**Computer Science 380 Research Project**

Final Report

A new tool for the Search and Classification of Binding Sites

Hasnain Cheena

Dr. Joerg Wicker

Dr. Katerina Taskova

[02/11/2020]

**ABSTRACT**

This report proposes a new method for binding site search and investigates future work into using machine learning for binding site classification.

**LIST OF FIGURES**

Fig. 1. This is an example of figure caption………………………………………………………………… (6)

**LIST OF TABLES**

Table 1: This is an example of a table caption………………………………………………………………. (6)

# Introduction

The usage of mass spectrometry within the drug development industry is extensive. However, a large amount

of the analysis work is still completed manually. The time taken to complete analysis tasks such as finding

where the substances have bound range from days to months of work depending on the complexity of the

reaction and expertise of chemist. In this report we propose a new analysis tool, called *BindingSearch*, which

can be used in the identification of binding sites from mass spectrometry data. The aim of the tool is to be able

to identify binding sites in an automated fashion reducing the lead time of the analysis to hours. With such a

tool the lead time to

complete such analysis works. In this study, the tool was tested using data from multiple top-down electrospray

ionization mass spectrometry sessions observing reaction of metal complex (Oxaliplatin, Transplatin and

Cisplatin) reactions with Ubiquitin. The proposed method has been compared to the current method proposed

in [5]. Results show that the proposed method can beats the current method and reduce the time taken for

binding site analysis to a few hours.

# Related Work

The small amount of past work relating to automated binding site analysis comes from [5]. In [5], the search uses a brute-force method to find binding locations. The method creates a difference spectrum by subtracting the unbound and bound spectrums and then matches the peaks of the difference spectrum to a theoretical spectrum of the primary reactant. In [5] the primary reactant was Ubiquitin. This method has a low accuracy for finding binding sites and produces many false positives. The large number of false positives arise because the unbound and bound spectrums are not aligned before subtraction. Furthermore, the low accuracy is a consequence this incorrect alignment. This is because the creation of the difference spectrum is a method to perform filtering removing noise and peaks that are unbound.

Taking the learnings of this study, in our proposed method we have added dynamic time warping stage to align the sequences before subtraction, decreasing the number of false positives and increasing accuracy.

# BindingSearch

## Data Preprocessing

In this study, broadband spectra of Ubiquitin incubated with cisplatin, transplatin, oxaliplatin and rapta-c was spectraly deconvoluted using the Bruker DataAnalysis software. Spectral deconvolution is the process by which groups spectral peaks into isotopic envelopes and extracts the monosioptic masses of the fragmented ions. Maximum Entropy deconvolution (using default parameters) was used between 5000 m/z and 10000 m/z. Figure X shows an example of the spectral deconvolution. The black spectrum is the raw experimental spectrum and the red spectrum is the deconvoluted spectrum.

Graphical user interface, application

Description automatically generated

Figure 1: Example of spectral deconvolution. Black spectrum is raw and red spectrum is deconvolution

## Method

This section contains the methodology and usage of the proposed tool.

### Required Search Inputs

The proposed binding site search method takes in three inputs:

1. **Unbound spectrum**: Deconvoluted spectrum of protein. In this study, it was the deconvoluted spectrum of Ubiquitin. It was passed into the algorithm as a comma-separated-value file (.csv) containing mass-charge and intensity. An example of the unbound spectrum is shown below in Figure X.
2. **Bound spectrum**: Deconvoluted spectrum of protein bound with the key reactant. In this study, it was the deconvoluted spectrums of Ubiquitin bound with Cistplatin, Transplatin, Oxaliplatin and Rapta-C.

This spectrum was passed into the algorithm as comma-separated-value file (.csv) containing mass-charge and intensity. An example of the bound spectrum is shown below in Figure X.

1. **Reactant list**: List of all substances in the reaction. This was passed into the algorithm as a comma-separated-value file (.csv) containing compound formula, compound mass, minimum number of atoms that can be present and maximum number of atoms of the compound that can be present. An example of the reactant list is shown below in Figure X.

### Stages of Search

The proposed search method has 4 core stages: Firstly, a difference spectrum is created by aligning the

unbound/bound spectrums using dynamic time warping (DTW) and then subtracting them. Secondly, a list of

theoretical binding sites is created from the reactant list by generating all possible combinations of the final

product. Thirdly, peak search is conducted by which peaks in the difference spectrum are identified. These

peaks represent potential experimental binding sites. Finally, the peaks found in the previous stage are matched

to the theoretical binding list. Figure X shows the entire algorithm.

Graphical user interface, text, application

Description automatically generated

Figure 2: Overview of BindingSearch

### Create Difference Spectrum

A difference spectrum is created by first aligning the unbound and bound spectrums and then the aligned spectrums are subtracted from one another. This step is performed to

First, the spectrums are aligned using dynamic time warping. This is necessary to ensure that the peaks align in the unbound and bound spectrums.

Once the peaks are aligned, the spectrums are simply subtracted from one another. This creates a difference spectrum.

### Generate Theoretical Binding Site List

In this stage, a theoretical list of binding sites is created from the reactant list by generating all possible

combinations of the final product. The output of this stage is a list containing the formula and mass of a

The two factors needed to generate the various combinations are the formula of the substance and its min/max

availability. The min/max availability dictates how many of a chemical can be in the final product.

The generation problem can be represented as a tree where:

1. Each node is a combination of reactants.
2. The root node is the primary reactant. In our case it was Ubiquitin.
3. Each level has one more reactant than the previous. Therefore, for ‘n’ reactants the tree will be capped at a depth of n.

Diagram

Description automatically generated

Figure 3:Example of Search Tree

The tree is generated in a breadth first search pattern. Where each level created iteratively and stored in memory. The nth level contains all possible combinations of final product that are n+1 in size.

### Peak Search

In this stage, peaks are identified within the difference spectrum. Peaks were found were classified according to the following criteria: Peaks are points:

1. That have two direct neighbors with a smaller amplitude
2. Whose intensity value is in the top n%, where ‘n’ represents how reactive the secondary reactant is. For example, this parameter was 0.05 for cisplatin, transplatin and oxaliplatin. Furthermore, it was 0.01 for Rapta-C. This was because Rapta-C is less reactive than the platinum complexes.

Figure X below an example of the peak identification algorithm.

### Match Peaks

In this stage, potential binding sites within the theoretical list are matched to the found experimental peaks. The mass value of a potential binding site is compared against the masses of experimental peaks. A potential binding site is successfully matched to a peak if the mass value of the potential binding site is within 0.5 Da of the experimental peak.

The output of this stage is a list of the matched potential binding sites.

## Search Speed Improvements

Changes were made to the search itself to improve its performance. These improvements can be categorized into two groups; tree pruning and complier setup.

### Tree Pruning

Pruning the tree reduces the number of nodes to be searched and thus improves the memory/speed of the algorithm. Tree pruning techniques used were:

1. *Unique permutation constraint:* Only unique permutations can be added to the tree. For example, in Figure X level 3 you can see a repetition of combinations. Therefore, with the unique permutation constraint only one of the nodes would be allowed in the search tree.
2. *Max peak constraint*: Only permutations whose total mass is lower than the mass of the largest detected peak can be added to the tree.
3. *Key reactants constraint*: Only nodes that contain both the primary and secondary reactant can be added to the tree. In Figure X, Ub is the primary reactant and Cl is the secondary reactant. Therefore, nodes like *Ub + Na* would be pruned off as they don’t contain Cl.

### Compiler Setup

Complier Setup methods used:

1. *Numba*: Numba is an optimization library that complies Python to machine code at runtime to speed up the code. It was used to speed up the search

## Results

The search was validated using the binding site analysis results from these papers [1], [2]. The metrics used to measure the effectiveness of the search include accuracy and number of results produced. The accuracy metric measures whether the final results contain the actual binding sites. In comparison, the number of results produced is key because the algorithm is conservative and tends to report many false positives, therefore this metric gives an indication of that. Figure X shows the accuracy of the proposed method (Method A) compared to an alternative method proposed in [5] which is Method B.

Figure X above shows that the search proposed in this report (Method A) clearly outperforms the current  
method (Method B). The figure shows that the current method does not capture all the correct binding sites but the proposed method always captures all the correct binding sites.

Figure X below shows the number of binding sites predicted (number of results produced) for the proposed method (Method A), current method (Method B) and actual number of binding sites. From the figure it can be observed that the proposed method produces far less false positives than the current method.

## Flask Web Development

To serve the algorithm for use, BindingSearch was incorporated within a web application that can be accessed by chemists. The Flask framework, which is a Python micro-web development framework, was used to perform this task. As shown in Figure X. the application is a single web page, where the user can upload the three required datasets (as stated in Section 4.1). Once processing is complete, the user can also download the analysis. Figure X shows screenshots of the web application, before and after processing has been completed.

# Future Work

Generally, this report has shown that applying a search method to automate binding site search has been successful. However, there is more work to be done in this area.

Firstly, the search algorithm can be further fine-tuned to increase its effectiveness. A key issue with the algorithm currently is its tendency to produce many false positives. To reduce the number of false positives encoding chemistry knowledge within a filter stage at the very end would reduce the number of false positives drastically. Furthermore, adding drift correction [4] would ensure that the final binding sites reported would be more accurate.

Secondly, machine learning can be applied to classify peaks into binding sites. In this situation, features would need to be extracted from the spectrums such as proposed in [3], [4] and [5]. Computational techniques from areas such as spectrum matching can be applied to binding site classification. The aim is that the learning algorithm would instead figure out what binding sites appear as through the features and classify peaks into binding sites.

# Conclusions

Conclusions should be derived based on the overall work. It is a good practice to highlight the future directions.

# References

1. K. Qian, C. Zhou, M. Allan, and Y. Yuan, “Modeling of load demand due to EV battery charging in distribution systems,” *IEEE Transactions on Power Systems*, vol. 26, no. 2, pp. 802–810, 2011.
2. A. Ukil, Y. M. Yeap, K. Satpathi, and N. Geddada, “Fault identification in AC and DC systems using STFT analysis of high frequency components,” in Proc. IEEE Innovative Smart Grid Technol. Conference (ISGT), Auckland, New Zealand, 2017.
3. J.J. Grainger and W.D. Stevenson, *Power System Analysis,* New Jersey, USA: McGraw-Hill, 1994.
4. Cigré Joint Working Group A3-B4.34, “Technical requirements and specifications of state-of-the-art HVDC switching equipment,” Technical Report, TB 683 2017, 2017.
5. P. Cairoli, “Fault protection in DC distribution systems via coordinated control of power supply converters and bus tie switches,” Ph.D. dissertation, Dept. Elect. Eng., Univ. South Carolina, SC, USA, 2013.
6. J. E. Hill, S. D. A. Fletcher, P. J. Norman, and S. J. Galloway, “Protection system for an electrical power network,” U.S. Patent 8842401B2, 2014.
7. *IEEE Standard Common Format for Transient Data Exchange*, IEEE Standard 37.111-1991, ver. 1.8, 1991.
8. Infineon Technologies, “Technical Info & Datasheet: IGBT Modules FF1000R17IE4,” 2013.
9. *Reference Manual, SimulationX*, ver. 3.8, ESI ITI GmbH, Dresden, Germany, 2016.
10. Solar dataset, 2017. [Online]. Available: <http://www.solarrepository.sg/pv-systems-database>
11. Paris agreement, 2016. Available: <https://unfccc.int>