

Automated experiment design for drug development

Master of Science Thesis in Computer Science - Algorithms, Languages and Logic

HANNES ERIKSSON

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Automated experiment design for drug development

Hannes Eriksson

© Hannes Eriksson, June 2015.

Examiner: Graham Kemp

Supervisors: Christos Dimitrakakis

Lars Carlsson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

A compound transformed into high dimensional space represented in a Gaussian process, see page 10.

Department of Computer Science and Engineering
Göteborg, Sweden June 2015

Abstract

Testing drugs in discovery is time consuming and expensive. An idea is then to eliminate unpromising compounds from the testing phase by using online learning methods to predict properties of yet to be tested compounds and determining which drugs to test. This is done by comparing substructures in the graph representation of compounds, transformed into a compressed high dimensional space where a Gaussian process bandit and a linear bandit is used to predict properties of new compounds. Results show that the bandits perform significantly better than random selection and that the feature compression probably does not decrease the overall accuracy of the predictions.

Keywords

Contextual bandits, Gaussian process bandit, experimental design, linear bandit, signature descriptor, compressed sensing, online learning, reinforcement learning

Acknowledgements

I would like to give special thanks to my supervisors Christos Dimitrakakis (Chalmers) and Lars Carlsson (AstraZeneca) for their help with this master thesis.

Glossary

IC₅₀ A metric for the potency of a drug which indicates how much of the drug is needed to inhibit a process by half

UCB Upper confidence bound

R-group is a side chain or a subgraph of a compound

Scaffold A core substructure of a graph that has connecting R-groups

Contents

1	Introduction	4
1.1	Aims	4
1.2	Scope	4
1.3	Methodology	5
1.4	AstraZeneca	5
1.5	Outline	6
1.6	Related work	6
2	Background	8
2.1	Biochemistry terminology	8
2.2	Bayesian probability	8
2.3	Gaussian process	9
2.3.1	Definition	9
2.3.2	Motivation and application	9
2.4	Signature descriptors	10
2.4.1	Definition	10
2.4.2	Motivation and application	11
2.5	Compressed sensing	13
2.5.1	Definition	13
2.5.2	Motivation and application	13
2.6	Bandits	13
2.6.1	Definition	14
2.6.2	Motivation and application	14
2.7	Selection algorithms	15
2.7.1	Linear Bandit selection using Thompson Sampling	15
2.7.2	Gaussian Process Bandit selection	16
3	Experiment	19
3.1	Details	19
3.2	Initial setting	20

3.2.1	Data sets	20
3.3	Randomized setting	22
3.3.1	Generation of data	22
3.3.2	Details	23
4	Results	24
4.1	Initial setting	24
4.1.1	CDK5	24
4.1.2	GNRHR	24
4.1.3	MAPK14	24
4.1.4	MGLL	25
4.2	Randomized setting	25
5	Discussion	32
5.1	Results	32
5.1.1	Initial setting	32
5.1.2	Randomized setting	32
5.2	Compressed sensing	33
5.2.1	Gain for GP-UCB	33
5.2.2	Gain for LinearBandit-TS	34
5.3	Exploitation vs. exploration with δ	34
5.4	Selection budget	34
5.5	GP-UCB vs. LinearBandit-TS	34
5.5.1	LinearBandit-TS	35
5.5.2	GP-UCB	35
5.6	General discussion	35
6	Conclusion	37
6.1	Results	37
6.2	GP-UCB or LinearBandit-TS?	37
6.3	Concluding remarks	38
6.4	Future work	38
	References	43

1

Introduction

Synthesizing and testing compounds for their potential use as drugs is time consuming and expensive [1]. An idea came about to alleviate this by restricting the amount of compounds sent for testing. The way to decide which compounds to continue research on would be by using Artificial Intelligence (AI) methods to predict properties of yet to be tested compounds and to select which compounds to test. By doing this, if the predictions are accurate enough, some unpromising compounds could perhaps be ruled out even before they have gone through a real testing phase. A breakthrough here would be interesting for the biopharmaceutical industry as it could decrease the costs since not as many compounds have to be synthesized and tested.

1.1 Aims

The aims of this Master Thesis are to investigate algorithms for predictions on data in high dimensional space and to develop methods for automated experiment design for drug development such that when given a set of compounds they should together with experimentalists rule out unpromising compounds. This by having the methods successively predict properties (toxicity, target proteins and so on) of yet to be tested compounds and then conducting tests on the compounds with the most promising properties. These properties will then be verified by the experimentalists who will feed the tested properties back to the methods for more accurate predictions on future compounds.

1.2 Scope

Signature descriptors are a way of identifying similarities in the graphical model of compounds by looking at which atoms are present, what kind of bonds connect the atoms and so on. This is used to predict properties of yet to be tested compounds. This work will only consist of using signature descriptors to compare the compounds.

One problem with this is that a compound in reality is a 3D-structure and in this work compounds are modeled as 2D-structures. A consequence of this is that compounds with different 3D-structure and properties might be represented as the same 2D-structure with different properties.

Global graph kernels that work on a whole graph unlike signature descriptors can find even more information hidden in a graph as shown by Johansson et al. [2]. These will however not be investigated because of time constraints but could be interesting to compare against the combination of signature descriptors together with bandits as used in this work.

1.3 Methodology

The work done in this thesis consisted of three phases, namely a literature study phase, a data acquisition and pre-processing phase, and finally the implementation phase. These phases were mostly run in parallel of each other.

In the literature study part the first goal was to read up on and become acquainted with the following things: graph kernel methods to be able to give a metric on the similarity or dissimilarity of different compounds, contextual bandits for selection of the most promising compounds and Gaussian processes for the online learning and prediction. Techniques for dimensionality reduction were also investigated because of the high dimensionality of the data, covering techniques such as compressed sensing.

The molecular data and their corresponding properties used in this report were taken from literature [3]. This data is organized into several data sets where each of the compounds in a particular data set was considered for the development of a particular drug. Each compound has its corresponding IC_{50} value for a distinct process that the drug in development is intended to affect.

The implementation phase consisted of both implementing the various algorithms and a testing stage where the algorithms were compared with each other, this to decide on which methods were best used to tackle the goal of the project.

1.4 AstraZeneca

AstraZeneca is a biopharmaceutical company whose work spans all the way from medicine research and medicine development to the commercialization of medicines [4]. During the development of new drugs these drugs normally have to undergo a testing phase to determine which drugs have the most promising properties. It is in this phase where the work conducted in this thesis is interesting. Instead of having to test the whole batch of drugs to find the most promising ones we test a part of the batch and hope to identify the best drugs in the batch without testing all of them.

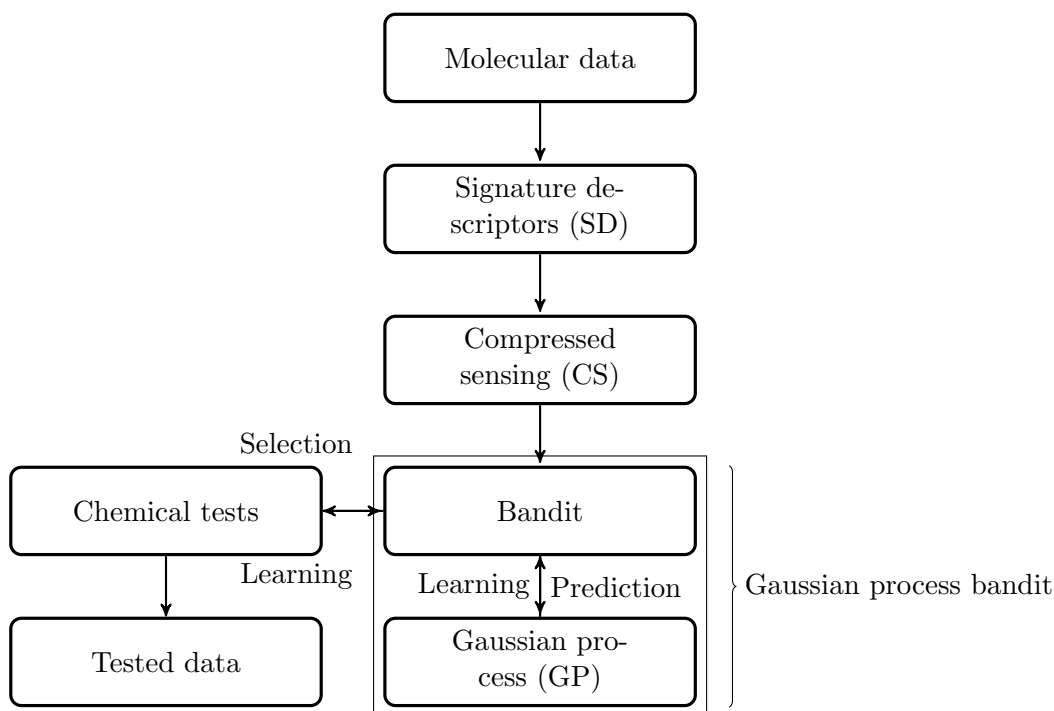


Figure 1.1: Project outline and work outline.

1.5 Outline

The Figure 1.1 depicts the information flow from molecular structures mapped onto some hyperspace to what the algorithm deems are the most promising compounds out of the data set.

In chapter 2 we cover the theoretical part of the different techniques used throughout this work. In chapter 3 we describe the experiment settings in which these techniques are used. In section 3.2 we describe the setting of an experiment with set data. In section 3.3 we describe the setting of an experiment on generated data. In chapter 4 we show the results of running these experiments. In chapter 5 we discuss these results. In chapter 6 we give our concluding remarks on this work as well as possible improvements on our solutions and other ideas to explore that did not fit in this work.

1.6 Related work

The paper by King et al. [5] describes a robot scientist that is capable of reasoning about data, deciding on experiments to run to test some hypothesis, physically conducting the experiments and interpreting the results of the experiments. In that work the robot is able to do all that without the aid of a person. This could be likened to the work done in this thesis since a robot could probably be used to conduct the chemical experiments and have the results fed back into the system used here. Note however that in this work

human intelligence is used to both test and interpret properties of the compounds.

The work done in Srinivas et al. [6] uses the very same idea for bandit selection as in this work (GP-UCB) and they show regret bounds for different kernels and compare the algorithm with heuristic methods such as Expected Improvement and Most Probable Improvement. They then use these methods to try to find the most congested part of a highway.

Krause and Ong [7] gave multi-task experiment design as an example application of their CGP-UCB algorithm where the idea was to design a vaccine through a sequence of experiments. In that paper they wanted to identify the maximally binding peptides for complex compounds. The similarity measure used for the context is the Hamming distance between amino acids in the peptide bindings as opposed to the signature descriptors used in this work and they focus on a select few compounds that the peptides are supposed to bind.

In a recent paper by Williams et al. [1] one of the things the authors do is employing AI methods in the drug discovery process to discover new compounds to combat tropical diseases. They do this by using a Gaussian process with a linear kernel.

The contribution of this thesis to the field of experiment design is that we show that using a Gaussian process for prediction and Upper Confidence Bound for selection works well in the drug development setting. We compare that combination with another selection algorithm using a Linear bandit with Thompson sampling as well as with random selection. We show that using a dimensionality reduction on the data probably does not lower the overall accuracy of the selection and prediction.

2

Background

This chapter will serve as a baseline description for the different methods used throughout this project. Firstly, we will explain the basic terminology used in this work, then the methods of processing the data and finally the algorithms we used to control the prediction and selection process of the drugs in development. The order of things in this chapter mainly follows the structure depicted in Figure 1.1.

2.1 Biochemistry terminology

Drugs in development are often assigned metrics that tell how potent the drugs are at inhibiting particular processes. One of these metrics is the IC_{50} metric. This IC_{50} value is defined by Soothill et al. [8] as a metric that denotes the concentration required of an inhibitor to reduce a response by half. This value is generally transformed into its pIC_{50} value in optimization problems where $pIC_{50} = -\log(IC_{50})$. A higher pIC_{50} value means greater potency.

A scaffold is a core part of a compound on which functional R-groups can be substituted or exchanged for as described by Chen et al. [9]. An example of a scaffold can be seen in Figure 2.1. In this case the scaffold is the carbon ring and the R_1, R_2, \dots, R_6 are the functional R-groups. These R-groups can be substituted by atoms, e.g. hydrogen, oxygen, fluorine or others, or by other subgraphs.

2.2 Bayesian probability

In Bayesian methods a belief about the correctness of each possible hypothesis is maintained in the form of a probability distribution as described by de Finetti [10]. This is opposed to frequentist probability where the probabilities are calculated from long-running averages over repeated experiments. The main advantage of Bayesian methods is that the probability computations require nothing more than the standard framework

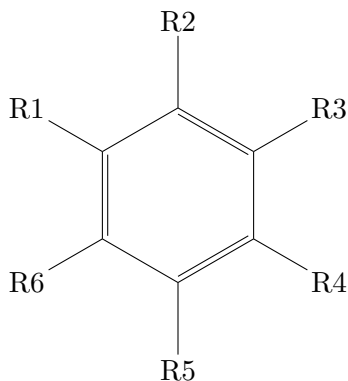


Figure 2.1: Graph of a scaffold with its corresponding R-groups.

of probability. All the methods used in this work that are probabilistic are based on Bayesian probability.

2.3 Gaussian process

A Gaussian process (GP) as defined by Rasmussen and Williams [11], is a generalization of the Gaussian probability distribution where instead of distributing over a finite number of variables the Gaussian process distributes over an infinite number of variables or functions.

2.3.1 Definition

Let $\mathbf{x} = (x_1, x_2, \dots, x_i) \in \mathcal{X}$ be a set of points in some input space. Let then $\mathbf{f} = f(\mathbf{x})$ where f is a function $f : \mathcal{X} \rightarrow \mathbb{R}$. $\mathcal{P}(f)$ is then a \mathcal{GP} if the marginal distribution over the set $\mathcal{P}(\mathbf{f})$ is distributed as a multivariate Gaussian.

A Gaussian process is defined by its mean function $m(\mathbf{x})$ and its covariance function $k(\mathbf{x}, \mathbf{x}')$. As shown in Rasmussen and Williams [11] this means the GP can be written as $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$. The mean function can often be taken as zero but the chosen covariance function or kernel is critical for the GP's ability to predict and compare points with each other. A commonly used kernel is the *Radial Basis Function* kernel as described in subsection 2.7.2.

2.3.2 Motivation and application

Figure 2.2 displays what a Gaussian process with its predicted mean and variance for points $[0, 2\pi]$ may look like. The shaded area represents the interval $[\mu_x - \sigma_x, \mu_x + \sigma_x]$ for the points on the x-axis. The variance is greater for points where the GP is less confident in its predictions, for example in areas that are quite different from the points the GP has learned so far. This data can then be used in selection algorithms like Algorithm 2.

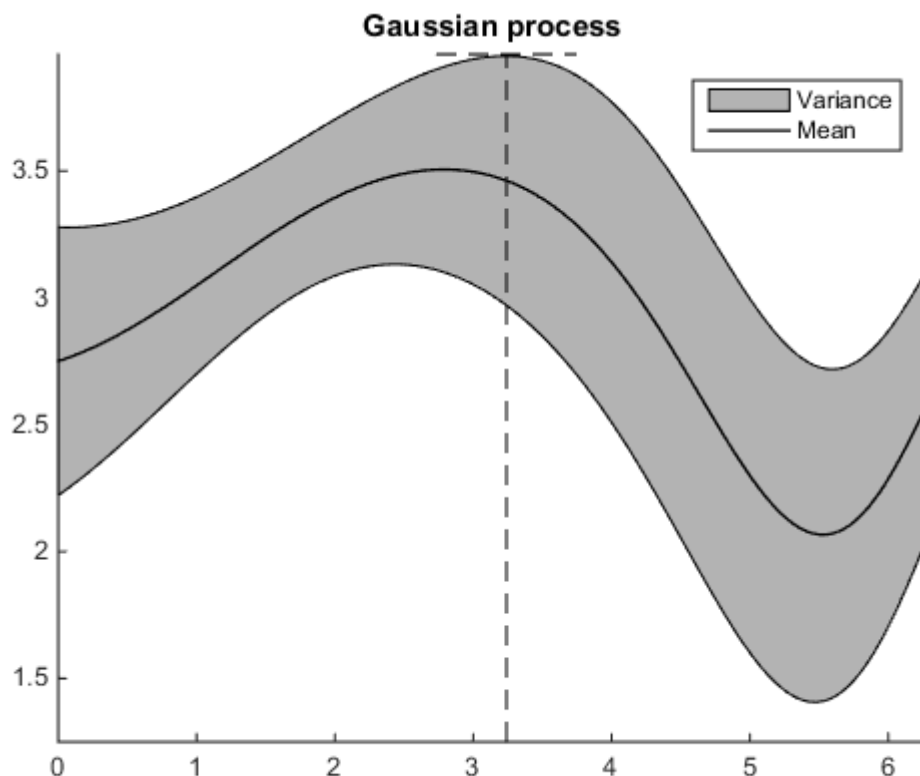


Figure 2.2: A Gaussian process with predicted mean and variance for points in $[0, 2\pi]$.

In general this x-axis is replaced by some multidimensional space as in this work where the points on the x-axis are bandits which will be explained further in section 2.6.

2.4 Signature descriptors

Signature descriptors can be used to identify substructures in a given graph. The definition of those and how they are used in this project is described below.

2.4.1 Definition

A molecular 2D-structure could be seen as an undirected graph G containing a set of vertices V and a set of edges E . These vertices can be atoms or more complicated chemical structures that make up a compound. The edges represent the molecular bonds present in the complete compound.

Faulon et al. [12] described a way to try to identify properties of a graph given its spatial structure using different operators on the graph itself. These operators are

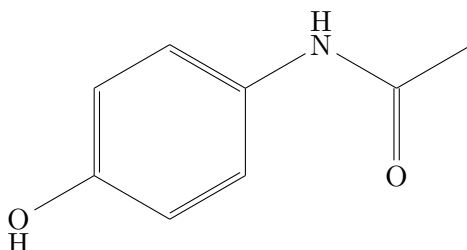


Figure 2.3: The paracetamol or acetaminophen compound modeled as a graph.

known as descriptors. The descriptor considered in this work identifies substructures of the molecular graph and will henceforth be referred to as *signature descriptors*.

A signature descriptor is defined as a subgraph $G_i^{(h)}$ of G where i is a vertex in G . This $G_i^{(h)}$ represents the set of paths from i given a particular height h . The height h is the maximum length of each path in $G_i^{(h)}$. These subgraphs can also be seen as trees where the root is the vertex i and the children of i are its neighbors and their connecting edges, then their children are their neighbors, and so on.

Basic chemistry explains that an atom may be connected to at most four other atoms so we can achieve an upper bound on the number of elements present in each of these trees. This bound is $O(4^h)$ for one tree since each vertex might be connected to four other vertices. We trivially attain the bound $O(|V|4^h)$ on the number of vertices or substructures in all of the signature descriptors of G .

Eklund et al. [13] then explained how to map each of these substructures present in the subgraphs $G_i^{(h)}$ of G to a finite alphabet Σ . This alphabet is also bounded by $O(4^h)$ as in the worst case each substructure is unique.

It is then possible to map a compound to a point in $\mathbb{N}^{|\Sigma|}$ where the number $n \in \mathbb{N}$ represents the number of times a particular substructure is repeated throughout the compound. If the alphabet Σ is set to be the same for a set of compounds then these compounds will all live in the same hyperspace where it is possible to use traditional similarity metrics to compare them.

2.4.2 Motivation and application

One motivation for using this is, as mentioned earlier, to identify similarities in compounds. If two compounds are composed of similar substructures, then they might have similar properties. Another reason is that all that is needed to construct the signature descriptors is the graph G and the height h .

Which height to use is an interesting problem in itself; if the height is too small, then two very differently structured compounds will appear similar while if the height is too great, then the number of substructures will explode and make the computations infeasible in reasonable time.

Two signature descriptors of the paracetamol compound, shown in Figure 2.3, can be seen in Figure 2.4, for the heights $h = 1$ and $h = 2$. Note that bonded hydrogen atoms are disregarded in this model.

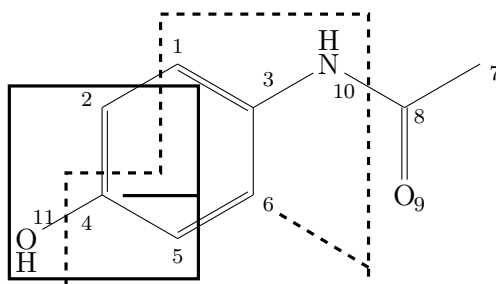


Figure 2.4: Paracetamol compound with its atoms numbered and with *some* of its signature descriptors shown for $height = 1, n = 4$ (—) and $height = 2, n = 6$ (- - -).

n	$h = 0$	$h = 1$	$h = 2$
1	C	$C(C, = C)$	$C(C(= C), = C(N, C))$
2	C	$C(C, = C)$	$C(C(= C), = C(O, C))$
3	C	$C(C, N, = C)$	$C(C(= C), N(C), = C(C))$
4	C	$C(C, O, = C)$	$C(C(= C), O, = C(C))$
5	C	$C(C, = C)$	$C(C(O, = C), = C(C))$
6	C	$C(C, = C)$	$C(C(N, = C), = C(C))$
7	C	$C(C)$	$C(C(= O), N)$
8	C	$C(C, N, = O)$	$C(C, N(C), = O)$
9	O	$O(= C)$	$O(= C(C, N))$
10	N	$N(C, C)$	$N(C(= O, C), C(= C, C))$
11	O	$O(C)$	$O(C(= C, C))$

Table 2.1: Table of signature descriptors of the paracetamol compound. Single bonds are left implicit and hydrogen atoms are ignored.

The Table 2.1, adapted from [13], shows all the subgraphs originating from each of the vertices for the heights $h = 1, 2, 3$. The subgraph for $n = 3, h = 1$ is read as following for instance: a carbon atom that is singly bonded to another carbon atom, singly bonded to a nitrogen atom and doubly bonded to another carbon atom. This follows a recursive pattern.

2.5 Compressed sensing

Compressed sensing is a way to recover sparse or compressible signals from approximations on a few elements relative to the size of the original signal. It is also possible to go the other way and compress sparse signals using this technique and that is the interesting part for this work. Below follows the definition of this and how it is used in this work.

2.5.1 Definition

Given a sparse structure of data \mathbf{x} the question is then if it is possible to compress this data to a much smaller data set \mathbf{y} such that the elements in \mathbf{y} preserve the information contained in \mathbf{x} with high probability. Baraniuk et al. [14] argues that this is the case given that the signal has the *restricted isometry property*. The idea is then to construct a matrix Φ such that $\mathbf{y} = \Phi\mathbf{x}$.

Let the density, i.e. the number of non-zero elements, of the data set \mathbf{x} be S . Let the dimensionality of the data set \mathbf{x} be N . Let $K = O(S \log \frac{N}{S})$.

Candés [15] showed that it is then possible to generate Φ with dimensions $N \times K$ using a Gaussian matrix consisting of values $o_{i,j} \sim \mathcal{N}(0, \frac{1}{S})$.

$$\begin{pmatrix} y_1 & y_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}_{K \times 1} = \begin{pmatrix} o_{1,1} & o_{2,1} & \cdots \\ o_{1,2} & o_{2,2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}_{K \times N} \begin{pmatrix} x_1 & x_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}_{N \times 1} \quad (2.1)$$

Equation (2.1) shows the compression taking place, compressing an $N \times 1$ signal to a much smaller $K \times 1$ signal.

2.5.2 Motivation and application

The high dimensionality of the problem is the main reason why this technique is used. Solving optimization problems with machine learning in high dimensional space is not something trivial according to Djolonga et al. [16]. One of the reasons for this is because the computational power required for the optimization problems tend to scale rapidly with the input size of the problem. This is also the case in this work and such decreasing the size cuts down on the computational time. As an example, running this algorithm on the largest data set shown in section 3.2, the sparse matrix of size 1230×4621 can be compressed to a 1230×386 matrix.

It remains to be verified that employing this compression does not give significantly worse predictions than not using it, something that will be explored in chapter 4.

2.6 Bandits

A bandit is, as described by Auer et al. [17], a theoretical machine, inspired by regular slot machines such that when played it produces some reward from some unknown probability

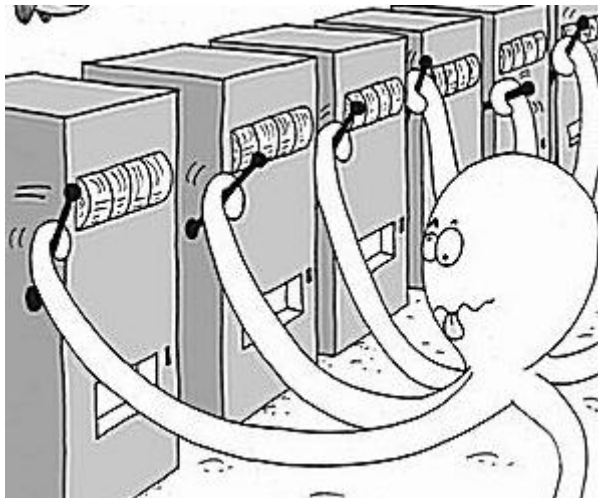


Figure 2.5: An example of what a multi-armed bandit looks like in theory.

Source: <http://research.microsoft.com/en-us/projects/bandits/MAB-2.jpg>

distribution specific to that machine. The inspiration comes from the following problem. Suppose you are a gambler in a casino. Your goal is to maximize your winnings and you have a number of slot machines to play on. At each time step you select one slot machine and play it. Some reward, in this case the amount of money won for that round, is observed and recorded. At the next time step you have access to the observed data of all the previous rounds. Now your goal is to maximize your winnings given this historic data. See Figure 2.5 for an idea of what a multi-armed bandit problem can look like.

2.6.1 Definition

A contextual multi-armed bandit (MAB) problem can have the following parameters: K is the number of arms, $a_{i,t} \in \mathcal{A}$ is a set of actions for a specific arm at a given time and $\mathbf{x}_{i,t} \in \mathbb{R}^d$ a set of contexts. Each arm has its specific unknown probability distribution D_i and produces a reward $r_{a_{i,t}} \sim D_i$ when played. The goal is then to either maximize the total reward over some time T or to minimize the regret over some time T . Regret here is defined as the difference from the observed rewards of the selected arms and the reward from the optimal arm at each time step $t = 1, 2, \dots, T$.

$$\text{Regret} = \sum_{t=1}^T r_{a_i^*, t} - r_t \quad (2.2)$$

2.6.2 Motivation and application

Many problems can be modeled as a MAB problem where you are left with a number of choices of actions and you can only select one at a time and only observe the result of taking that specific action.

Clinical trials could be seen as a MAB problem, as described by Katehakis and Arthur F. Veinott [18], where each treatment corresponds to one bandit. At each time step one of the treatments is selected and the corresponding machine is played and some reward is observed, commonly in a binary fashion, 1 is the treatment was successful or 0 is the treatment was not successful. A common goal in this setting is to try to maximize the number of successful treatments, which in the most basic setting where the probability that a treatment is successful is the same for all patients, is simply finding the best bandit to play.

The step from a clinical trial model to one of drug design is quite simple. Instead of maximizing the number of successful treatments as in the clinical trial case we maximize over the IC_{50} values of the compounds.

One motivation behind using this particular model is that bandits can handle the exploitation vs. exploration trade-off quite well. A common issue in optimization problems is getting stuck in local maxima/minima and exploitation vs. exploration deals with just that. The gambler has to choose whether to play the bandit that seems to be optimal right now (exploitation) or to play a bandit that the gambler currently has very little information of (exploration).

Figure 2.6 shows a bandit algorithm used in conjunction with a Gaussian process. It clearly shows the exploitation vs. exploration dilemma as the best points are rarely the ones with the highest mean so in order to find the optimal bandit to play it has to explore unknown space. Exactly how it does this is explained in subsection 2.7.2.

2.7 Selection algorithms

This section deals with two different ways of selecting which bandit to play. The first method, LinearBandit-TS is a Linear bandit based selection algorithm using a non-standard interpretation of Thompson sampling as it only samples from the posterior. The second one, GP-UCB, is a Gaussian process based selection algorithm that selects the bandit based on its predicted mean and variance for that particular bandit.

The main difference between the two is that Algorithm 2 uses a kernel to compare each of the considered bandits with what the GP has already learned so far whereas Algorithm 1 continuously updates a feature matrix which it uses to sample a new mean vector from a Gaussian distribution. Furthermore, Algorithm 1 requires that the reward distribution is R -sub-Gaussian and Algorithm 2 requires a scaling factor for its kernel.

2.7.1 Linear Bandit selection using Thompson Sampling

This algorithm is based on the work by Agrawal and Goyal [19] and simplified here since the contexts for this application do not change with time. The idea with this algorithm is that the rewards are distributed as a Gaussian distribution parameterized by our sample mean $\hat{\mu}$ and variance that is learned from the observed contexts.

Details There are N arms. Each arm has a context $\mathbf{x}_i \in \mathbb{R}^d$ associated with it. B is a $d \times d$ matrix capturing the observed contexts. This matrix is used to update the posterior mean and variance. $\hat{\mu}$ is a d -vector that is the sample mean of the rewards of the observed contexts. v is set to $v = R\sqrt{\frac{24}{\epsilon}d\log(\frac{1}{\delta})}$, where ϵ and δ are the hyper-parameters of the algorithm. R is set such that $r_{i,t} \in [\mathbf{x}_i^\top \mu - R, \mathbf{x}_i^\top \mu + R]$. The algorithm then samples from the posterior distribution and selects the arm that maximizes $\mathbf{x}_i^\top \hat{\mu}$. For more details, see the work by Agrawal and Goyal [19].

Motivation The range of rewards when optimizing for pIC₅₀ values is approximately known beforehand so the R parameter can be set without issue. Furthermore, its implementation is quite simple and the idea behind it is different enough from GP-UCB. It has also been shown as in Deshpande and Montanari [20] that linear bandits can work well for problems of high dimensionality.

Algorithm 1 LinearBandit-TS

```

 $B = I_d, \hat{\mu} = 0_d, f = 0_d.$ 
for all  $t = 1, 2, \dots, T$  do
    Sample  $\tilde{\mu} \sim \mathcal{N}(\hat{\mu}, v^2 B^{-1})$ .
    Play arm  $i \in \operatorname{argmax}_i \mathbf{x}_i^\top \tilde{\mu}$ . Observe reward  $r_{i,t}$ .
     $B \leftarrow B + \mathbf{x}_i \mathbf{x}_i^\top$ .
     $f \leftarrow f + \mathbf{x}_i r_{i,t}$ .
     $\hat{\mu} \leftarrow B^{-1} f$ .
end for

```

Algorithm 2 GP-UCB

```

GP prior  $\mu_0 = 0, \sigma_0$ , kernel  $\mathcal{K}$ .
for all  $t = 1, 2, \dots, T$  do
    Play arm  $i \in \operatorname{argmax}_i \mu_{t-1}(\mathbf{x}_i) + \sqrt{\beta_t} \sigma_{t-1}(\mathbf{x}_i)$ . Observe reward  $r_{i,t}$ .
    Perform Bayesian update to obtain  $\mu_t$  and  $\sigma_t$  by letting GP learn  $\mathbf{x}_i = r_{i,t}$ .
end for

```

2.7.2 Gaussian Process Bandit selection

This algorithm is based on the work by Srinivas et al. [6] but modified for this application since the contexts are static. The confidence bound is evaluated for each bandit and the bandit with the highest confidence bound is played.

Details There are N arms. Each arm has a context $\mathbf{x}_i \in \mathbb{R}^d$ associated with it. μ_0 and σ_0 are the initial hyper-parameters of the GP. \mathcal{K} is a covariance function that is used to predict the mean of other points. The kernel used in this work is the *Radial Basis Function* (RBF) kernel. Two points $\mathbf{x}_i, \mathbf{x}'_i$ have their similarity defined by

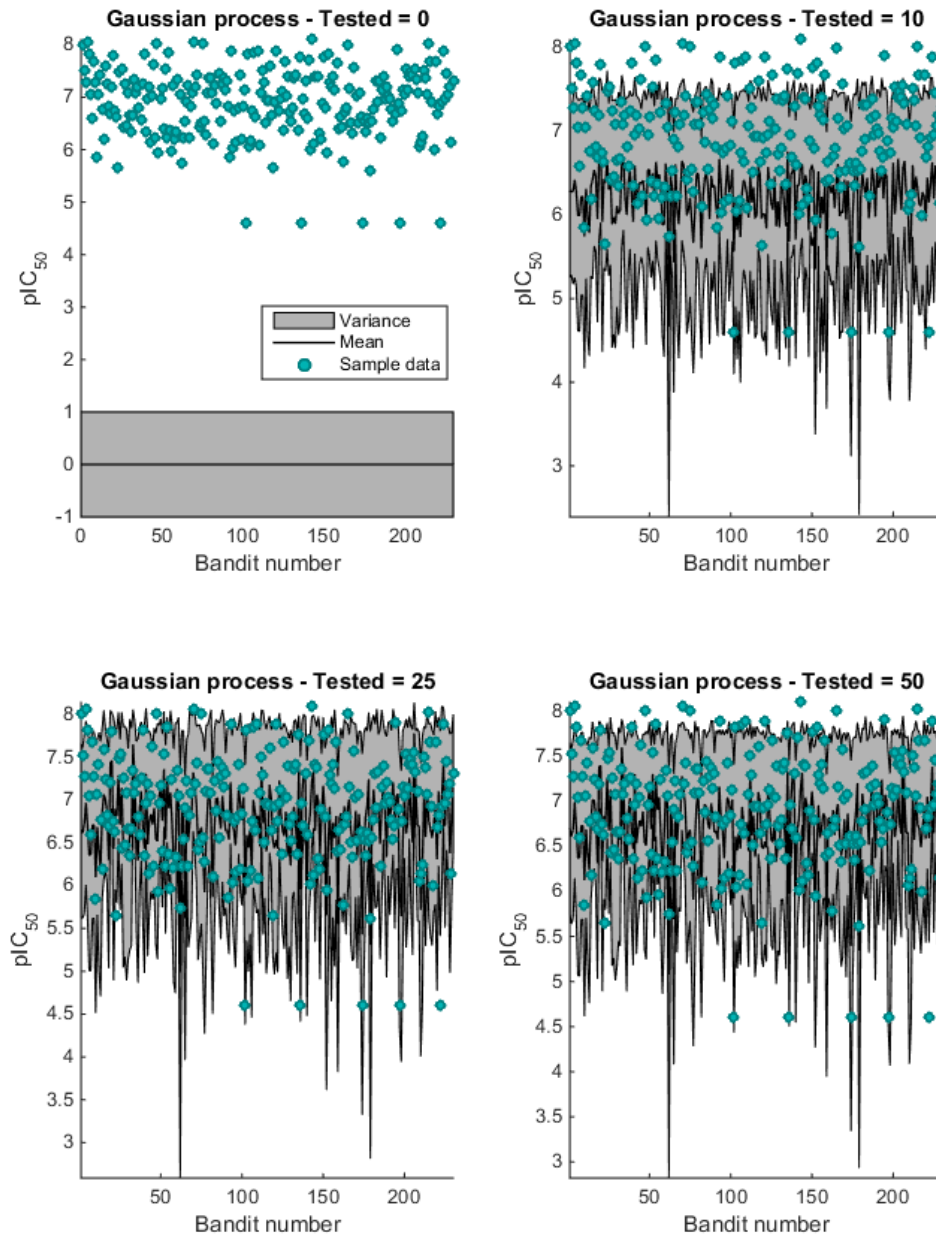


Figure 2.6: A plot of a Gaussian process in several stages. The points represent the true reward, the shaded area the variance and the solid line represents the Gaussian process mean.

$\mathcal{K}(\mathbf{x}_i, \mathbf{x}'_i) = \exp(-\|\mathbf{x}_i - \mathbf{x}'_i\|^2 / (2\sigma^2))$. $\beta(t)$ is a function that decays with time that is used to handle the exploitation vs. exploration trade-off. $\beta(t) = 2\log(dt^2\pi^2/6\delta)$. For more specific details, see the work by Srinivas et al. [6].

Motivation The reasoning for using a GP for bandit selection is that a way to compare points comes innately with the GP in the covariance function \mathcal{K} . Should the RBF kernel not work well enough then it could be substituted with some other kernel. The GP can also be used to predict properties of arbitrary points, something that is used in section 3.3.

The motivation behind using UCB as the selection method is that it handles the exploitation vs. exploration trade-off in a clever way. GP-UCB in action can be seen in Figure 2.6. The Gaussian process has no information of the reward values initially but as more observations are made it becomes more confident in its predictions. The numbers on the x-axis are the compound or bandit numbers. Each of those numbers corresponds to a specific context, as described in section 2.6.

3

Experiment

In this chapter we will describe what is meant by an experiment. In addition to this two different settings of the experiment will be discussed. The methods will be used on real data sets in the first setting.

In the other setting the data will be generated using a Gaussian process that has learned a subset of one of the real data sets. The reasoning for this is mainly that there is not that much data in the data sets and there are no repeated measurements for each compound so to strengthen the results the algorithms will also be tried on generated data.

3.1 Details

The experiments are both modeled as multi-armed bandit problems where each compound or data point has its own bandit. Playing these bandits is equivalent to observing the test results from running real chemical tests on the compounds.

The metrics used to test the performance of the algorithms used in this work is *cumulative regret* and *simple regret*. Regret is defined as in Equation 2.2 and the reasoning for testing two separate definitions of regret is that for cumulative regret we can see how close on average the whole tested set is to the optimal tested set but we miss out on information on how close the best compound in the tested data set is to the optimal compound in the data set. Let d be the size of the feature vector. Let N be the total number of samples. Each set contains a set of points (\mathbf{x}_i, y_i) , where $i = 1, 2, \dots, N$. Where $\mathbf{x} \in \mathbb{R}^{d \times N}$ is the feature matrix and $y \in \mathbb{R}^N$ is the reward vector. r^* is defined to be $r^* = \max_i y_i$. Let $r^{(i)}$ be the i -th highest reward in the data set. From these definitions we get the two metrics as defined in Equation 3.1 and Equation 3.2.

$$\text{Simple regret} = r^* - \max_{r \in \text{Tested}} \quad (3.1)$$

DataSet	#compounds	#distinct-substructures
CDK5	230	677
GNRHR	198	1523
MAPK14	610	2656
MGLL	1230	4621

Table 3.1: Four data sets with compounds with different target proteins of varying complexity.

$$\text{Cumulative regret} = \sum_{t=1}^T r^{(t)} - r_t \quad (3.2)$$

The goal is then to select a number of data points $0 < x \leq N$ that minimizes the regret as defined in Equation 3.1 and Equation 3.2. The tests of compounds are here assumed to be noise-free, i.e. multiple tests of the same compounds give the same result. Because of this assumption it does not make sense to play the same bandit twice and such the regret definition can be simplified to Equation 3.2.

3.2 Initial setting

The broad details of the four data sets considered in this setting are depicted in Table 3.1. The data sets were selected because of the contrast in sample size vs. space dimensionality compared with each other. This is most prevalent when the third and the fourth data sets are compared. In CDK5 there are few samples in a relatively small hyperspace while in GNRHR there are even fewer samples but with a much greater dimensionality. The explanation for this is that the compounds in CDK5 are more similar to each other than the compounds in GNRHR.

3.2.1 Data sets

The data sets come from patent data where a specific protein is targeted and its activity is sought to inhibit. The data sets are named after that protein and a short description of the protein follows. Each of the data sets has its common core substructure, or scaffold, for all the compounds in that set, as shown by Chen et al. [9]. This core can be connected with other atoms or more general substructures through the R-groups shown in Figure 3.1.

CDK5 Cyclin-dependent kinase (CDK) plays an important role in the cell division cycle and is often deregulated in developing tumors, as described by Meijer et al. [21].

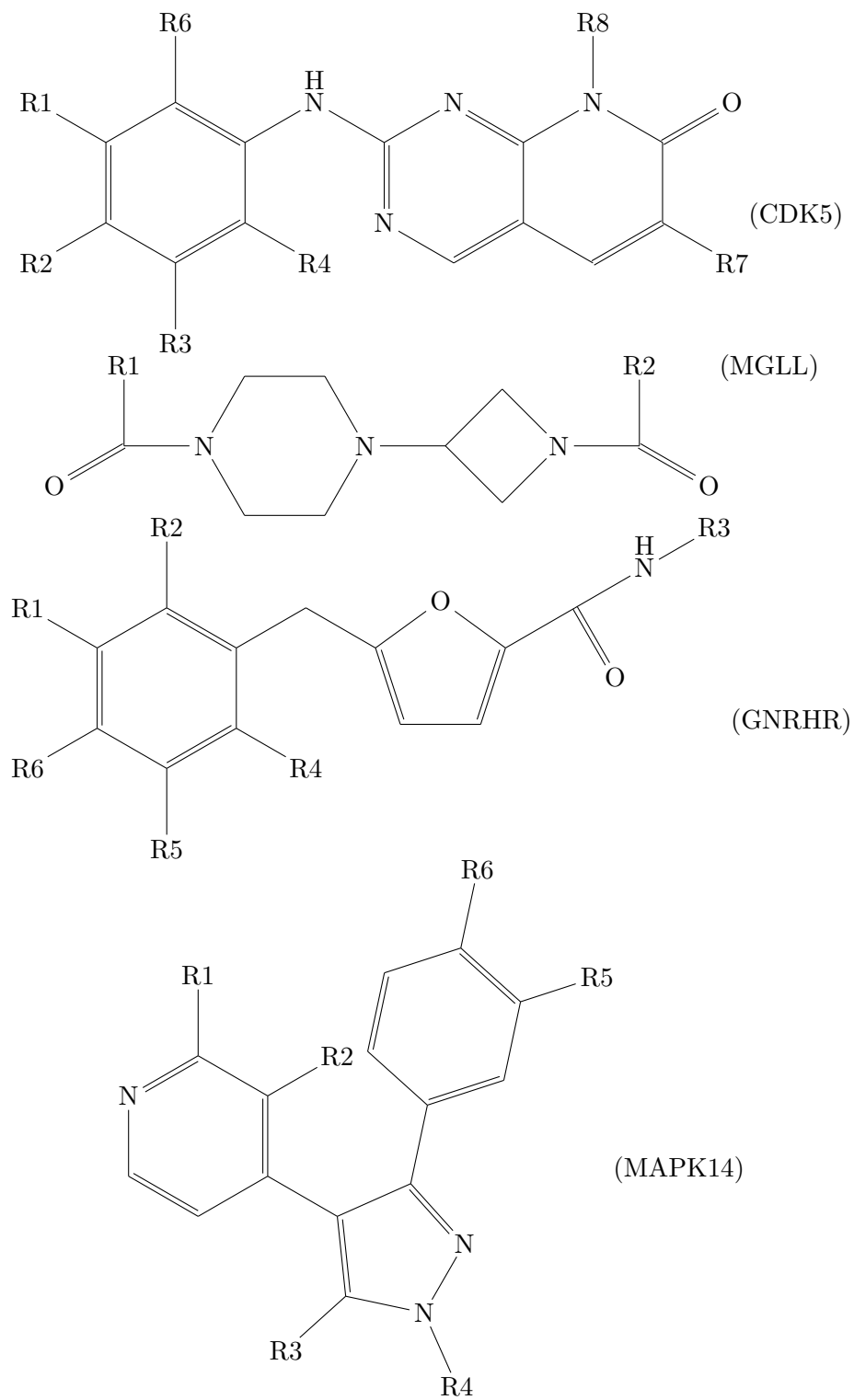


Figure 3.1: Graphs of the scaffolds and R-groups of the data sets.

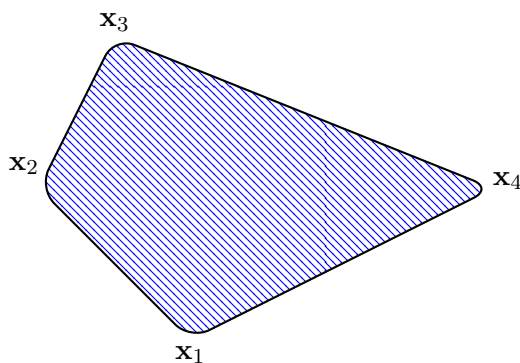


Figure 3.2: Dirichlet mixture of points in two dimensions.

GNRHR Gonadotropin-releasing hormone receptor (GNRHR) which activity of has been shown to indicate the progress of some cancers by Harrison et al. [22].

MAPK14 Mitogen-activated protein kinase 14 (MAPK14) has been shown by Paillas et al. [23] to make colon cancer cells more resistant to camptothecin-related drugs.

MGLL Monoacylglycerol lipase (MGLL) which levels of may be regulated for pain suppression and for inflammatory disorders, as described by Labar et al. [24].

3.3 Randomized setting

In this setting the data points will not necessarily correspond to realizable compounds since they are randomly generated points in the same dimension as learned by the GP. The GP can then predict the pIC_{50} values at those points which will then be used as rewards for the bandit selection. For this setting the noise-free assumption will be relaxed. Compounds may also be tested multiple times. The data set will remain consistent for a run with each algorithm and then generated anew for five runs in total. If the algorithms perform well in this setting, then there is good reason to believe they will work well in general.

3.3.1 Generation of data

A complete data set, e.g. (CDK5, MGLL, GNRHR or MAPK14) is first learned by a GP. New data points are then generated in the following way, $\mathbf{x}' = \sum_{i=1}^n w_i \mathbf{x}_i$, where $w_i \sim$

$\text{Dir}(1,1, \dots)$. Since $\sum_{i=1}^n w_i = 1$ and $0 \leq w_i \leq 1$ this corresponds to generating a point that is a mixture of the other points which lies inside a polygon as shown in Figure 3.2. Note that in the work here the dimensionality of that polygon is typically $\gg 2$.

3.3.2 Details

Since the model now takes noise and multiple testings into account the regret definitions have to be changed to accommodate this. The new definitions are shown in Equation 3.3 and Equation 3.4. The main difference is that each bandit now produces noisy rewards and so which compound is the best is not immediately apparent even after testing. Since it is now possible for the selection algorithm to choose the optimal bandit multiple times the new cumulative regret is just the difference between playing the best bandit for all trials and the sum over the entire reward history.

The noise for both the GP model and the rewards are assumed to be distributed as $\mathcal{N}(0,1)$.

$$\text{Simple regret} = r^* - \max_{r \in \text{Tested}} \quad (3.3)$$

$$\text{Cumulative regret} = Tr^* - \sum_{t=1}^T r_t \quad (3.4)$$

4

Results

This chapter contains the results achieved by running the methods used throughout this work in the two settings described in chapter 3.

4.1 Initial setting

The methods are all set to run for 25% of the total number of trials, meaning up to 25% of the compounds will have their properties revealed. The run is repeated ten times and the cumulative and simple regrets are averaged over all trials. The GP-UCB algorithm is initially tested with $\delta = 0.5$. The LinearBandit-TS algorithm is only run with $\delta = 0.5$.

4.1.1 CDK5

All methods both with and without compressed sensing are tried on the CDK5 data set. The results are shown in Figure 4.1.

4.1.2 GNRHR

The LinearBandit-TS algorithm without compressed sensing is skipped in this test because of the long time it takes for it to evaluate. Other than that all methods are present once again. Results shown in Figure 4.2.

4.1.3 MAPK14

The results are shown here only with compression since not using it takes a lot of time. In addition to the normal tests two tests with other δ values for GP-UCB are conducted. Results are shown in Figure 4.3.

4.1.4 MGLL

As in the previous section only tests with compression are conducted and with the very same delta values as in the previous section. Results are shown in Figure 4.4.

4.2 Randomized setting

The results of running the algorithms on the generated data set are shown in Figure 4.5. The results of running the algorithms on the generated data sets with noise are shown in Figure 4.6.

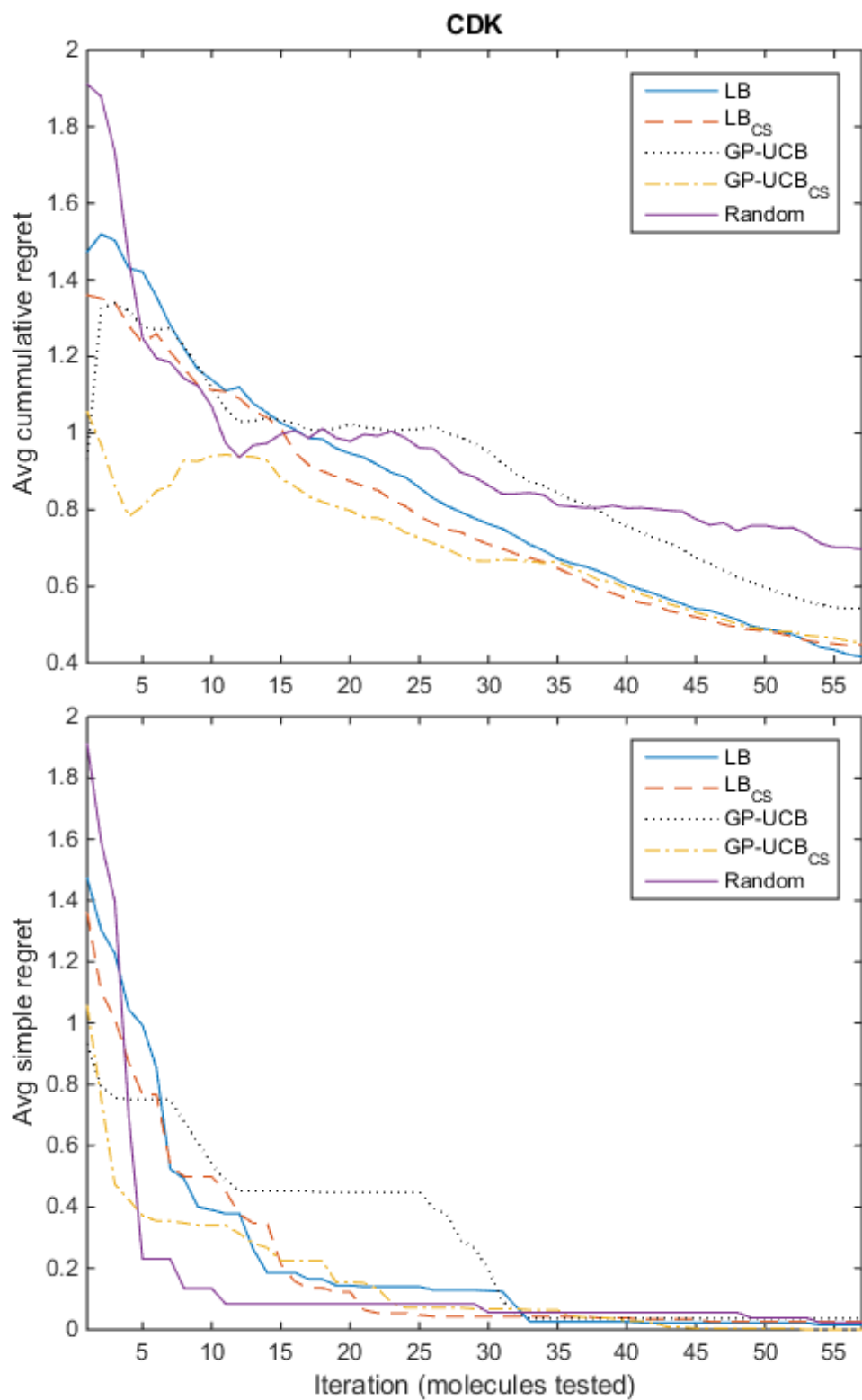


Figure 4.1: Test results for the CDK5 data set showing the average cumulative and simple regret.

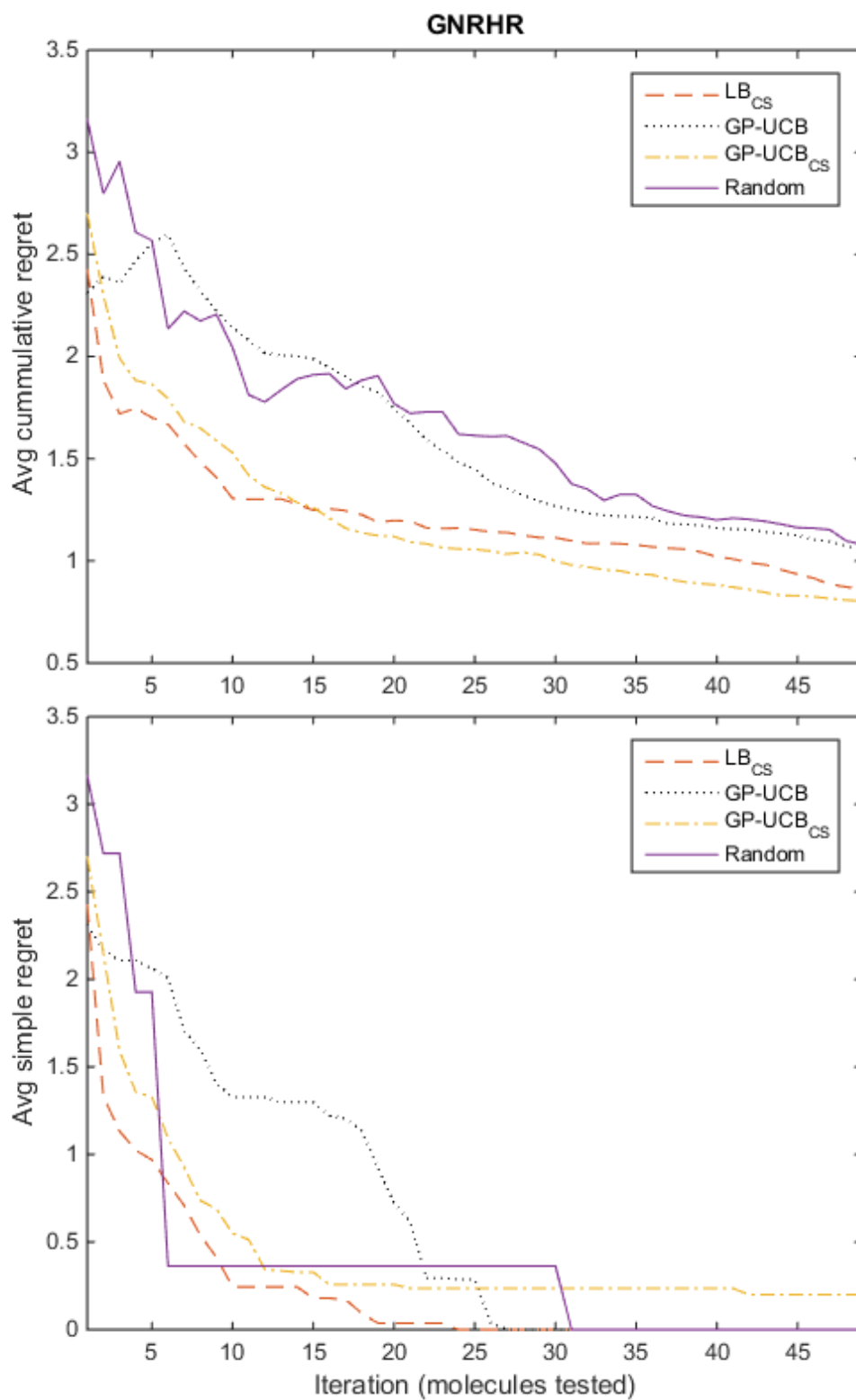


Figure 4.2: Test results for the GNRHR data set showing the average cumulative and simple regret.

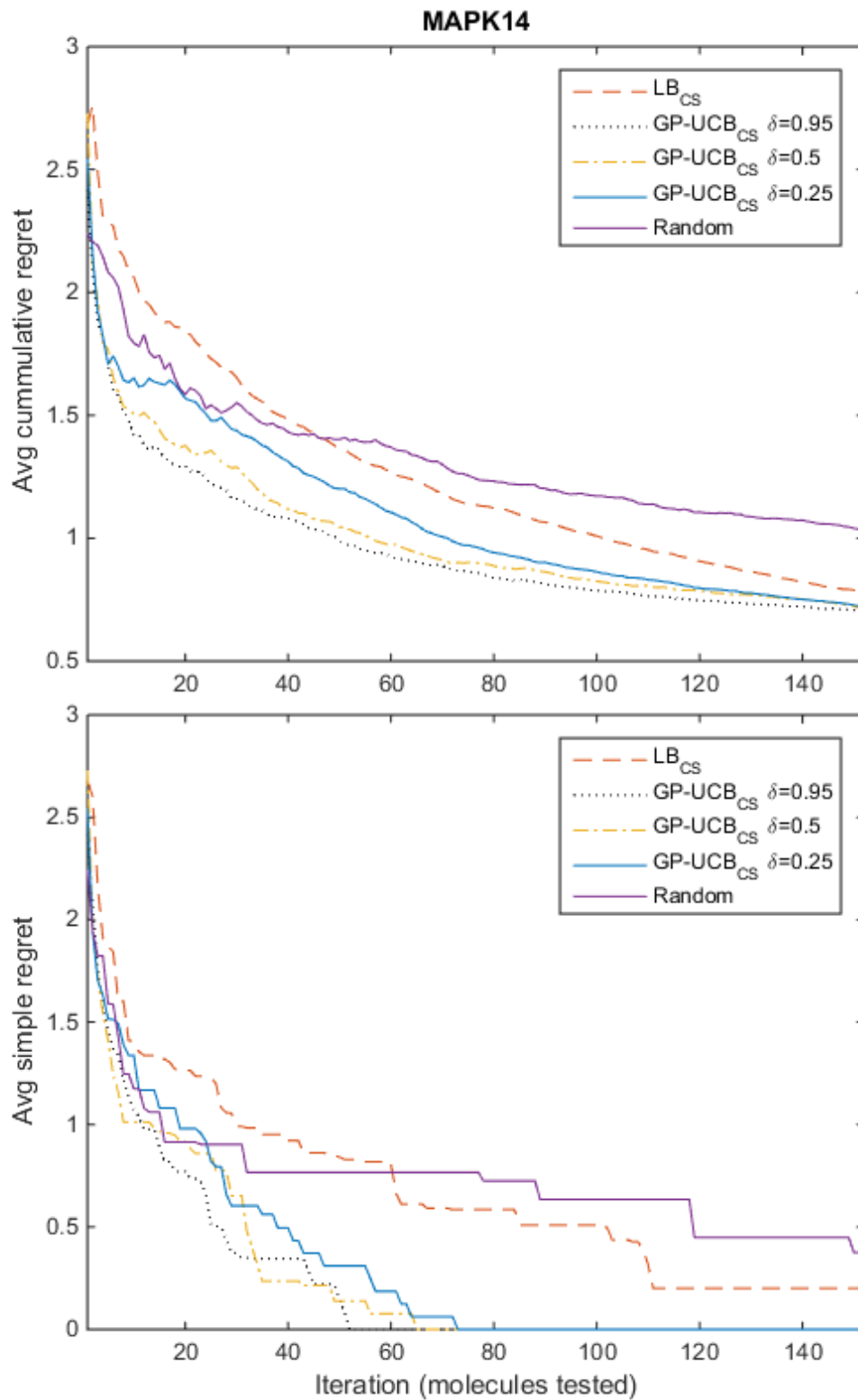


Figure 4.3: Test results for the MAPK14 data set showing the average cumulative and simple regret.

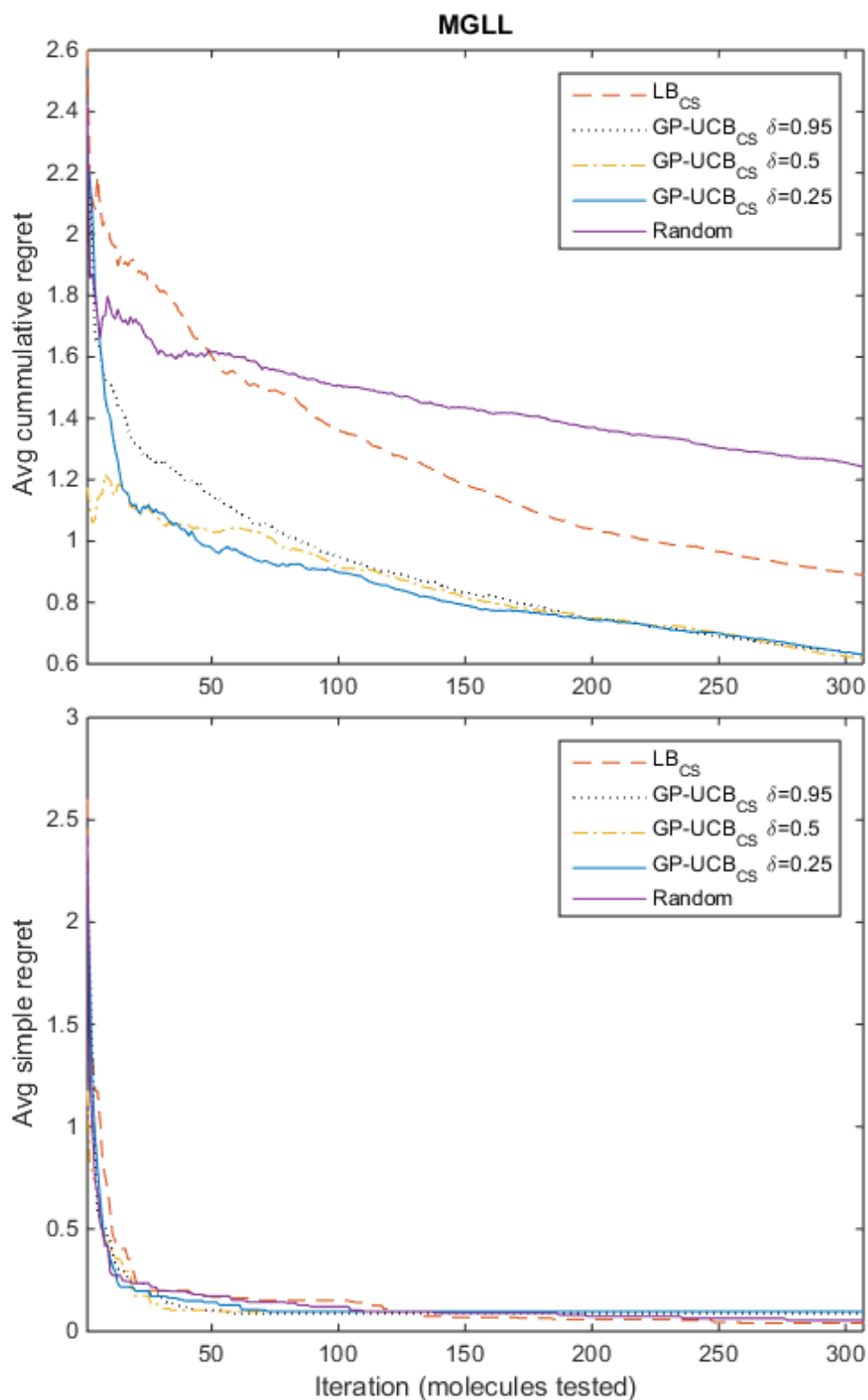


Figure 4.4: Test results for the MGLL data set showing the average cumulative and simple regret.

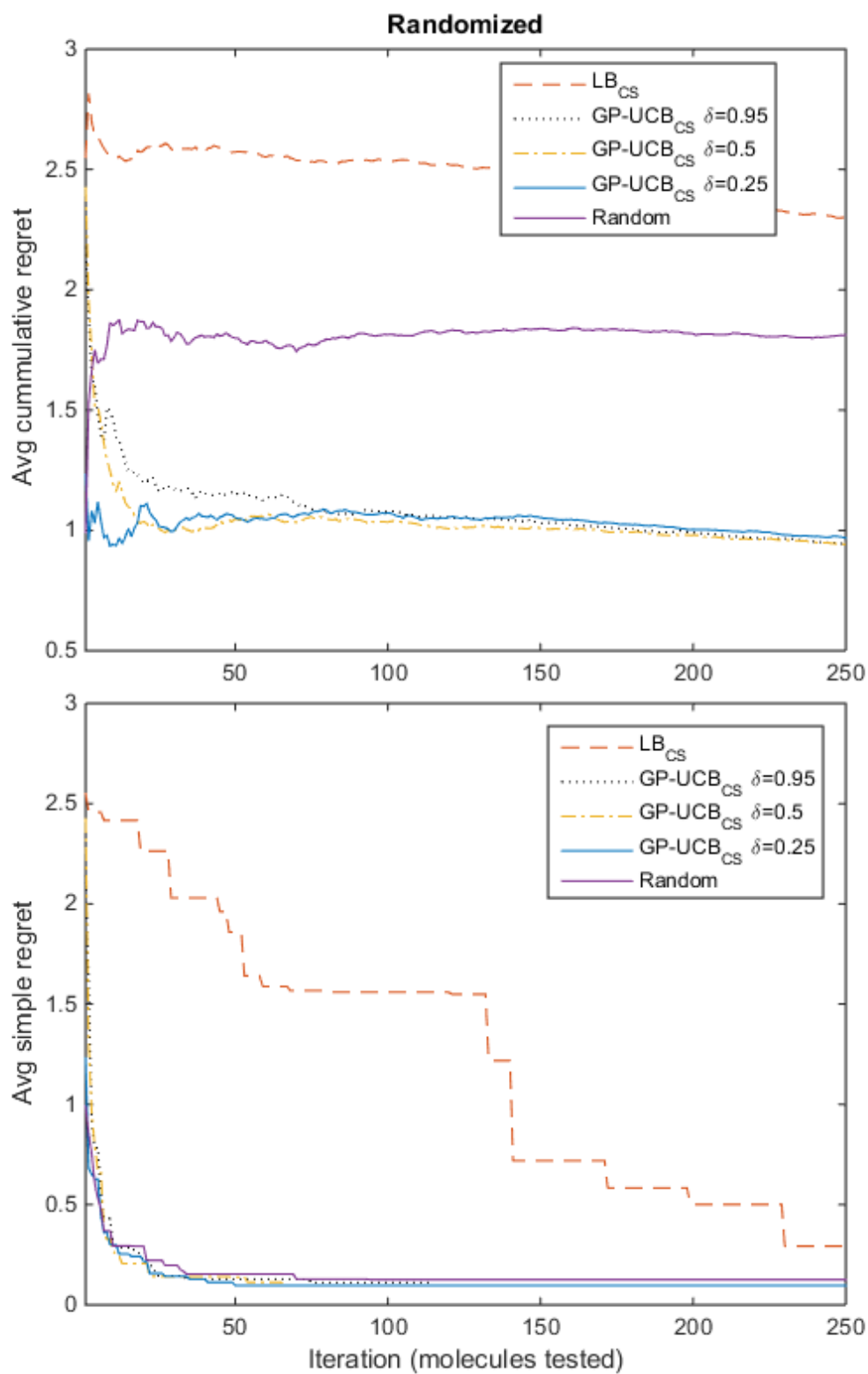


Figure 4.5: Test results for the generated data set showing the average cumulative and simple regret.

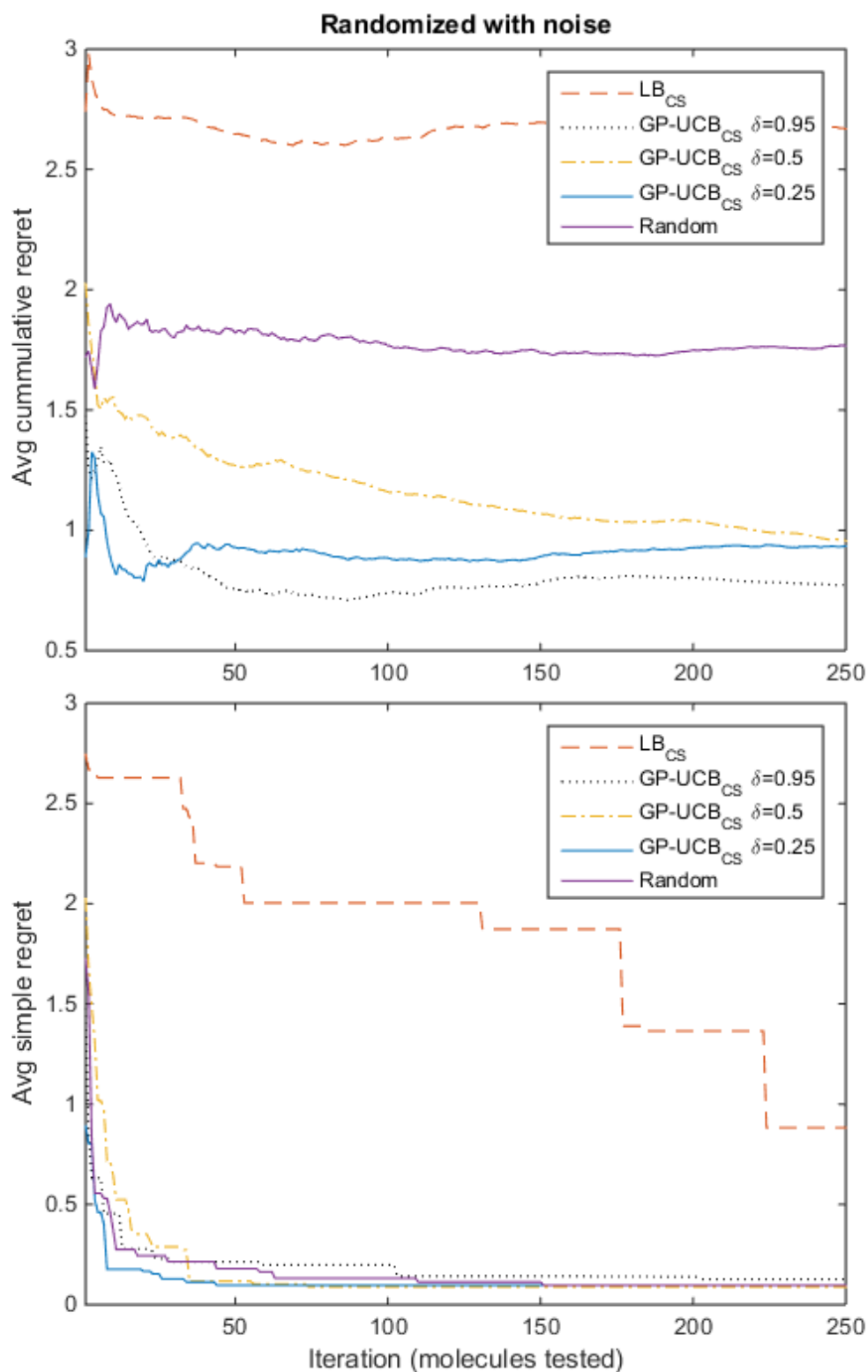


Figure 4.6: Test results for the generated data set with noise showing the average cumulative and simple regret.

5

Discussion

In this chapter our interpretation of the results shown in chapter 4 will be discussed as well as a comparison between the two different selection algorithms. It will also feature arguments for why the problems tackled and solutions described in this project might be interesting in the future.

5.1 Results

This section contains our interpretation of the results achieved from running the methods described in this work on the four known data sets as well as on the newly generated data.

5.1.1 Initial setting

The LinearBandit-TS algorithm seems to achieve similar results as the GP-UCB for the smaller data sets (CDK5, GNRHR). They both perform noticeably better than random selection and work especially well when using the compressed sensing algorithm to compress the features. For the larger data sets (MAPK14, MGLL) the algorithms seem to work even better compared to the random selection as shown in MAPK14 where GP-UCB manages to identify the best compound in the set in all runs. There is a great difference in the performance of the two algorithms for the MGLL data set if the average cumulative regret is compared. However, they perform similarly if the objective is to just identify one of the better or the best compound in the data set.

5.1.2 Randomized setting

It is quite clear that the GP-UCB algorithm works better than the LinearBandit-TS algorithm in this case but perhaps not that surprising since the data itself is generated from another GP. It is obvious that the linear bandit is learning in Figure 4.5. However,

it is learning far too slowly compared to GP-UCB. This because the linear bandit quite often decides to play the very same bandit a few times before moving on to another bandit.

The biggest difference of the results in the no-noise setting compared to the noisy setting is that it takes GP-UCB quite a bit more time in the noisy setting to identify the promising compounds. Note that we are dealing with quite heavy noise here. The rewards range from around 6 – 8 pIC₅₀ and the noise is $x \sim \mathcal{N}(0,1)$ so it is pretty likely that a relatively bad compound will be taken as a good one and similarly a good compound assumed to be bad. Some test setups performed okay even in this difficult task, especially the GP-UCB setups that value exploration more than normal ($\delta = 0.95$).

Overall this non-standard linear bandit with Thompson sampling did not work well in the randomized setting with repeated measurements. It is possible the standard algorithm would work better in this case.

Unfortunately we did not have access to information on how noisy the tests for the compounds in the patent data were however, realistically the noise should be quite a lot lower than what we used here.

5.2 Compressed sensing

The main reasoning for using the compression was to cut down computational time since the algorithms involve many heavy computations such as matrix inversions. Since the data sets contained relatively little information compared to their size (very sparse data) we had hoped the computational performance gain would weigh up for the potential information loss in the compression algorithm. So the need to verify that the accuracy of the predictions using compressed sensing would not be significantly worse than without compressed sensing became apparent.

Tests with compressed data performed as well or even better in all setups except on the CDK5 data set where LinearBandit-TS without compression achieved the best results. The reason for this may be that features with similar effect on the resulting reward might be clumped together which might make it easier for the selection algorithm to predict the rewards of untested compounds.

5.2.1 Gain for GP-UCB

If we assume the most computationally expensive operation is the matrix inversion which has been shown by Gall [25] to have a worst-case time complexity of about $O(n) = n^{2.373}$, where n is the amount of samples in the data set, then we can easily calculate the worst-case time complexity for using CS and not using CS for the GP-UCB algorithm. The number of features d for each sample is typically greater than the amount of samples but it only scales at most quadratically in the number of features as the kernel has to compare each point to each every other point. We then have a time complexity bound that is $O(n,d) = n^{2.373} + d^2$.

If we assume a density of ≈ 100 in the sparse feature vectors as has been common

<i>Algorithm</i>	GP-UCB	LinearBandit-TS
Regret R_T	$\sqrt{T(\log T)^{d+1}}$	$d^2\sqrt{T}$

Table 5.1: Theoretical regret bound comparison of GP-UCB and LinearBandit-TS.

in the relatively small compounds featured in this work, then we can reduce the d^2 term to $(100 \log(\frac{d}{100}))^2$ which is a lot smaller than d^2 for big d .

5.2.2 Gain for LinearBandit-TS

As in the previous subsection, the most expensive operation is the matrix inversion. The main difference here, however, is that the inverted matrix is of size $d \times d$ rather than $n \times n$ as in GP-UCB. This means the time complexity with compression is $O(d) = (100 \log(\frac{d}{100}))^{2.373}$ rather than $O(d) = d^{2.373}$ which is a significant difference for large d .

5.3 Exploitation vs. exploration with δ

The later tests with GP-UCB features different setups with varying δ values used to handle the exploitation vs. exploration trade-off. It is quite clear from the results that the differences are minor but probably favoring higher δ values overall meaning more focus on exploration. The main difference comes in the randomized setting with heavy noise. Initially the GP-UCB setup with less focus on exploration performed significantly worse than the other two which is likely the result of it getting stuck in a local maximum that in reality are just bad compounds masked as good compounds by the heavy noise.

5.4 Selection budget

Throughout this work the number of tested compounds in each data set has been set to 25% of the total number of samples. This number could be changed depending on how certain you would want to be of finding the best compounds in the data set as the algorithms become more accurate in their prediction as they learn more data. This is especially important for very noisy observations, as in the later experiments, since it may take a few tests of a single compound to determine how good its reward is compared to the rest of the compounds.

5.5 GP-UCB vs. LinearBandit-TS

The two algorithms achieved similar results apart from in the randomized setting. There are a few important differences between them, however, that we will go over here. Shown in Figure 5.1 is a comparison of the theoretical regret bounds of the two algorithms where T is the number of trials and d is the number of features.

5.5.1 LinearBandit-TS

To use this version of Thompson Sampling with a linear bandit it is crucial that it is known beforehand in what range the rewards can come in. The implementation mainly scales with the dimensionality of the features and so is quite slow for complex data sets without compression.

5.5.2 GP-UCB

It is important that the RBF kernel knows how much to scale the input data for the algorithm to work properly. This means that the input data has to be of about the same magnitude. This implementation scales with both the number of samples and the number of features but in general the number of samples takes precedence. It is easy to control the exploitation vs. exploration trade-off which might be important for it to work well, both for data sets where the data is vastly different and for data sets where the data is quite similar in nature.

5.6 General discussion

Since automated experiment design has gained increased visibility in recent times thanks to papers like Williams et al. [1] and King et al. [5], it is interesting to think of what impact intelligent experiment design can have on our society in general. The way from a drug in thought to a realizable drug is long. There are a number of phases it has to go through before we end up with a satisfying result, e.g. drug screening, drug design, drug development and clinical trials.

When a disease, process, trauma or similar is first sought to be treated, enhanced or inhibited potential compounds have to be found. One way to find suitable compounds is to use virtual screening as in Rester [26]. Here very large libraries of compounds are stepped through in hope of finding conceivable compounds.

Another automated way of doing drug screening is high-throughput screening as described by Hertzberg and Pope [27]. With this method researchers can quickly conduct several thousands of chemical tests by using high-throughput screening robots to aid with reagent mixing, preparation, transportation and analysis. The goal of this phase is to narrow down the potential compounds to perhaps a couple of hundred or a few thousands candidates for development.

It is in the following phase where the work done in this thesis is interesting. Now the number of candidate compounds is of a magnitude small enough such that the experimentalists can conduct meaningful experiments on the compounds in reasonable time. Even this phase can be fully automated as shown by the robot Eve in Williams et al. [1].

If the drug or drugs are promising enough then they can undergo clinical trials and later possibly commercialization. A fair assumption is then that since the automated experiment design decreases the time and effort required to develop these drugs, the resulting drugs should likely be either cheaper to develop or more potent. If either of

those cases is true, then automated experiment design is surely helping society to become a better place to live in.

6

Conclusion

In this chapter we will summarize our results and our thoughts of this work. We will conclude on when to use GP-UCB and when to use LinearBandit-TS going by our simplified explanations in section 5.5. We will also go over some improvements and interesting ideas that might be interesting in the future if one were to continue on with this work.

6.1 Results

We conclude that these methods work quite well in this context. Identifying promising elements in a data set when the number of tests is much smaller than the amount of elements in the data set is not a trivial task, especially when there is no training data or previous observations to make use of. The methods work online and they still manage to identify promising compounds in a relatively short period of time.

6.2 GP-UCB or LinearBandit-TS?

There is no clear winner between the two algorithms on the four data sets and instead one should look at the details of the problem and from that decide which one is better suited for the task. There is, however, a striking difference in performance in the randomized setting where a bandit may be played more than once.

- **GP-UCB**

- + δ for exploitation vs. exploration
- + Easy to change GP kernel
- Relatively complex
- Scales with samples – relatively slow for $N > 1000$
 - Needs to know how much to scale the vectors

- **LinearBandit-TS**

- + Simple
- + δ for exploitation vs. exploration
- Scales with features – slow without compression
- Slow to adapt?
 - Needs to know reward range

6.3 Concluding remarks

What we have not done in this work is to show that one of the algorithms is better than the other. Instead, we have rather simply noted their differences, special requirements, pros and cons.

We have not shown that compressed sensing works well for all kinds of data sets nor that the prediction and selection methods retain the same accuracy with compression as without compression. Our limited testing seems to imply that such is the case, however, extended testing would be required to show this.

6.4 Future work

More extensive tests with and without compressed sensing to determine gains and losses of using and not using it. Optimizing the different Gaussian process parameters for better results. Optimizing the linear bandit parameters for better results.

Replacing the current graphlet method for identifying similar structures in graphs (signature descriptors) by graph kernels with geometric embeddings as in Johansson et al. [2]. The authors show in that paper that their algorithm can work better or at least as good as current graphlet methods such as the one used in this work. There is a possibility that signature descriptors miss out on some interesting information concealed in the graph that a global graph kernel could discover.

Trying out other ways for the exploration and not only Thompson Sampling and UCB. There might be a better way of selecting the next drugs to test.

Combining GP with Pareto fronts to be able to do multi-objective optimization when there are multiple interesting properties for the given compounds as shown in Binois et al. [28].

Developing a bandit that can use an arbitrary predictor. As an example, Eklund et al. [13] has noticed that conformal predictors work well for prediction in the drug development setting so trying out conformal predictors could be interesting.

Using a fully Bayesian agent or a Bayes-Adaptive agent to do Bayesian Monte-Carlo planning, something that has been shown by Guez et al. [29] to be very powerful and which could avoid certain problems that simpler planning methods such as Thompson sampling might run into.

Combining GP with Thompson sampling to be able to "predict future predictions". It would then be possible to generate a tree of posterior distributions. These posterior distributions could then perhaps be used to make meaningful predictions using not only data from past and current observations, but also future predictions. The idea is shown in Figure 6.1.

The idea would be something like this: at each time step t , the Gaussian process ξ_t has made some observations $\{\mathbf{x}_i, y_i\}$, where $i = 1 \dots t$. For each bandit arm $a = 1 \dots N$, generate the *posterior predictive distribution* $p(y_a | \xi_t)$. Recursively play bandits until some *horizon* or *budget* is met. Each time a bandit is played, record the posterior predictive distribution $p(\mathbf{y} | \xi_t \cup \{x_j, \hat{y}_j, x_k, \hat{y}_k, \dots, x_a, \hat{y}_a\})$, where j, k are previously played bandits *following* trial t . Each of the nodes in the tree will thus correspond to different Gaussian processes ξ_T^* that has learned not only the observations from $t = 1 \dots t$, but also future predictions $t = t + 1 \dots T$. In the end, select and play the bandit that seems most promising, now using future predictions in addition to what has been used throughout this work.

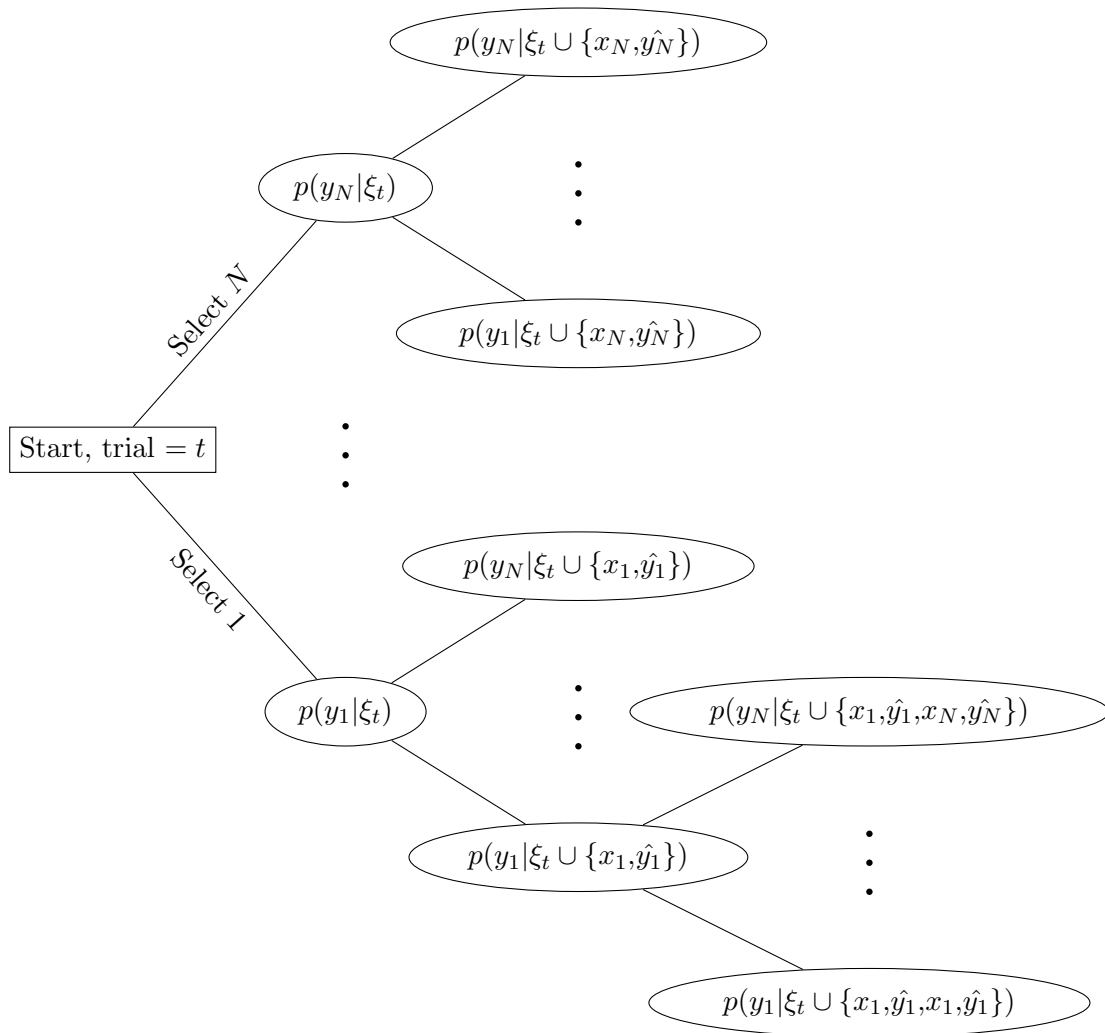


Figure 6.1: The tree of posterior predictive distributions of a bandit run at trial t .

References

- [1] K. Williams, E. Bilsland, A. Sparkes, W. Aubrey, M. Young, L. N. Soldatova, K. D. Grave, J. Ramon, M. de Clare, W. Sirawaraporn, S. G. Oliver, and R. D. King, “Cheaper faster drug development validated by the repositioning of drugs against neglected tropical diseases,” *Journal of The Royal Society Interface*, vol. 12, no. 104, 2015.
- [2] F. Johansson, V. Jethava, D. Dubhashi, and C. Bhattacharyya, “Global graph kernels using geometric embeddings,” in *Proceedings of the 31st International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 2014, p. 694–702.
- [3] GOSTAR databases 2012; GVK Biosciences Private Ltd.: Hyderabad, India.
- [4] “Astrazeneca,” <http://www.astrazeneca.com/About-Us/Key-facts>, accessed: 2015-05-21.
- [5] R. D. King, K. E. Whelan, F. M. Jones, P. G. K. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver, “Functional genomic hypothesis generation and experimentation by a robot scientist,” *Nature*, vol. 427, pp. 247–252, 2004.
- [6] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, “Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting,” *Information Theory, IEEE Transactions on*, vol. 58, no. 5, pp. 3250–3265, 2012.
- [7] A. Krause and C. S. Ong, “Contextual Gaussian Process Bandit Optimization,” in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2447–2455.
- [8] J. S. Soothill, R. Ward, and A. J. Girling, “The IC50: an exactly defined measure of antibiotic sensitivity,” *Journal of Antimicrobial Chemotherapy*, vol. 29, no. 2, pp. 137–139, 1992.

-
- [9] H. Chen, L. Carlsson, M. Eriksson, P. Varkonyi, U. Norinder, and I. Nilsson, "Beyond the Scope of Free-Wilson Analysis: Building Interpretable QSAR Models with Machine Learning Algorithms," *J. Chem. Inf. Model.*, vol. 53, no. 6, pp. 1324–1336, May 2013.
- [10] B. de Finetti, *Theory of Probability*. New York: Wiley, 1974, vol. 1.
- [11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- [12] J.-L. Faulon, D. P. Visco, and R. S. Pophale, "The Signature Molecular Descriptor. 1. Using Extended Valence Sequences in QSAR and QSPR Studies," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 3, pp. 707–720, 2003.
- [13] M. Eklund, U. Norinder, S. Boyer, and L. Carlsson, "The application of conformal prediction to the drug discovery process," *Annals of Mathematics and Artificial Intelligence*, pp. 1–16, 2013.
- [14] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based Compressive Sensing," *IEEE Trans. Inf. Theor.*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.
- [15] E. J. Candés, "Compressive sampling," *Proc. International Congress of Mathematicians*, vol. 3, pp. 1433–1452, 2006.
- [16] J. Djolonga, A. Krause, and V. Cevher, "High-Dimensional Gaussian Process Bandits," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 1025–1033.
- [17] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*. IEEE, 1995, pp. 322–331.
- [18] M. N. Katehakis and J. Arthur F. Veinott, "The Multi-Armed Bandit Problem: Decomposition and Computation," *Mathematics of Operations Research*, vol. 12, no. 2, pp. 262–268, 1987.
- [19] S. Agrawal and N. Goyal, "Thompson Sampling for Contextual Bandits with Linear Payoffs," *CoRR*, vol. abs/1209.3352, 2012. [Online]. Available: <http://arxiv.org/abs/1209.3352>
- [20] Y. Deshpande and A. Montanari, "Linear Bandits in High Dimension and Recommendation Systems," *CoRR*, vol. abs/1301.1722, 2013. [Online]. Available: <http://arxiv.org/abs/1301.1722>
- [21] L. Meijer, A. Borgne, O. Mulner, J. P. Chong, J. Blow, N. Inagaki, M. Inagaki, J.-G. Delcros, and J.-P. Moulinoux, "Biochemical and cellular effects of roscovitine, a potent and selective inhibitor of the cyclin-dependent kinases cdc2, cdk2 and cdk5," *European Journal of Biochemistry*, vol. 243, pp. 527–536, 2004.

-
- [22] G. S. Harrison, M. E. Wierman, T. M. Nett, and L. M. Glode, "Gonadotropin-releasing hormone and its receptor in normal and malignant cells," *Endocr Relat Cancer*, vol. 11, pp. 725–748, 2004.
- [23] S. Paillas, A. Causse, L. Marzi, P. de Medina, M. Poirot, V. Denis, N. Vezzio-Vie, L. Espert, H. Arzouk, A. Coquelle, P. Martineau, M. D. Rio, S. Pattingre, and C. Gongora, "MAPK14/p38 α confers irinotecan resistance to TP53-defective cells by inducing survival autophagy," *Autophagy*, vol. 8, pp. 1098–1112, 2012.
- [24] G. Labar, J. Wouters, and D. M. Lambert, "A Review on the Monoacylglycerol Lipase: At the Interface Between Fat and Endocannabinoid Signalling," *Current Medicinal Chemistry*, vol. 17, pp. 2588–2607, 2010.
- [25] F. L. Gall, "Powers of Tensors and Fast Matrix Multiplication," *CoRR*, vol. abs/1401.7714, 2014. [Online]. Available: <http://arxiv.org/abs/1401.7714>
- [26] U. Rester, "From virtuality to reality - Virtual screening in lead discovery and lead optimization: a medicinal chemistry perspective," *Curr Opin Drug Discov Devel*, vol. 11, no. 4, pp. 559–568, 2008.
- [27] R. P. Hertzberg and A. J. Pope, "High-throughput screening: new technology for the 21st century." *Curr Opin Chem Biol*, vol. 4, no. 4, pp. 445–451, Aug. 2000.
- [28] M. Binois, D. Ginsbourger, and O. Roustant, "Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations," *European Journal of Operational Research*, vol. 243, no. 2, pp. 386–394, 2015.
- [29] A. Guez, D. Silver, and P. Dayan, "Better Optimism By Bayes: Adaptive Planning with Rich Models," *CoRR*, vol. abs/1402.1958, 2014. [Online]. Available: <http://arxiv.org/abs/1402.1958>