$

$$\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, k+1, b+1} = \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, k, b} - \frac{1}{\gamma^{k,b}} (\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, k, b} \boldsymbol{\phi}^{k,b} (\boldsymbol{\phi}^{k,b})^T \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, k, b}), \quad (22)$$

---

**Algorithm 4** Monte Carlo simulation for calculating KG values

---

**Require:**  $b$ ,  $\boldsymbol{\theta}^{k,0}$ ,  $\tilde{\boldsymbol{\sigma}}^0, \tilde{\boldsymbol{\sigma}}^1, \dots, \tilde{\boldsymbol{\sigma}}^{b-1}, \boldsymbol{\Sigma}^{k,b}$ , and  $Q$

```

1: for all  $x \in \mathcal{X}$  do
2:    $\text{sum}_x = 0$ 
3:   for  $q = 1$  to  $Q$  do
4:     for  $j = 0$  to  $b$  do
5:       Generate a realization  $z_q^j$  of  $Z^{k,j}$ 
6:     end for
7:     temp  $\leftarrow \max_{x'} (\boldsymbol{\theta}_{x'}^{k,0} + \sum_{j=0}^{b-1} \tilde{\boldsymbol{\sigma}}_{x'}^j z_q^j + \tilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^{k,b}, x) z_q^b)$ 
8:     sumx  $\leftarrow \text{sum}_x + \text{temp}$ 
9:   end for
10:  end for
11:  return sum/Q

```

---

where  $\hat{\epsilon}^{k+1,j} = y^{k+1,j} - (\boldsymbol{\vartheta}^{k,j})^T \boldsymbol{\phi}^{k,j}$ , and  $\gamma^{k,j} = \sigma_x^2 + (\boldsymbol{\phi}^{k,j})^T \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},k,j} \boldsymbol{\phi}^{k,j}$ .

Furthermore, for the sparse linear model, the posterior distribution of the sparsity structure parameter  $\zeta$  will be updated according to (10) and (11) after the observations are revealed for the whole batch. Since  $\boldsymbol{\alpha}|\zeta^k \sim \mathcal{N}(\boldsymbol{\vartheta}^k, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},k})$ , then given  $\zeta$ ,  $\boldsymbol{\vartheta}_{\zeta^k,l}^{k,b}$  and  $\boldsymbol{\Sigma}_{\zeta^k,l}^{\boldsymbol{\vartheta},k,b}$  can be updated according to (21) and (22). The batch SpKG algorithm works by greedily adding in each alternative that maximizes the expected marginal value until  $B$  alternatives are chosen given the sparsity structure unchanged within a batch. The KG value of measuring alternatives  $x_1, \dots, x_j$  at time step  $k$  can be computed as

$$v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BSpKG}}(S^k) = \sum_{l=1}^L \mathbb{E}_{p^k}(p^{k,l}) v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BKG}}(S^{\text{Sp},k,l}), \quad (23)$$

where  $S^{\text{Sp},k,l}$  is the knowledge state given  $\zeta^{k,l}$ , and  $S^{\text{Sp},k,l} = (\boldsymbol{\Phi}_{*,\zeta^{k,l}} \boldsymbol{\vartheta}_{\zeta^{k,l}}^{k,0}, \boldsymbol{\Phi}_{*,\zeta^{k,l}} \boldsymbol{\Sigma}_{\zeta^{k,l}}^{\boldsymbol{\vartheta},k,0} (\boldsymbol{\Phi}_{*,\zeta^{k,l}})^T)$ . After  $x^{k,0}, \dots, x^{k,b-1}$  are chosen,  $v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BKG}}(S^{\text{Sp},k,l})$  can be approximated using Algorithm 4. Then the  $b$ -th decision within a batch is given by  $x^{k,b} = \arg \max_x v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BSpKG}}(S^k)$ .

The batch SpKG algorithm can be summarized in Algorithm 5.

### 3.3. The Batch SpKG Algorithm with Length Mutagenesis

For all of the above algorithms, we use a fixed set of discrete alternatives  $\mathcal{X}$ . For this RNA accessibility identification problem, if we consider the probe sequence with length of

**Algorithm 5** Batch sparse knowledge gradient policy for the  $k$ -th batch decision

---

**Require:**  $\vartheta^{k,0}, \Sigma^{\vartheta,k,0}$ , and the number of sample  $Q$  for the Monte Carlo simulation

```

1: Use Algorithm 2 to find  $x^{k,0}$ ;
2: for  $l = 1$  to  $L$  do
3:    $\tilde{\sigma}^{0,l} \leftarrow \tilde{\sigma}(\Phi_{*,\zeta^{k,l}} \Sigma_{\zeta^{k,l}}^{\vartheta,k,0} (\Phi_{*,\zeta^{k,l}})^T, x^{k,0})$ 
4:   Update  $\Sigma_{\zeta^{k,l}}^{\vartheta,k,1}$  according to (22)
5: end for
6: for  $b = 1$  to  $B - 1$  do
7:   for  $l = 1$  to  $L$  do
8:     Use Algorithm 4 to calculate  $v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BKG}}(S^{Sp,k,l})$  with input parameters  $\theta^{k,0} = \Phi_{*,\zeta^{k,l}} \vartheta_{\zeta^{k,l}}^{k,0}$ ,  $\Sigma^{k,l} = \Phi_{*,\zeta^{k,l}} \Sigma_{\zeta^{k,l}}^{\vartheta,k,b} (\Phi_{*,\zeta^{k,l}})^T$ , and  $\tilde{\sigma}^{0,l}, \dots, \tilde{\sigma}^{b-1,l}$ 
9:   end for
10:  Calculate  $v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BSpKG}}$  according to (23)
11:   $x^{k,b} = \arg \max_{x \in \mathcal{X}} v_{x^{k,0}, \dots, x^{k,b-1}, x^{k,b}=x}^{\text{BSpKG}}$ 
12:  for  $l = 1$  to  $L$  do
13:     $\tilde{\sigma}^{b,l} \leftarrow \tilde{\sigma}(\Phi_{*,\zeta^{k,l}} \Sigma_{\zeta^{k,l}}^{\vartheta,k,b} (\Phi_{*,\zeta^{k,l}})^T, x^{k,b})$ 
14:    Update  $\Sigma_{\zeta^{k,l}}^{\vartheta,k,b+1}$  according to (22)
15:  end for
16: end for
17: return batch decisions  $x^{k,0}, x^{k,1}, \dots, x^{k,B-1}$ 

```

---

8~16, the number of alternatives can be  $\sim 4000$ . Including all of them in the alternative library would be computationally expensive while working with only a small subset would possibly miss the most accessible region. To compromise, we propose a novel procedure that we call *length mutagenesis*, which sequentially enlarges the probe library through an adaptive probe refinement procedure. We simply refer to this as “mutagenesis” throughout the remainder of the paper for compactness. The mutagenesis works as follows. Suppose that at time  $n$  we have a library of probes that we denote as

$$\mathcal{X}^n = \{x_1, \dots, x_{M^n}\}.$$

For the next experiment, we could consider a larger library of sub-probes obtained through mutagenesis. With mutagenesis, we either add or delete nucleotides at one end of a probe.

For now, let us think of each alternative  $x$  as a probe sequence representing a region in the target molecule. Specifically, given a probe  $x$ , we can alter it through a round of mutagenesis to get a new probe  $x'$  of the form

$$x = [i, j] \rightarrow x' = \begin{cases} [i+k, j], & i+k < j \\ [i+k, j], & i < j+k \end{cases}, \quad 0 < |k| \leq 7.$$

Since a probe of length less than 4 does not necessarily bind to the correct targeted region experimentally, we limit the probe length to be no less than 4. Then for a probe  $x$ , let  $\mathcal{M}(x)$  denote the set of possible probes obtained from  $x$  through mutagenesis. At time  $n$ , we get an expanded library through mutagenesis, that is

$$\bar{\mathcal{X}}^n = \mathcal{X}^n \cup \bigcup_{i=1}^{M^n} \mathcal{M}(x_i).$$

From this expanded library, we pick the alternative with the highest KG score, that is

$$x^n = \arg \max_{x \in \bar{\mathcal{X}}^n} v_x^{\text{KG}, n}.$$

Then we add this probe to our library if it is new,

$$\mathcal{X}^{n+1} = \mathcal{X}^n \cup \{x^n\}.$$

This approach allows us to add a new probe which potentially has a higher fluorescence signal at each time and work dynamically with the alternative library. The computation is simpler and more efficient than maintaining all the possible alternatives. As shown in the simulations in Figure 8, we have a much better chance of obtaining information about the accessibility of the molecule relative to maintaining a fixed probe region.

#### 4. Empirical Study

In this section, we present the simulation results of the SpKG algorithms for the RNA accessibility identification problem described in Section 2. The target RNA molecule is the Group I intron, a mid-size RNA molecule sequence with a length of 414. Due to the nature of this problem, we are not able to identify the 21 nucleotides at one end of the molecule sequence. Therefore, we only work with a sequence with length of 393.

In Section 4.1, we describe the prior in vitro DMS footprinting data and the methods for generating the prior covariance matrix. In Section 4.2, we present the simulation results of the performance of SpKG on a collection of probe sequences as well as those of the batch SpKG algorithm and the batch SpKG with mutagenesis scheme. We also compare these policies with several other policies.

#### 4.1. Prior Distribution

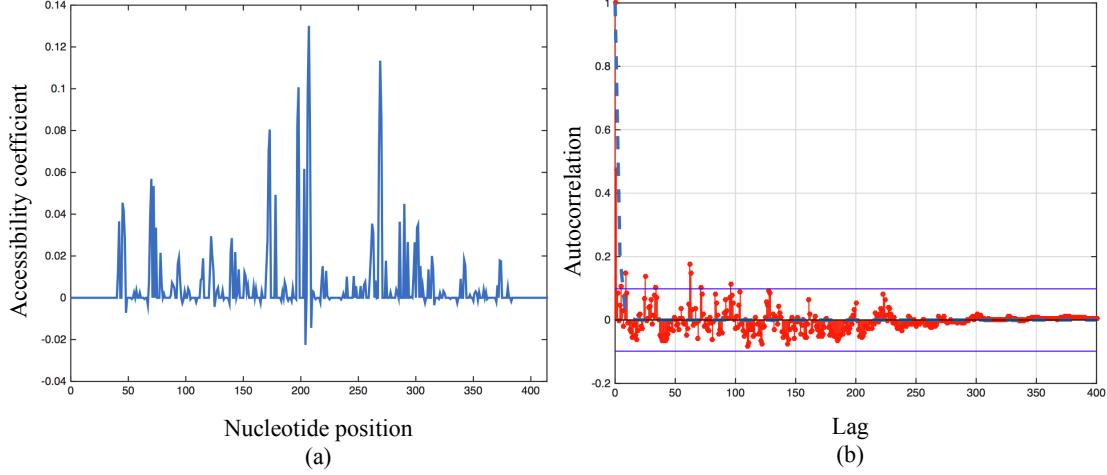
When choosing a prior distribution, the domain experts can have many ways to articulate their prior belief on the accessible regions. In this problem, we have the accessibility profile obtained from the in vitro DMS footprinting. The DMS footprinting is a standard chemical method to study RNA structure. It relies on the reactivity of a small molecule Di-methyl sulfate (DMS) with the base-pairing molecular faces of adenosines and cytidines (A and C). The higher the DMS reactivity is for a nucleotide site, the more the nucleotide is exposed. By reversely transcribing the DMS reacted RNA into DNA, we can determine sites of reaction and thus the levels of protection exposure at a single-nucleotide resolution. Here we use in vitro DMS data from Russell et al. (2006) as an initial estimation of nucleotide accessibility. One may think of this dataset as providing the priors  $\vartheta^0$  and  $(\xi_j^0, \eta_j^0)$  for  $j = 1, \dots, p$ .

We now discuss how we generate the prior covariance matrix  $\Sigma^{\vartheta,0}$ . For some previous work, this matrix is generated by taking the diagonal matrix with the variance from the measurement noise. This means we begin with independent beliefs. However, for this problem, the weight accessibility coefficients have natural proximity correlations. As can be seen from Figure 2 (a), the value of the accessibility coefficients are quite close locally. In fact, if we plot the sample autocorrelation function as shown in Figure 2(b), we can see that the correlation is 0.4718 when the lag is 1, jumps in the interval [-0.1, 0.2] for lag until 100 and almost decays to 0 after 250. If we fit an exponential function  $y = e^{-\kappa x}$  to the sample autocorrelation, the best fitted decay rate by least squares is  $\kappa^* = 0.39728$ .

This proximity correlation detection is also consistent with the domain experts' experience that the probes tend to perform similarly within a window of up to  $\sim 40$ . Theoretically, the larger this window is, the less measurements we need to identify the accessibility patterns of the target RNA. This is because the advantage of our algorithm is to incorporate the covariance matrix  $\Sigma^\vartheta$ , so we can locally infer more information based on what we have learned. Taking advantage of this proximity correlation knowledge, we use the exponential covariance function to model the prior covariance  $\Sigma^{\vartheta,0}$ . Under the exponential covariance functions, for any two points  $i$  and  $j$  from  $1, \dots, p$ ,

$$\text{Corr}(\vartheta_i, \vartheta_j) = \exp\{-\gamma|i - j|\}, \quad (24)$$

$$\text{Var}(\vartheta_i) = \beta_i^2, \quad (25)$$



**Figure 2** (a) Prior Accessibility Coefficients of the In vitro DMS footprinting Data; (b) The Sample Autocorrelation Function for Prior Accessibility Coefficients

*Note.* The fitted exponential decay function with a decay rate of 0.39728 is plotted in the blue dash line in (b).

where  $\gamma > 0$  and  $\beta_1, \dots, \beta_p > 0$  are hyperparameters chosen to reflect our belief. Here,  $\beta_i$  should be chosen to represent our confidence that  $\vartheta_i$  is close to our chosen mean function. The  $\gamma$  should reflect how quickly we believe  $\Sigma_{i,j}^{\vartheta}$  changes as  $i$  and  $j$  move further apart, with larger values of  $\gamma$  suggesting more rapid change. This simple family of covariance functions produces Gaussian process priors that are stationary and thus can be used for modeling the accessibility coefficients in this problem.

In practice, when one is unsure about the value of these hyperparameters, second-level priors can be put to model the coefficients  $\gamma$  and  $\beta_i$ . However, instead of using these hierarchical maximum a posteriori (MAP) approaches, we directly set up the values according to the prior in vitro DMS footprinting data and the domain experts' experience. Specifically, we let  $\gamma = \kappa^* = 0.39728$  from the fitted decay rate of the in vitro DMS footprinting data as shown in Figure 2(b). Also, according to our domain experts, the noise ratio for estimating the accessibility coefficient is  $10\% \sim 15\%$ , so we set the noise ratio  $r = 20\%$  to be conservative. That is

$$\beta_j = 20\% \times \tilde{\vartheta}_j, \text{ for } j = 1, \dots, p, \quad (26)$$

where

$$\tilde{\vartheta}_j := \begin{cases} \vartheta_j & \text{for } j : \vartheta_j \neq 0 \\ \sum_{j=1}^p \vartheta_j / p & \text{for } j : \vartheta_j = 0, \end{cases} \quad (27)$$

Combining (24), (25), and (26), we set the prior covariance matrix  $\Sigma^{\vartheta,0}$  as

$$\Sigma_{i,j}^{\vartheta,0} = r^2 \tilde{\vartheta}_i \tilde{\vartheta}_j \exp\{-\kappa^*|i-j|\}, \quad \text{for } i, j = 1, \dots, p, \quad (28)$$

where  $r = 20\%$ ,  $\kappa^* = 0.39728$ , and  $\tilde{\vartheta}_1, \dots, \tilde{\vartheta}_p$  are from (27).

Besides the prior covariance matrix, we also have to set the shape parameters  $(\xi_j^0, \eta_j^0)$  for the beta distribution (the frequency priors) in (7). For  $j = 1, \dots, p$ , we propose to set the frequency priors as

$$(\xi_j, \eta_j) = \begin{cases} (1, 1) + (w, 0), & \text{for } j : \vartheta_j \neq 0 \\ (1, 1) + (0, w), & \text{for } j : \vartheta_j = 0 \end{cases}.$$

Here  $w \geq 0$  is a hyperparameter representing our confidence in the prior sparsity pattern. A smaller  $w$  reflects less confidence in the prior while a larger  $w$  represents more confidence. If at the end of the experiments our algorithm uses probability 0.5 as a threshold to choose the nonzero coefficients, then  $w$  should not be larger than the sampling budget  $N$ . Otherwise, the sparsity pattern of the posterior estimate is totally identical with the prior data no matter what Lasso estimates we get. In the following simulations, we treat  $w$  as a tunable parameter depending on either the good prior or the bad prior cases.

## 4.2. Simulation Results

Notice that for this RNA accessibility identification problem, the real accessibility profile is unknown, while we can approximately learn this through various experimental methods. In this paper, we perform various simulations in which we sample a truth from a stochastic process and then run this trial with this fixed truth for some fixed number of measurement budget  $N$ . Then we replicate this over several runs to assess the performance of various policies. This truth coefficient is usually sampled through both vertically perturbing the values of the prior coefficient by a normally distributed random variable, and horizontally rotating the prior along the RNA molecule. In this section, we show the simulation results for all of the algorithms demonstrated in Section 3.

**4.2.1. Results using the SpKG Algorithm** We begin by describing the results for the SpKG policy presented in Algorithm 2. In this simulation, as suggested by the scientists on the team, we try all the probes of length 10 with 3 overlaps for the adjacent ones. Then we have  $M = 55$  number of alternatives. In this setting, we compare Algorithm 2 with two

other policies: a pure exploration policy, which randomly chooses an alternative to test at each time, and Algorithm 1, which uses KG with a nonsparse, linear belief model. It is worth emphasizing that for the pure exploration policy here, although it does not use the sparse linear structure to make measurement decisions, it still updates the belief in the same way as that of SpKG.

In this simulation, we generate the true accessibility coefficient vector from a multivariate normal distribution with mean  $\boldsymbol{\vartheta}^0$ , with the covariance matrix being the same form as (28). The differences are the noise ratio  $r$  is chosen to be as large as 10, and  $\kappa$  is drawn from a normal distribution with mean  $\kappa^*$  and standard deviation 0.1. Then we take this vector and right circularly shift it by an integer value uniformly sampled from 20 to 50 at each time. Since the prior has now been significantly altered from the truth, we believe it is not a good prior and set  $w = 10$  with the measurement budget  $N = 100$ . To quantitatively measure the performance of different policies, we consider the opportunity cost (OC), defined as the difference between the true value of the alternative that is actually the best and the true value of the alternative that is the best according to the policy's posterior belief distribution, i.e.,

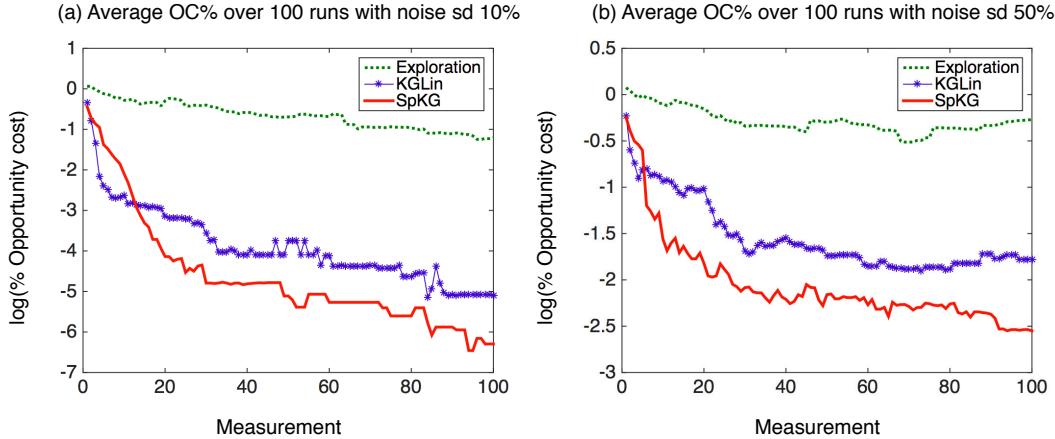
$$\text{OC}(n) = \mu(x^*) - \mu(x^{n,*}),$$

where  $x^*$  is the true optimal alternative, and  $x^{n,*}$  is the estimated optimal alternative at time  $n$ . So OC describes how far from optimal the current estimate of the optimal solution is after each experiment and thus can serve as a metric for the performance of a specific decision policy. For illustrative purposes, we also consider the percentage OC with respect to the optimal value,

$$\text{OC\%}(n) = \frac{\mu(x^*) - \mu(x^{n,*})}{\mu(x^*)}.$$

This normalized representation is unit-free and better illustrates how far in percentage we are from the optimal.

Figure 3 plots the average percentage OC on a log scale over 100 runs for three different policies in both low and high noise settings. The standard deviations of noise are 10% and 50% of the expected range of the truth. As both figures show, compared with pure exploration, SpKG results in a significant reduction in the opportunity cost, although the exploration policy also takes advantage of the sparse linear structure for the Bayesian



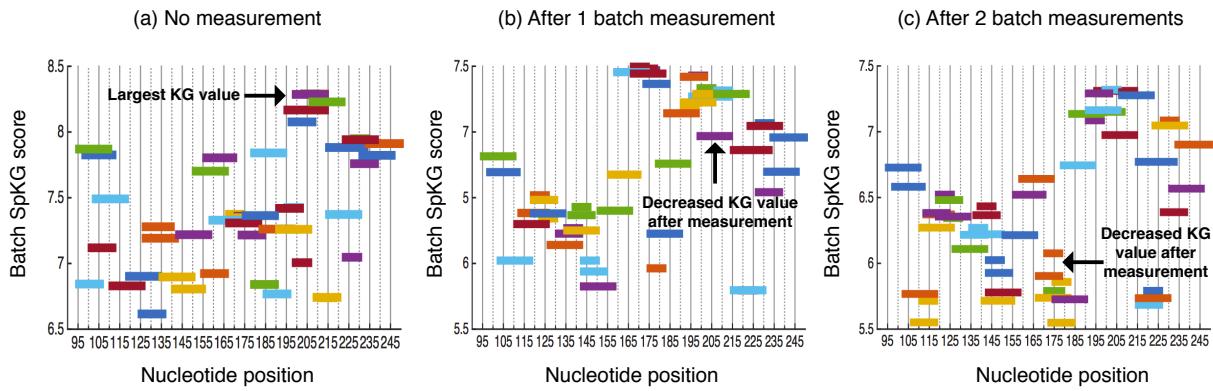
**Figure 3 Average Opportunity Cost Comparing Policies with Low and High Measurement Noise**

*Notes.* This simulation is for the whole target molecule sequence. The alternative probes are of length 10 with 3 overlaps for the adjacent ones. To better visualize the difference in OC, the average percentage OC is plotted on a log scale over 100 simulation trials.

implementation. When comparing with KG for a nonsparse, linear belief model, SpKG outperforms with lower average opportunity costs most of the time. However, during the initial stage when there are less than five measurements, SpKG behaves no better than KGLin, especially in low noise settings. This is because it takes several samples for Lasso to identify the true support. When it finds the sparsity pattern, SpKG is able to find alternatives converging to the truth much more efficiently than other policies.

**4.2.2. Results using the Batch SpKG Algorithm** In this simulation, we try testing the batch SpKG algorithm described in Algorithm 5. For the real experiments, the experimentalist is able to synthesize three probes to test the fluorescence intensities in parallel at each time. So in the batch setting, we let  $B = 3$ . From now on, let us take a specific region of the RNA molecule from site 95 to site 251, as this region is suggested by the domain expert to be the most promising sub-sequence. We try a larger set of probes than before: all 8-long probes shifted by 4, all 12-long probes shifted by 6, and all 16-long probes shifted by 8. That is we take all the regions  $[4k + 1, 4k + 8]$ ,  $[6k + 1, 6k + 12]$ , and  $[8k + 1, 8k + 16]$  starting from 95 and ending at 251. Beside these, we also include ten other probes as suggested by the domain experts:  $[98, 112]$ ,  $[113, 126]$ ,  $[127, 140]$ ,  $[141, 155]$ ,  $[156, 170]$ ,  $[171, 179]$ ,  $[179, 194]$ ,  $[195, 214]$ ,  $[215, 233]$ , and  $[234, 251]$ . In total, we have  $M = 91$  number of alternatives. We generate the true accessibility profile in the same way as before.

First, we illustrate how the batch SpKG policy works under a measurement noise of 30%. For one such simulated truth, we depict the batch SpKG value initially, after one, and two batch measurements, respectively in Figure 4. For these figures, we only include those probes with batch SpKG values above the mean to better visualize the KG scores. As indicated by the arrows, for the probes with the largest batch SpKG scores, the KG scores drop after they have been measured. As we only plot those with KG scores above average, some probes with high KG scores in Figure 4(a) have the scores dropped below average after being measured and are therefore not shown in Figure 4(b). This observation is also consistent with our intuition of SpKG as a measure of the value of information, and thus we can use this policy as a guideline to pick the next experiments.

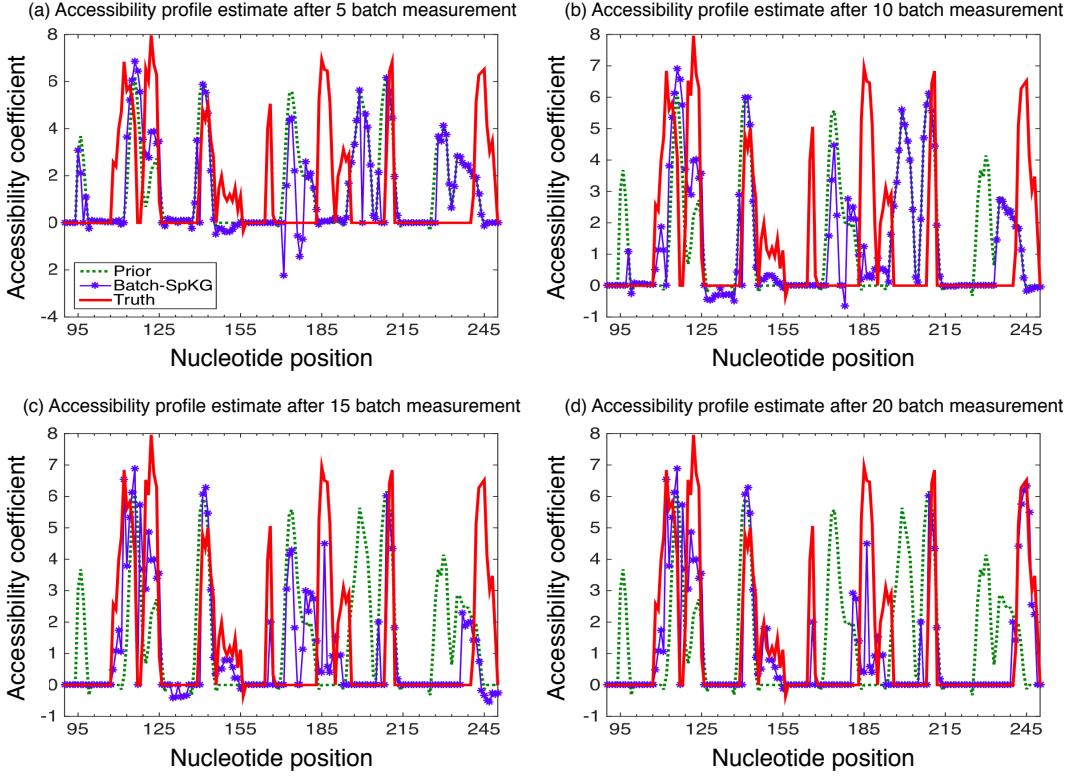


**Figure 4 Batch SpKG Values Before and After 1 and 2 Batch Measurements with Noise Ratio of 30%.**

*Notes.* This simulation is for a selected set of probes ranging from site 95 to site 251. Each bar is a potential range of a probe. The vertical axis is the batch SpKG score. Only those probes with the batch SpKG score above the mean are plotted. The arrows indicate the decreases in KG values for the probes that were previously measured. Note that some of these are not shown since they have KG values below average.

Furthermore, for one simulated truth, we also plot the estimates of accessibility profiles (coefficients) after 5, 10, 15, and 20 batch measurements with a noise ratio of 20% in Figure 5.

As one can see from Figure 5, after five batch measurements, the estimate is still closer to the prior than the truth. After 10 batch measurements, we have discovered many of the accessible regions. After 15 batch measurements, we have not only discovered the location of the accessible regions, but also obtained good estimates for the actual accessibility value.



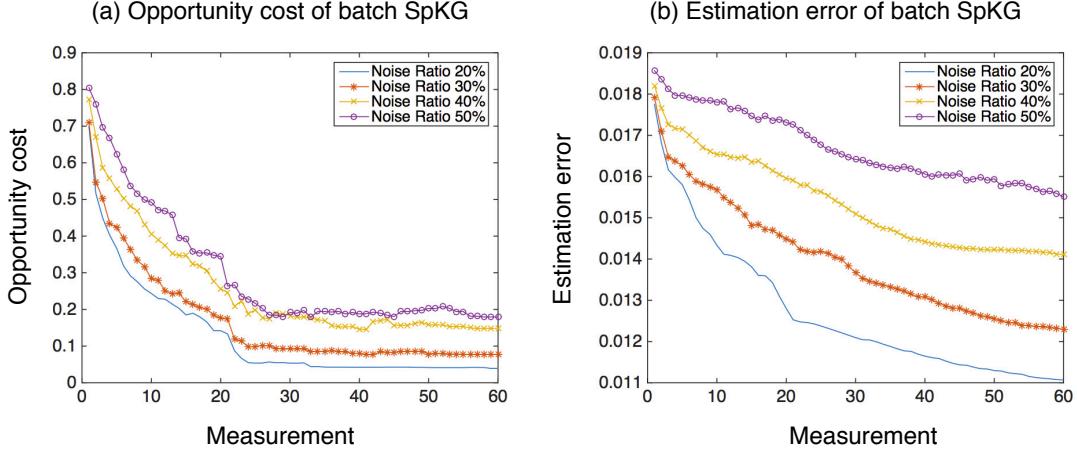
**Figure 5 Accessibility Profile Estimate by the Batch SpKG Algorithm After 5, 10, 15, and 20 Batch Measurements with Noise Ratio of 20%**

*Note.* This is for the accessibility coefficient ranging from site 95 to site 251.

And after 20 batch measurements, our estimate closely matches the truth. At last, we also try different noise levels 20%, 30%, 40%, and 50% and repeatedly run such simulations for 200 times for each level. The averaged percentage OC and estimation error are plotted in Figure 6. Here the normalized estimation error is the  $\ell_2$  distance between the estimated coefficient and the truth divided by the length of the coefficient vector, which is 157 currently.

Experimentally, fluorescent measurements are made by performing induction assays on prepared cell cultures. For each cell culture prepared, several samples are obtained, and fluorescence measurements are made via flow cytometry. Measurement dispersity in a small number of samples can be as large as 15% ~ 20% in standard deviation. For this noise level, we can see from the figure that most locations of the highly accessible regions can be found after about 25 observations. However, note that the true accessibility profiles sampled for the simulations are perturbed by a large amount. In reality, we suspect the truth to be

more in agreement with the prior footprinting data, which implies better performance by the SpKG algorithm.



**Figure 6 (a) Averaged Percentage Opportunity Cost of Batch SpKG with Various Noise Ratios; (b) Averaged Normalized  $\ell_2$  Estimation Error of Accessibility Coefficient with Various Noise Ratios**

*Note.* Both figures are for replicated runs averaged over 200 times.

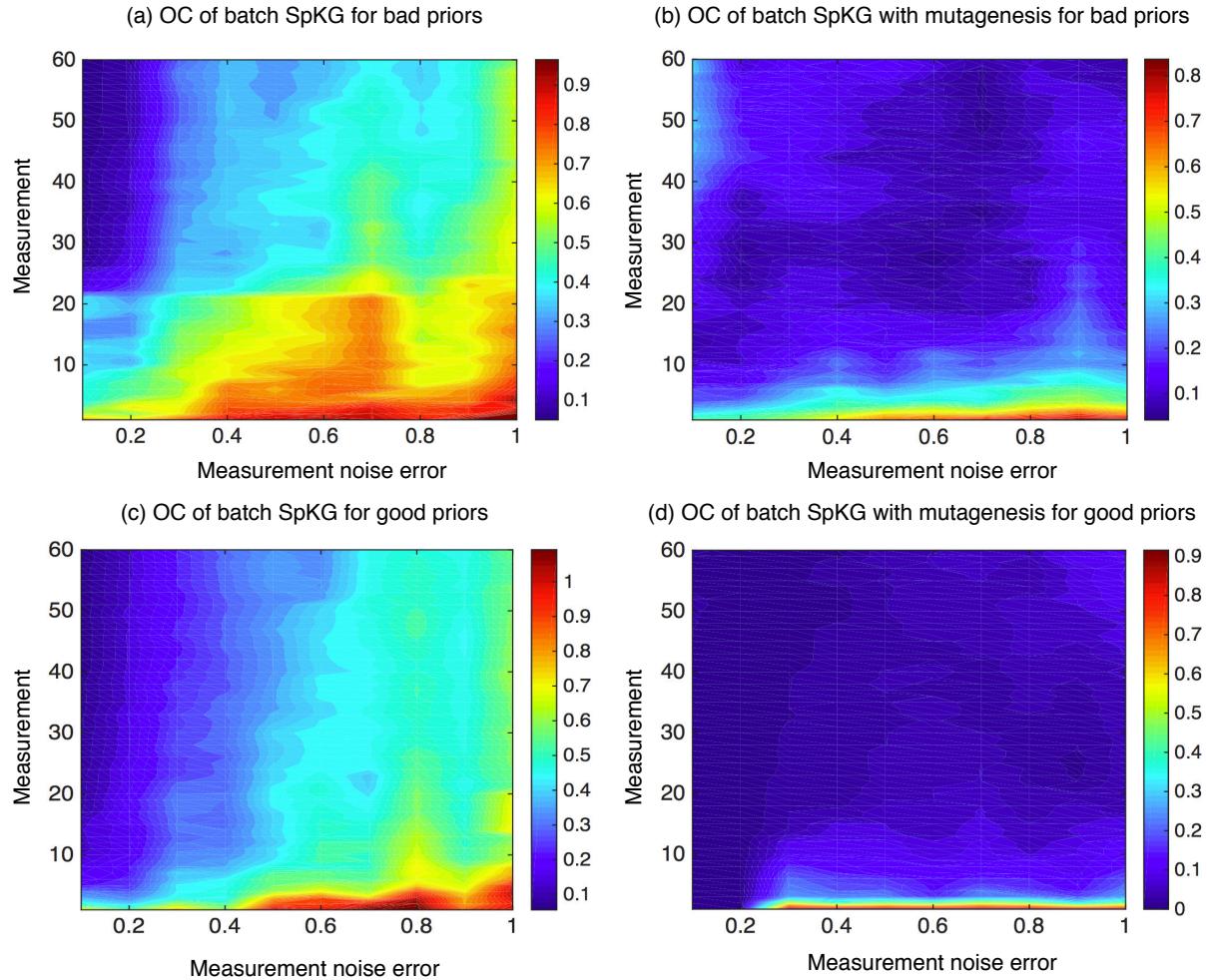
**4.2.3. Results using the Batch SpKG Algorithm with Length Mutagenesis** In this set of experiments, we use the same set of probes in Section 4.2.2 as the initial alternative library. The simulations are run with two different priors: a good prior and a bad prior. The bad prior is the one used in the above experiments, which is obtained by doing both vertical perturbation and horizontal shift from the in vitro DMS footprinting data. For the good prior, we only do vertical perturbation with the same amount, so we would think of the sparsity pattern of the good prior more proximal to the truth. Therefore, for this set of simulations with  $L = 20$ ,  $B = 3$ , we set  $w = 10$  for the bad prior and  $w = 20$  for the good prior.

Figure 7 compares the batch SpKG algorithm with and without mutagenesis for both good and bad priors averaged over 300 simulation trials. It shows the mean percentage opportunity cost as a function of measurements and measurement noise errors. We try ten different noise levels from 10% to 100%. For all of the figures, we observe that the OC% decays as the number of measurements increases and the measurement noise decreases. Such plots can be useful in experimental budgeting and show the required number of measurements needed to obtain a certain level of optimality for some noise level.

Comparing Figure 7(a)(b) with (c)(d), we can see that the OC% decreases much faster in the good prior cases as a function of the number of experiments, as expected. Comparing Figure 7 (a)(c) with (b)(d), we find that the OC% generated from the mutagenesis procedures tends to be smaller than those without mutagenesis. This is because when we add a new probe with the largest KG score into the alternative library with mutagenesis, we often add the one with a higher fluorescence level than the current fluorescences. In other words, the true highest fluorescence level is increasing as a new probe is added into the library. In such cases, it would be more intuitive to see how the highest fluorescence is varying over time. Figure 8 provides a more illustrative explanation for how the actual highest fluorescence changes. For this set of figures, we compare how the actual fluorescence changes over measurement. The red star line is the true highest fluorescence, and the blue solid line is the value of true fluorescence by estimation. So the difference between the two lines is the OC. We compare three different policies: batch SpKG, batch SpKG with mutagenesis, and exploration mutagenesis. Exploration mutagenesis involves randomly adding new probes if they are not in the current library. From Figure 8, we can see that SpKG with mutagenesis has the ability to find new probes with fluorescence values about three to five times the highest values in the initial set. However, for exploration mutagenesis, the highest fluorescence improves less, as expected. This also proves the power of the KG policy to identify the potential alternative that can outperform the current optimal one. Furthermore, it is also worth noting that with mutagenesis, the value of the true fluorescence through the Bayesian estimation is pretty close to the truth. That means the OC is close to zero, which is consistent with Figure 7(b)(d).

## 5. Conclusion

Identifying the accessibility pattern of an RNA molecule is an important topic in molecular biology. On one hand, the real experimental study is a long and expensive process for which adaptive learning procedures like the knowledge gradient policy are well suited to make experimental decisions. On the other hand, this problem naturally incorporates sparsity linear structure and thus requires more sophisticated statistical techniques to analyze the underlying model. To better help learn the accessibility profile of the RNA molecule, we use a recently derived SpKG policy (Li et al. 2015), which is a novel hybrid of Bayesian Ranking & Selection and frequentist  $\ell_1$  penalized regression approach called Lasso. This

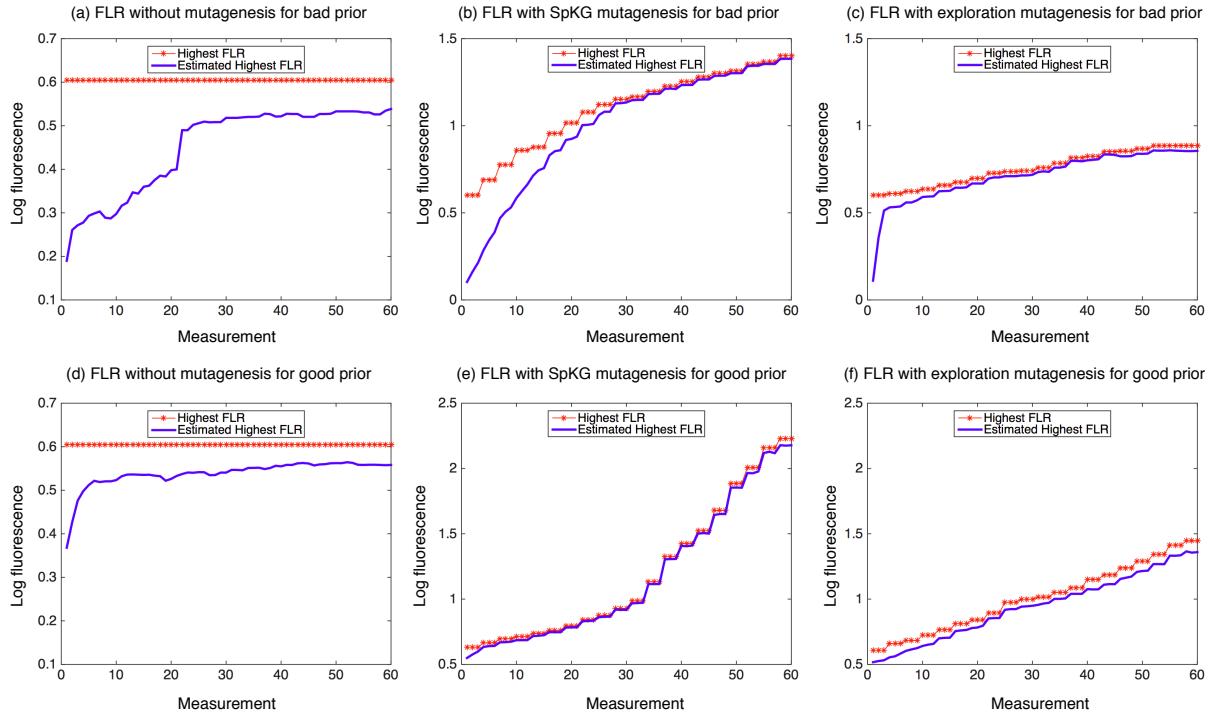


**Figure 7 Averaged Percentage Opportunity Cost of Batch SpKG with and without Mutagenesis for Good and Bad Priors over 300 Runs**

*Note.* The contour plots show averaged percentage OC as a function of measurements and measurement noise errors.

optimal learning algorithm has been shown to efficiently identify the accessibility pattern and learn the underlying sparsity structures. Controlled experiments also show that it outperforms several other policies.

Algorithmically, we also extend the SpKG policy into a general framework for batch mode learning, where the experimenter can run several experiments in batch. Empirical studies demonstrate the effectiveness of this policy for various experimental setups. Besides, we also derive the batch SpKG algorithm using length mutagenesis to expand the set of alternatives. In this procedure, the alternative library is adaptively enlarged as the most



**Figure 8 True Highest Log Fluorescence and Estimated Highest Fluorescence for Good and Bad Priors Comparing Three Policies: Batch SpKG, Batch SpKG with Mutagenesis, and Exploration Mutagenesis**

*Notes.* These plots show the values in true fluorescence the highest and the estimated highest. The difference between the two lines is the OC. The highest fluorescence for batch SpKG policy remains constant since we maintain the same probe alternative library throughout measurements. With mutagenesis, most of the time we could find the probe with higher fluorescence than before. For batch SpKG with mutagenesis, the new probe with the largest KG score is added. For exploration mutagenesis, the new probe is randomly added. The results are averaged over 300 runs.

promising alternative is added in at each time. Controlled experiments also demonstrate its efficiency in identifying the accessibility pattern of the RNA molecule.

In conclusion, it is worth noting that the SpKG algorithm has only been applied to sparse linear beliefs. Possible future directions of the work would include the study of more general nonlinear beliefs that incorporate sparsity structure. Despite this limitation, we still believe the SpKG algorithm would allow efficient implementation for large data sets, and we would like to suggest this algorithm for solving more general application problems with sparse linear beliefs.

## Acknowledgments

We thank Dr. Rick Russell for kindly providing the raw in vitro DMS footprinting data used in this work and previously published in Russell et al. (2006). We thank the Welch Foundation (Grant NO. F- 756); the Air Force Office of Scientific Research (AFOSR) Young Investigator program (FA9550-13-1-0160); and the Consejo Nacional de Ciencia y Tecnología for the graduate 445 fellowship granted to J.V.A (CONACYT-94638). This research was supported in part by AFOSR grant contract FA9550-12-1-0200 for Program on Optimization and Discrete Mathematics, with valuable support from the program on Natural Materials and Systems.

## References

- Auer, P., N. Cesa-Bianchi, P. Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* **47** 235–256.
- Bennett, C. F., E. E. Swayze. 2010. RNA targeting therapeutics: molecular mechanisms of antisense oligonucleotides as a therapeutic platform. *Annu. Rev. Pharmacol. Toxicol.* **50** 259–293.
- Bubeck, S., N. Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning* **5** 1–122.
- Bull, A. D. 2011. Convergence rates of efficient global optimization algorithms. *J. Mach. Learn. Res.* **12** 2879–2904.
- Cech, T. R., S. H. Damberger, R. R. Gutell. 1994. Representation of the secondary and tertiary structure of group I introns. *Nat. Struct. Mol. Biol.* **1** 273–280.
- Cech, T. R., A. J. Zaug, P. J. Grabowski. 1981. In vitro splicing of the ribosomal RNA precursor of Tetrahymena: involvement of a guanosine nucleotide in the excision of the intervening sequence. *Cell* **27** 487–496.
- Chan, J. H., S. Lim, W. S. Wong. 2006. Antisense oligonucleotides: from design to therapeutic application. *Clin. Exp. pharmacol. physiol.* **33** 533–540.
- Chen, C. 2010. *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*, vol. 1. World Scientific.
- Chen, X., Q. Lin, J. Pena. 2012. Optimal regularized dual averaging methods for stochastic optimization. *NIPS*, vol. 25. 404–412.
- Chen, Y., A. O. Hero. 2012. Recursive group Lasso. *IEEE Trans Sig. Process.* **60** 3978–3987.
- DeVos, S. L., T. M. Miller. 2013. Antisense oligonucleotides: Treating neurodegeneration at the level of RNA. *Neurotherapeutics* **10** 486–497.
- Frazier, P. I., W. B. Powell, S. Dayanik. 2008. A knowledge-gradient policy for sequential information collection. *SIAM J. Control Optim.* **47** 2410–2439.
- Frazier, P. I., W. B. Powell, S. Dayanik. 2009. The knowledge-gradient policy for correlated normal beliefs. *INFORMS J. Comput.* **21** 599–613.

- Friedman, J., T. Hastie, R. Tibshirani. 2001. *The elements of statistical learning*, vol. 1. Springer series in statistics Springer, Berlin.
- Garrigues, P., L. El Ghaoui. 2008. An homotopy algorithm for the Lasso with online observations. *NIPS*. 489–496.
- Gelderman, G., L. M. Contreras. 2013. Discovery of post-transcriptional regulatory RNAs using next generation sequencing technologies. *Systems Metabolic Engineering*. Springer, 269–295.
- Gittins, J., K. Glazebrook, R. Weber. 2011. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons.
- Gold, L., B. Polisky, O. Uhlenbeck, M. Yarus. 1995. Diversity of oligonucleotide functions. *Annu. Rev. Biochem.* **64** 763–797.
- Golden, B. L., A. R. Gooding, E. R. Podell, T. R. Cech. 1998. A preorganized active site in the crystal structure of the Tetrahymena ribozyme. *Science* **282** 259–264.
- Gupta, S. S., K. J. Miescke. 1996. Bayesian look ahead one-stage sampling allocations for selection of the best population. *J. Stat. Plan. Inference* **54** 229–244.
- Haning, K., S.H. Cho, L.M. Contreras. 2015. Strain Engineering via regulatory noncoding RNA mechanisms: not a one-blueprint fits all. *Curr. Opin. Chem. Eng.* .
- Huang, D., T. T. Allen, W. I. Notz, N. Zeng. 2006. Global optimization of stochastic black-box systems via sequential kriging meta-models. *J. Global Optim.* **34** 441–466.
- Jones, D. R., M. Schonlau, W. J. Welch. 1998. Efficient global optimization of expensive black-box functions. *J. Global Optim.* **13** 455–492.
- Kieft, J. S., I. Tinoco. 1997. Solution structure of a metal-binding site in the major groove of RNA complexed with cobalt (III) hexammine. *Structure* **5** 713–721.
- Koduvayur, S. P., S. A. Woodson. 2004. Intracellular folding of the Tetrahymena group I intron depends on exon sequence and promoter choice. *RNA* **10** 1526–1532.
- Kruger, K., P. J. Grabowski, A. J. Zaugg, J. Sands, D. E. Gottschling, T. R. Cech. 1982. Self-splicing RNA: autoexcision and autocyclization of the ribosomal RNA intervening sequence of Tetrahymena. *Cell* **31** 147–157.
- Li, Y., H. Liu, W. B. Powell. 2015. The knowledge gradient policy using a sparse additive belief model. *arXiv:1503.05567* .
- Mückstein, U., H. Tafer, J. Hackermüller, S. H. Bernhart, P. F. Stadler, I. L. Hofacker. 2006. Thermodynamics of RNA–RNA binding. *Bioinformatics* **22** 1177–1182.
- Pain, A., A. Ott, H. Amine, T. Rochat, P. Bouloc, D. Gautheret. 2015. An assessment of bacterial small RNA target prediction programs. *RNA Biol.* **12** 509–513.
- Powell, W. B., I. O. Ryzhov. 2012. *Optimal Learning*. John Wiley and Sons, Hoboken, NJ.

- Rodrigo, G., T. E. Landrain, E. Majer, J.A. Daros, A. Jaramillo. 2013. Full design automation of multi-state RNA devices to program gene expression using energy-based optimization. *PLoS Comput. Biol.* **9** e1003172.
- Russell, R., R. Das, H. Suh, K. J. Travers, A. Laederach, M. A. Engelhardt, D. Herschlag. 2006. The paradoxical behavior of a highly structured misfolded intermediate in RNA folding. *J. Mol. Biol.* **363** 531–544.
- Russell, R., X. Zhuang, H. P. Babcock, I. S. Millett, S. Doniach, S. Chu, D. Herschlag. 2002. Exploring the folding landscape of a structured RNA. *Proc. Natl. Acad. Sci. U.S.A.* **99** 155–160.
- Ryzhov, I. O., W. B. Powell, P. I. Frazier. 2012. The knowledge gradient algorithm for a general class of online learning problems. *Oper. Res.* **60** 180–195.
- SantaLucia, J. 1998. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc. Natl. Acad. Sci. U.S.A.* **95** 1460–1465.
- Scherr, M., J. J. Rossi, G. Sczakiel, V. Patzel. 2000. RNA accessibility prediction: a theoretical approach is consistent with experimental studies in cell extracts. *Nucleic Acids Res.* **28** 2455–2461.
- Sowa, S. W., J. Vazquez-Anderson, C. A. Clark, R. De La Peña, K. Dunn, E. K. Fung, M. J Khouri, L. M. Contreras. 2015. Exploiting post-transcriptional regulation to probe RNA structures in vivo via fluorescence. *Nucleic Acids Res.* **43** e13–e13.
- Swisher, J. R., S. H. Jacobson, E. Yücesan. 2003. Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey. *ACM Trans. Model. Comput. Simul.* **13**.
- Tibshirani, R. 1996. Regression shrinkage and selection via the Lasso. *J. Roy. Statist. Soc. Ser. B-Statist. Methodology* 267–288.
- Vazquez-Anderson, J., L. M. Contreras. 2013. Regulatory RNAs: charming gene management styles for synthetic biology applications. *RNA Biol.* **10** 1778–1797.
- Wan, Y., H. Suh, R. Russell, D. Herschlag. 2010. Multiple unfolding events during native folding of the Tetrahymena group I ribozyme. *J. Mol. Biol.* **400** 1067–1077.
- Wang, Y., K. G. Reyes, K. A. Brown, C. A. Mirkin, W. B. Powell. 2015. Nested-batch-mode learning and stochastic optimization with an application to sequential multistage testing in materials science. *SIAM J. Sci. Comput.* **37** B361–B381.
- Xia, T., J. SantaLucia, M. E. Burkard, R. Kierzek, S. J. Schroeder, X. Jiao, C. Cox, D. H. Turner. 1998. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry* **37** 14719–14735.