5-2013

Algorithms and a Software Application for the Discovery of Heparin-Binding Proteins for Chemical Analysis

Christopher Dempewolf
University of Arkansas, Fayetteville

ALGORITHMS AND A SOFTWARE APPLICATION FOR THE DISCOVERY OF HEPARIN-BINDING PROTEINS FOR CHEMICAL ANALYSIS

ALGORITHMS AND A SOFTWARE APPLICATION FOR THE DISCOVERY OF HEPARIN-BINDING PROTEINS FOR CHEMICAL ANALYSIS

A thesis submitted in partial
fulfillment of the requirements for the degree of
Bachelor of Science

By

Christopher L. Dempewolf
University of Arkansas, 2013
Bachelor of Science in Computer Science

May 2013
University of Arkansas

ABSTRACT

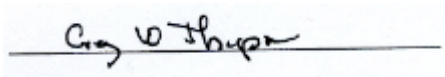
Heparin is a biological molecule that plays a vital role in anticoagulation. As such, it is used for the prevention of blood clotting in a variety of medical disorders. However, it is easily contaminated with foreign substances, and this can prove fatal for a person receiving heparin. In order to prevent this, proteins that easily bind to heparin need to be added to the solution. Then, the unwanted substances can easily be filtered out. Certain sequences or patterns of amino acids are known to have a high probability of binding to heparin. Thus, proteins that contain large numbers of these sequences of amino acids are more likely to bind to heparin. This research is focused on programmatically identifying these proteins so that they may be tested in the lab. The resulting software program was run on a number of different sets of proteins and various results were gathered. If the chemical phase of this research proves successful, many new proteins could be identified and used in practice to save lives from harmful heparin solution contaminants.

This thesis is approved for recommendation
to the Honors College of Engineering Council.

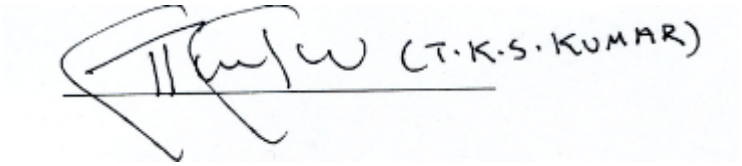
Thesis Director:

Wing-Ning Li

Thesis Committee:



Craig W. Thompson



Thallapuranam K. Suresh Kumar

THESIS DUPLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this thesis when needed for research and/or scholarship.

Agreed_____
Christopher Dempewolf

Declined_____

ACKNOWLEDGEMENTS

I thank Dr. Li for his continued support and for his patience with me. Without him I would not have been able to complete this work. I thank Dr. Thompson for all of his advice and help over the past two years and for providing the general outline for this thesis. I thank Dr. Kumar for providing the direction of the project and the chemical background. I also want to thank the Computer Science and Computer Engineering Department and the Honors College for giving me this wonderful opportunity to participate in undergraduate research.

I want to thank the research group that I worked with on this project – Dr. Li, Dr. Kumar, and Dr. Kumar's team: Srinivas Jayanthi, Jacqueline Morris, and Meghna Chopra. I have used a previous paper that our team wrote to help in writing this thesis.

I also thank my mom for always being there for me and helping me when times were rough.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Problem

Heparin is a molecule found in the extracellular matrix of all eukaryotic cells. Due to its superior blood clotting prevention abilities, it has received widespread medical attention. For example, blood clots may form before or after surgery and during some medical procedures, and heparin is used to prevent such occurrences. Specifically, it is used to treat blood clots that may form during acute coronary syndrome, atrial fibrillation, deep-vein thrombosis, pulmonary embolism, and heart surgery.

The problem is that heparin solutions can become easily contaminated. These contaminations are difficult to detect and can mean death for the heparin recipient. However, in recent years a novel filtration method has been developed to remove these contaminants. This filtration method requires the use of heparin-binding proteins, and thus, is the motivation for this project.

Currently, there exists no computer-aided solution for finding specific patterns of amino acids within proteins to indicate said protein's probability of binding to heparin. Identifying these proteins will make the aforementioned filtration method much easier, and will consequently help save lives in the process.

1.2 Objective

The objective of this research project is to develop a tool that will programmatically discover certain patterns of amino acid substrings in a protein that indicate said protein is likely

to bind to heparin, and furthermore, to give the amino acid composition of the substrings so that they may be synthesized and tested in the laboratory.

1.3 Approach

The first step of this project is a chemical one – examine a group of known heparin-binding proteins and find common subsequences of amino acids that may be an indicator of heparin-binding ability. An algorithm is then developed to search for these sequences given arbitrary proteins as input. The algorithm has three separate outputs: pattern location output, pattern composition output, and contiguous pattern output.

The pattern location output displays the numerical and graphical location of the matches of the patterns within the proteins. It is organized first by proteins, then by patterns. For each protein it displays the location of the occurrences for each pattern and the number of occurrences for that pattern.

The pattern composition output determines the amino acids that comprise each pattern, since each pattern is merely a binary sequence and conveys no information as to what its exact composition is. It displays a matrix for each pattern, and for each character in the pattern it displays which amino acid corresponds most frequently with each character in the pattern.

The contiguous pattern output finds two or more patterns whose matches within a protein are directly adjacent to one another. If one pattern occurs in a protein and is immediately followed by another pattern (possibly the same pattern), the resulting concatenation of these two patterns is also of interest. This output discovers and returns all such occurrences.

1.4 Organization of this Thesis

This thesis is organized as follows: Chapter 2 describes some necessary chemical background related to amino acids, proteins, heparin, and heparin-binding proteins. Chapter 3 describes the architecture of the system and the algorithms that were used. Chapter 4 talks about the results that were obtained and what these results mean. Chapter 5 summarizes this paper, explains the significance of this work, and details the next steps to be taken.

2. BACKGROUND

2.1 Key Concepts

This section provides a brief overview of some general chemistry concepts and terminology needed to read this paper.

2.1.1 Proteins and Amino Acids

Proteins and amino acids are the biological compounds that comprise all living things. There are 20 amino acids found in nature, and these 20 amino acids combine uniquely to form new proteins. Thus, every protein can be unambiguously represented by a sequence of a letters of an alphabet of size 20 where each letter corresponds to a particular amino acid.

2.1.2 Binding Affinity of Amino Acid Sequences to Heparin

All molecules are basic, acidic, or neutral depending on if they gain or lose protons in interactions with other molecules. There are three basic amino acids that easily bind to protons and, consequently, have a positive charge. Sequences of amino acids that contain a large number of these three basic amino acids will easily, due to their positive charge, bind to negatively charged molecules. Since heparin is a negatively charged molecule, this research project focuses on sequences of basic/non-basic amino acids. Thus, a sequence of amino acids can be represented as a binary sequence (e.g., XBXBX, where "B" is for "basic" and "X" is for "non-basic"). These binary sequences of basic and non-basic amino acids will hereafter be referred to as "patterns," whereas sequences of amino acids with no particular import will be referred to as "sequences."

The amino acid sequences of a set of proteins shown in vivo to bind to heparin were examined, and from these, 19 amino acid patterns were found to play a role in binding to heparin. These patterns were the primary patterns used for testing with this program.

2.1.3 Proteins and Amino Acid Patterns

The patterns described above are essentially regular expressions that represent a set of strings of amino acids. For example, the pattern XBX represents a set of 867 (17 * 3 * 17) amino acid strings. The "X" and "B" in a patterns string are, in essence, sets where B represents the 3 positively charged amino acids (histidine, arginine, and lysine), and X represents the remaining 17 natural amino acids. A peptide chain (protein string) $P = a_1a_2a_3$ is said to "match" XBX if and only if $a_1 \in X$, $a_2 \in B$, *and* $a_3 \in X$.

2.2 Related Work

There are several other programs that have been developed to search amino acid and nucleic acid sequences. Some of these tools are MEME Suite (Bailey, 2009), QuasiMotiFinder (Gutman, 2005), Block Maker (Henikoff, 1995), and DiliMoT (Neduva, 2005). However, none of these tools provides the specificity needed to search for patterns of basic and non-basic amino acids, determine the composition of amino acid patterns among a set of proteins, and, ultimately, determine if a particular protein is heparin-binding. This project is aimed at creating such software.

3. ARCHITECTURE

3.1 High Level Design

The objective of this research is to develop a software program that discovers certain patterns of amino acid substrings in a protein that indicate said protein is likely to bind to heparin, and furthermore, to give the amino acid composition of the substrings so that they may be synthesized and tested in the laboratory.

This program can be divided into three primary problems: linear motif matching, composite matrix formation, and contiguous patterns discovery. Before these problems can be addressed, however, some initial processing must be done. Lastly, there is also a Web interface for accessing the linear motif and composite matrix output online.

3.2 Processes

The following sections describe the different problems this program attempts to solve.

3.2.1 Initial Processing

The program has two inputs: a pattern file and a protein file. The pattern file contains the list of patterns to be searched for in the proteins, and the protein file contains the list of proteins to be searched. The patterns in the pattern file are simply sequences of "X" and "B" one to a line. While the protein file is in FASTA format.

In this stage, the program reads in and stores the patterns and proteins. It then converts each protein sequence into a parallel converted sequence of "X" and "B" depending on if each amino acid is in the set of X amino acids or B amino acids.

3.2.2 Linear Motif Matching

The linear motif matching problem takes as input a converted protein string $S = a_1a_2 \ldots a_n$ and a pattern string $P = p_1p_2 \ldots p_m$ where each element of P and S are over the alphabet, $\{X, B\}$. A substring $S_{sub} = s_1s_2 \ldots s_m$ of S, where m is the length of the pattern, matches the pattern P if and only if $s_i = p_i$ where $1 \leq i \leq m$. The linear motif matching problem then stores the numeric location of each match found (i.e., for a protein string of length n it stores the start location s and end location e where $s < e \leq n$). Once all the matches of a pattern with a protein are found, an output page displaying the results is created. An example match location output is shown in Figure 1.

```
Protein:  1CO1:A|PDBID|CHAIN|SEQUENCE
Pattern:  XBX
Occurrences = 14
Matches = {[2,5], [16,19], [18,21], [21,24], [28,31], [56,59], [63,66], [68,71], [73,7
, [77,80], [80,83], [82,85], [105,108], [112,115]}

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
E L V R T D S P N F  L  C  S  V  L  P  T  H  W  R  C
- - X B X - - - - - - - - - - - - X  B  X  B  X  ]
```

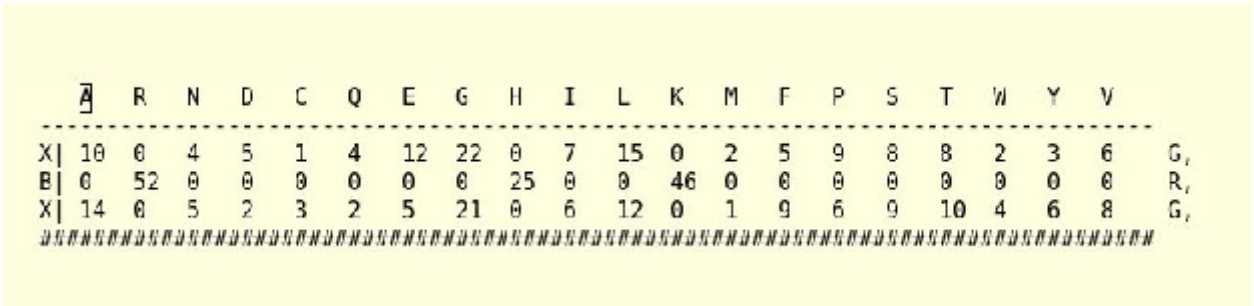
Figure 1: Match Location Output

3.2.3 Composite Matrix Formation

For each pattern there will be a set of matches, S *such that* $0 \leq |S| \leq n(length(n))$ where n is the number of proteins. The composite matrix formation problem looks at each match and determines which amino acid each X and B most commonly correspond to. It does this by searching through each match with each protein and keeps a tally of how many times a character

in the pattern corresponds to a particular amino acid and outputs the top 10% along with a matrix. For example, in the pattern XBX, the first X may correspond to, say, alanine 9 times and valine 2 times. The B may correspond to arginine 11 times and histidine and lysine 0 times, while the last X may correspond to serine 5 times and asparagine 6 times. The top amino acids for this pattern are alanine, arginine, and serine/asparagine respectively. The result of this computation is used in the lab to synthesize the peptide that results from the top occurring amino acids for each pattern and test if they in fact will bind to heparin.

The composite matrix *M* for a pattern $P = p_1 p_2 \ldots p_n$ has *n* rows for each element of *P* and 20 columns for each natural amino acid. An interior element $M(i, j)$ $(0 \leq i \leq n-1,\ 0 \leq j \leq 19)$ is the tally for an element of *P* with an amino acid. An example composite matrix is shown in Figure 2.



Firgure 2: Composite Matrix Output

3.2.4 Contiguous Patterns Discovery

Certain patterns of amino acids are known to have a particular affinity for binding to heparin, but not all of these are known. In general, the longer the pattern is, the greater its bonding potential is. The goal of the contiguous patterns discovery is to find all the patterns whose matches within a protein are directly adjacent to one another (could be two or more

adjacent patterns). These new patterns will be both longer and comprised of shorter patterns that have a known potential to bind with heparin. The hopes are that these new, longer patterns will have an even greater potential to bind with heparin.

The algorithm used for this phase looks at the start and end positions of each match in a set of matches for a protein using two nested for loops. The outer loop loops through a list of sorted intervals (start and end positions for the matches). At the end of each iteration, it adds the current interval to the list of intervals for the inner for loop (initially empty). The inner loop maintains a list of intervals whose end value is strictly greater than the start value of the outer loop (it is not possible to concatenate them otherwise). Once the outer loop's interval's start value is greater than the inner loop's end value, the current interval from the inner loop is removed for memory efficiency.

A match is found when the start value of the outer loop interval is equal to the end value of the inner loop interval. The new interval is then added to the list of intervals for the inner loop, so that it may be checked for further concatenations. Before each interval is removed from the inner loop list, its Boolean flag indicating whether it is a concatenation is checked. If it is a concatenation, then it is output.

The problem with this algorithm is that it produces many duplicates. These duplicates are divided into three (not mutually exclusive) classes: same-position duplicates, intra-protein duplicates (includes same-position duplicates), and inter-protein duplicates.

Every time a concatenation is found, the end value of the inner loop interval is equal to the begin value of the outer loop interval. The same-position duplicates can be removed by ensuring that the same begin and end values do not occur twice in a row, since these begin and end values correspond to the exact same substring of the protein string.

Intra-protein duplicates occur when two different patterns form the same concatenation. (Note that the same-position duplicates are subset of the intra-protein duplicates). For example, say we have patterns *p1, p2, p3,* and *p4*. The resulting concatenation of *p1p2* could be equal to *p3p4* (e.g. *p1* = XB, *p2* = BX, *p3* = X, *p4* = BBX then *p1p2* = XBBX = *p3p4* = XBBX). Even more simply, the concatenation *p1p2* may occur twice within a single protein. (Note that same-position duplicates are in fact a subset of intra-protein duplicates). These duplicates can be removed by adding all new concatenations to a set within the method. Then, each time a new concatenation is found, it is added to the set which does not allow duplicates.

The goal of the contiguous pattern discovery sub-problem is to output a list of new patterns that may be of importance. It is often the case that same concatenation of patterns is found in different proteins. For example, the concatenation *p1p2* may be found in protein1 and the concatenation *p3p4* may be found in protein2. If *p1p2* = *p3p4*, then the resulting concatenation will be duplicated in the output. This problem can be ameliorated by maintaining a set for the total patterns. Since the contiguous pattern discovery method is run for each protein, this set must be external to the method. Instead of simply outputting a new concatenation, it is added to the set of total patterns. The resulting set of total patterns is then written to a file or standard out for processing or studying duplicate-free.

3.3 Implementation

A Web interface was created as means to circumvent the burdens of compiling the program for different machines. The user uploads his or her protein and pattern files, and the resulting output is displayed on the screen. Using the Web interface, only the match location

output and composite matrix output are available. Also, due to bandwidth issues, use of the Web interface is restricted to smaller files.

This program was created using C++ and compiled using g++. It was run and testing on a computer running Linux 3.7.9-1. The Web interface was created using PHP and HTML and runs on an Ubuntu Server using Apache 2.2.14.

4. METHODOLOGY, RESULTS, AND ANALYSIS

The results of this program have yet to be tested on a chemical level. Thus far, they have only been verified for algorithmic correctness by comparing the program results to the input files and determining if the program's results match with the human's results.

4.1 Methodology

Amino acid patterns from a family of proteins called fibroblast growth factors (FGF), which are known to bind strongly to heparin were used for testing. Nineteen strings in total were selected. After the program's results were verified to be correct, the program was run using the candidate patterns and the FGF proteins to get a total count.

4.2 Results

The results of the frequency analysis for the candidate patterns mentioned above and a group of known heparin-binding proteins is shown in Figure 3.

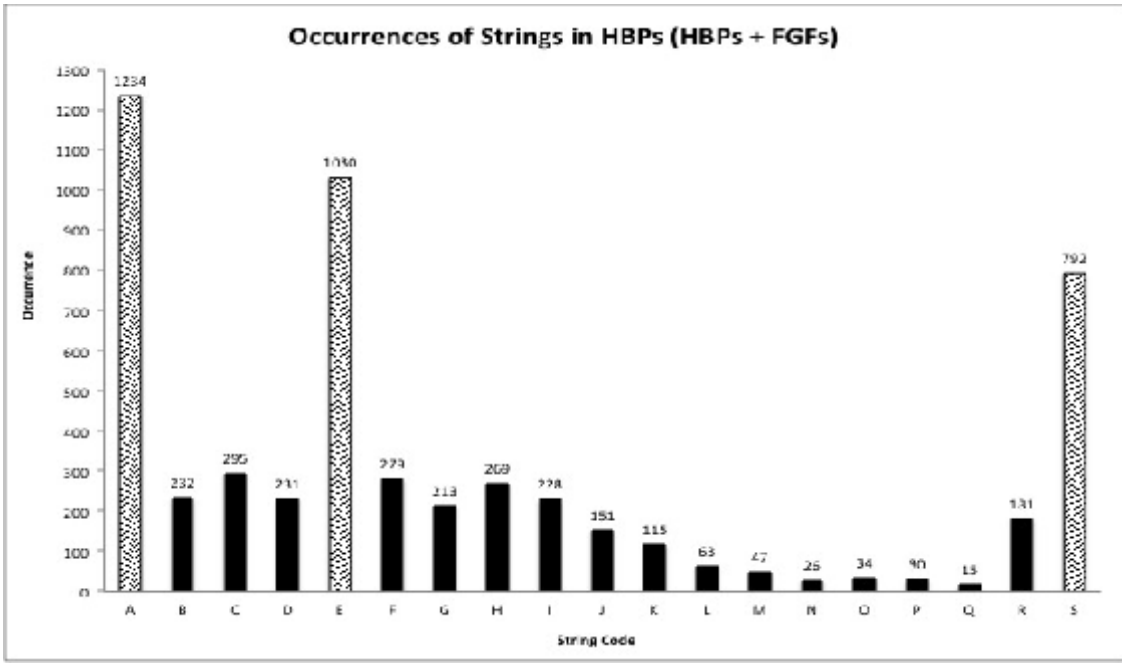


Figure 3: Heparin Binding Pattern Frequency Analysis

The X-axis (A-S) is the candidate patterns. The top three are XBXBX (A) at 1234, XBXXBX (E) at 1030, and XBXXXBX (S) at 792. The significantly greater number of occurrences of these patterns suggests that they may be important for heparin-binding proteins.

4.3 Analysis

This counting of patterns was previously done manually and, consequently, both tedious and prone to human error. The results of this program can now be used to find the counts of patterns with minimal effort.

5. CONCLUSIONS

5.1 Summary

Heparin is widely used and is being studied more and more lately particularly for the role it plays in preventing thrombosis. The key element of this study is finding patterns in proteins that imply that they possess an affinity to bind to heparin. With the millions of known proteins in existence, chemically testing each of them in the lab is infeasible. Thus, computer programs are necessary to facilitate this particular area of study.

This paper has detailed a program that finds sequences of non-basic/basic amino acids within protein strings, a process that was previously done manually. The program also returns a suggestive way to synthesize these peptide strings by providing the most commonly matched amino acids for each element in the pattern. Lastly, it finds contiguous sequences of patterns. These new sequences may have been undiscovered and could prove very useful in creating ideal peptides which strongly bind to heparin.

5.2 Potential Impact

This project has replaced a manual method of finding amino acid patterns in proteins with a programmatic one, greatly accelerating heparin research. After proper chemical analysis, this program could make it vastly easier to identify potential heparin-binding proteins, and, in turn, save people's lives by providing more effective heparin contaminant filtration methods.

Generalizing these algorithms to other proteins and other patterns could have far-reaching results and allow many new relationships between proteins to be discovered.

5.3 Future Work

There are several modifications that could be made to this program. First, many of the algorithms could be made more efficient. The naïve approach was taken for both the match location output and for the composite matrix production for faster prototyping and because it proved sufficient. One method for improving efficiency would be to parallelize these two algorithms. Also, using a hash for shifting substring algorithm such as the Rabin-Karp algorithm (Karp, 1987) for searching for patterns within the protein strings could also improve runtime.

Future work also lies in generalizing the overall program. Currently, the program is specific for searching for patterns of non-basic/basic amino acids. It would be ideal to let the user choose the type of pattern he or she wants to search for. Since the characters of the pattern are essentially sets, it would be possible to allow the user to first, select how many sets he or she needs (i.e. will the patterns be binary, tertiary, quaternary, etc.) and second, which amino acids go into which set. This would allow the program to be adopted for new problems with new proteins.

Lastly, due to time constraints, much of the chemical analysis phase (such as synthesizing the peptides returned from the program and checking if they indeed bind to heparin) has not been completed yet. The validity of the program's results have only been verified on a programmatic level, and it remains to be seen if the results will hold true in the lab. If not, then unforeseen modifications or adjustments may need to be made to the program.

REFERENCES

[1] Timothy L. Bailey, Mikael Bodén, Fabian A. Buske, Martin Frith, Charles E. Grant, Luca Clementi, Jingyuan Ren, Wilfred W. Li, William S. Noble, "MEME SUITE: tools for motif discovery and searching", *Nucleic Acids Research*, 37:W202-W208, 2009.

[2] R. Gutman, C. Berezin, R. Wollman, Y. Rosenberg, N. Ben-Tal, "QuasiMotiFinder: protein annotation by searching for evolutionarily conserved motif-like patterns", *Nucleic Acids Research*, 33:W255-W261, 2005.

[3] Henikoff, S., Henikoff, J.G, Alford, W.J, and Pietrokovski, S., "Automated construction and graphical presentation of protein blocks from unaligned sequences", *Gene* 163:GC17-26, 1995.

[4] Neduva V, Russell RB, "DILIMOT: Discovery of Linear Motifs in Proteins," *Nucleic Acids Research*, 34:W350-5, 2006.

[5] Karp, Richard M.; Rabin, Michael O., "Efficient randomized pattern-matching algorithms," *IBM Journal of Research and Development*, 31, 1987.