

# **Computational Detection of CPE Elements Within DNA Sequences**

**Report dated 19 July 2006**

**Author:**

**Ashutosh Koparkar**

**Graduate Student, CECS Dept., University of Louisville, KY**

**Advisor:**

**Dr. Eric C. Rouchka**

**Director, Bioinformatics Laboratory, University of Louisville, KY**

**Koparkar, Ashutosh\***

(Graduate Student, CECS department, University of Louisville, KY)

Submitted: 19 July 2006

\* Address for communication: [a0kopa01@louisville.edu](mailto:a0kopa01@louisville.edu)

---

## **Abstract**

CPE Stands for Cytoplasmic Polyadenylation Elements (CPE), these elements are known to have an active role in temporarily activating mRNAs. However the exact mechanism by which specific mRNAs are temporally activated is unknown it is important to study sequences that contain the CPE's. Biologists, biochemists always want to search some subsequence in a DNA sequence, for these searches and matching patterns resulting from these searches lead to better understanding of particular elements and there behavior.

This project focuses on creating a web based application that can be accessed by multiple user at various location. This web based application in conjunction with a local relational database of sequence provides an easy, reliable, reusable tool to aid in researching the CPEs as well as other elements. The primary objective of the tool thus created is to build a robust knowledge base within a local database that can be constantly updated with latest sequence information and the location of CPEs. This tool thus created must also provide a way to analyze stored sequences along with CPEs.

---

## **Introduction**

Since CPE detection along with identification of other strands are the focus of an increasing number of research efforts the requirement of an efficient tool that can provide localized persistence along with a capability to search these stored sequences arises. At present most of the researches researching in this field depend on running their search queries on various websites. Also the results obtained by the searches performed are not easy to manage as they need to be stored locally in either text files or excel sheets. While searching and managing the results is a tedious task, analyzing various results obtained from these searches is more troublesome.

There are multiple web based sequence search programs available the need for a localized tool was felt when it became difficult to manage results obtained by running the queries on remote sites. The queries run on the remote web sites take time sometimes days to return back results as most of the queries for sequence searches are performed on the complete database resulting in longer processing times. Also these queries tend to return results that sometimes may not be relevant to what the user intended. Managing results received from various search engines is often a very tedious process as the results received are formatted differently by different sites. Precious time that should have been invested in doing the actual research is wasted while performing searches at various sites and managing and analyzing results.

If a localized database of sequences that interest a researcher can be created in such a way that all the inherent intricacies are hidden from the user by way of a friendly user interface, it is hoped that such a database will be helpful in furthering the study and analysis of the sequences in a better manner. The Seq-Search web application introduced in this project does exactly that. This web based application hides all the details of storing, managing and analyzing sequences and searches from the user by providing a local database. The Seq-Search web application uses the data selectively saved by the user to perform requested tasks.

Indeed, time constraints on this project have meant that there is still a lot of scope for improvement in the web application. However, the procedures implemented and the work done up to this point will certainly work as a stepping stone for similar and more ambitious projects aiding the study of DNA sequences for CPE and various other strands.

It must be noted here that this project would not have been possible without the active guidance and support of Dr. Eric Rouchka, director of the Bioinformatics Lab at the University of Louisville.

## **The “Seq-Search” Web Application**

The Seq-Search web application presented in this project will provide the users of the application a robust user friendly way to save sequences, perform searches for CPE elements, filter results for easy viewing by the user in the form of report and analyze the results. Even though the primary goal of building this web application was to provide support in the research activity of CPE elements the web application can be used to search for strands that are unrelated to CPE searches.

The solution provided in the web application is java based and uses MySql as its choice of database for persistence. Web server of choice for this application is Tomcat 5.0. It is often very difficult to put all the pieces of a web application together to deploy web applications, to meet this challenge an automated build process is implemented using Maven 1.0 that on issuance of one command completely builds the web application starting with compiling the classes and deploys the newly built web application on to the Tomcat server. For the user interface part the web application uses MVC architecture that uses implementation provided by Struts. All the view components are implemented using Java Server Pages, using struts Action controller and action classes to complete the model. Along with the Struts framework Tiles framework is used to maintain a consistent look and feel all along the application for a user friendly environment. On the backend Spring IOC framework is used in conjunction with Hibernate and standard DAO's are used to provide business logic and database access.

As the web application depends on local database the onus of furnishing the data falls of the users of this application. Users can store sequences that they are interested in by entering them manually or uploading them using files stored in FASTA format. The FASTA format is one of the widely used data formats used to store sequence information. The in built parser provided with the web application parses the file provided by the user and extracts information such as gene sequence id, description of the sequence and the sequence itself to store into the local database. The program also provides a way for the user to input sequences manually by typing in the relevant information. User can then use the information stored in the database to perform various queries and analyze the results. The web application also provides a way for the user to perform searches on these saved database to identify CPE or any other elements, the search results thus received can be stored for future use. All the information about the sequences and the patterns found in those sequences is available to the user in the form of reports. All of the reports except user criteria to filter results based on user requirement.

It is hoped that the Seq-Search web application will provide an easy way for the researchers at the bioinformatics department to store, manage and analyze DNA sequences. The solution provided by this web application can be easily extended to implement any new features that may be required by the researching community after reviewing the initial product.

## **Technology/Tools**

### **Java Server Pages 1.2/ JSTL**

Java Server Pages commonly known as JSP's is an extension of Sun Microsystems Java Servlet technology. Java Server pages provide a developer an efficient and easy way to dynamically build web pages using dynamic content. JSTL is commonly known as the Java Server Pages Standard Tag Library and provides functionality in the form of tags that is commonly used while writing pages that use dynamic content.

In this project the JSP pages provide the building block of all the view components. In this project JSP's are exclusively used in conjunction with JSTL to generate dynamic content based on user selection. JSTL tags both core and formatting are used throughout the application to provide easy access to common tasks.

### **Struts MVC framework 1.1**

Apache struts is a free open source framework for creating Java applications using the Model View Controller (MVC) architecture. This type of architecture provides separation of functional areas in Model that deals with business logic, the view that represents user interface and the controller that represents the navigation between different pages. The struts project provides the basic framework required that developers can build on in order to avoid implementing their own MVC architecture. Using the Struts framework allows the developer to concentrate on the task of developing the application rather than worrying about implementing the architecture.

This project is completely based on the struts framework. The view is independently developed using JSP and JSTL while the front controller provided by the struts framework is used to provide navigation between user selections. For Implementing the Model part Spring application framework is used in conjunction with the struts framework that is described in the next topic of this report.

### **Spring/ Hibernate**

Spring is one of the most popular application development framework, that provides a full featured support for building java applications. The spring framework builds on the functionality provided by the J2EE and enhances that functionality without the tedious peculiarity of implementing a full J2EE solution. The framework effectively manages the middle tier objects and eliminates the need of multiple configuration files, simplifying configuration through features like Inversion of Control and Dependency Injection.

This project does not fully use spring instead spring is used in conjunction with the Struts framework to provide backend support to manage JDBC database connections and business object. The flexibility thus provided makes the application highly flexible in terms of database usage and portability. Hibernate is inherently used by the spring framework to provide database persistence support.

## **JUnit**

JUnit is an open source regression testing framework that allows developers to test their code before deploying it to the application servers. The framework was initially created for java based solutions, due to its popularity it is now ported to other languages as well.

In this web application it is very difficult to test the business logic and that is where JUnit helps out the most. Test cases written using JUnit provide a definitive answer to whether or not the implementation should be deployed on the server. All the business methods implemented in the web application have an associated JUnit test case.

## **Maven 1.1**

Maven is an open source software project management and comprehension tool that is based on Project Object Model. Maven allows developer a unique solution to most of the build and deployment problems in java environment. Maven makes the build process easy by predefining project dependencies, structures and a uniform build strategy.

This project uses the features provided by Maven to build and deploy the web application. Using Maven allows for easy dependency management and a transparent build system that with a single command builds the complete web application from scratch, cleans the deploy directory in the tomcat web server and deploys the new application.

## **Tomcat 5.0**

Apache Tomcat is an open source servlet container that provides a convenient and robust way to deploy web application. Apache tomcat container is also used in the official reference implementation of the Java Servlet and Java Server Pages Technology.

This project used Apache Tomcat to serve as primary host for the web application.

## **MYSQL**

The object of this project was to create a relational database of sequences, pattern matches and match results. At this stage, any standard relational database software would have sufficed. As such, the latest version, version 5.0, of the open-source MySQL database server was installed on the windows desktop computer. Version 5.0 of MySQL provides several enhancements over the previous versions, including support for running stored procedures storing large amount of text data and regular expressions. Although there are some experimental aspects in this current version, they do not come into play when storing and querying the data for the current project. There is however an issue with creating columns of type auto-increment with a primary key definition. Since this is a well known issue it is expected to be fixed in one of the future releases.

The advantages of using MySQL at this stage lie in the open-source nature of the software as well as the relatively lower level of complexity involved in installing and using MySQL as opposed to the popular commercial databases such as Oracle or Sybase.

### **Development /Test Environment**

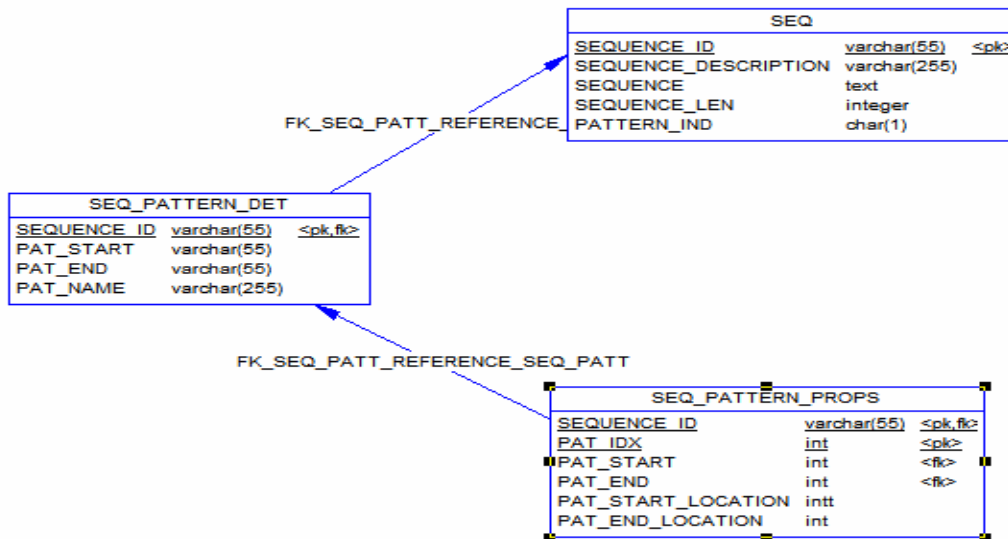
The bulk of the project is developed on a desktop with Intel Pentium 1.6 GHz processor. The system had a Java2 compliant Virtual Machine running in a Microsoft XP environment with 512 MB RAM. Disk space requirements will very depending upon size of the database, the test system had 100MB of available hard-disk space. The web interface is tested using Microsoft Internet Explorer 6.

## Implementation Details

### Database Schema

A schema is developed to provide all the persistence support for the Seq-Search application. The Schema is defined in the form of following three tables

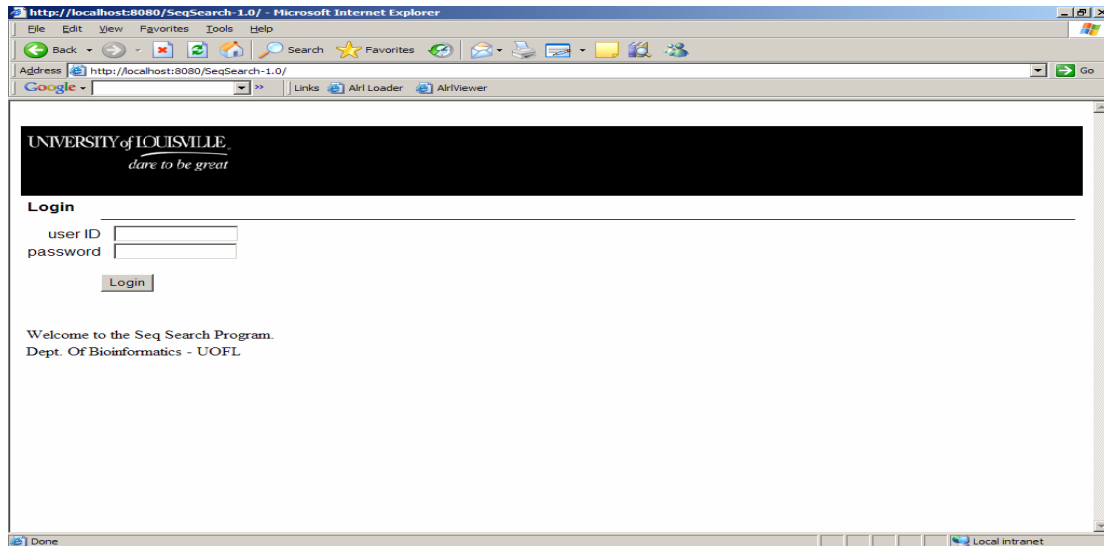
1. **SEQ**: This table holds a unique sequence id, description of the sequence, length of the sequence, a flag indicating whether or not that sequence has a known pattern and the sequence itself. All the uploaded sequence are maintained in this table.
2. **SEQ\_PATTERN\_DET**: This table provides storage for any patterns that may be associated with the base sequence. The information stored is the unique sequence id. String representing the start of the pattern and end of the pattern and name of the pattern. Sequence id along with the pat\_start and pat\_end fields defines the primary key for the sequences. The sequence id field has a foreign key reference on sequence id with the SEQ table.
3. **SEQ\_PATTERN\_PROPS**: This table is used to provide individual pattern information for each identified pattern in a sequence. The sequence id along with the pat\_idx defines the primary key for the sequence. and sequence id along with pat\_start and pat\_end columns define a foreign key relation with the SEQ\_PATTERN\_DET table. The pat\_start\_location and pat\_end\_location fields hold the start and end pattern location within the sequence.





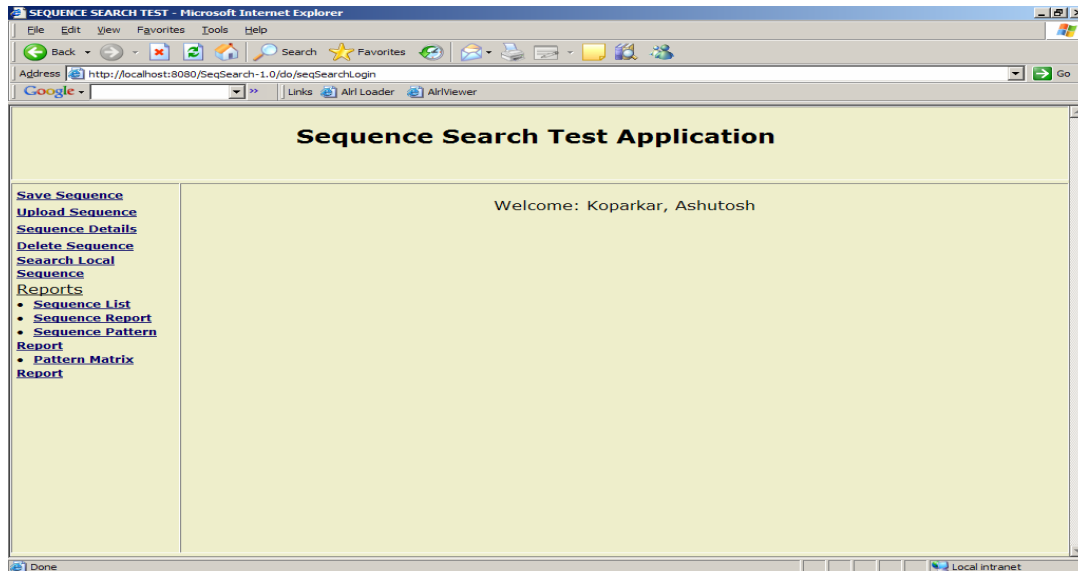
## User Interface – Login Screen

The login screen provides basic user id password combination login. User is presented with the login page at the beginning so that user credentials can be authenticated using information stored in the database. Upon success in authenticating the user is forwarded to the home page and if the authentication fails the user is asked to enter his/her credentials again for authentication.



## User Interface – Home Page

Upon successful authentication the user is forwarded to the home page, this page provides a welcome message to the user and provides all the navigation for performing a task.



## User Interface – Save A Sequence

User is navigated to the save a sequence screen upon selecting the save sequence link from the menu. This screen allows the user to save an sequence manually into the database. Once all the information is entered and submitted by the user, the sequence id is checked for possible duplicate sequences if there is no sequence in then the information entered by the user is saved into the database along with any known pattern user might have provided. If the sequence id entered by the user is already present an error message is displayed to the user indicating that there is already a sequence present in the database with the same sequence id.

The screenshot shows a web form titled "Save A Sequence". It contains the following fields and elements:

- Sequence Id (GenBank):** A text input field containing "313432391".
- Description:** A text input field containing "Bos taurus lactoglobulin, beta (LGB), mRNA".
- Sequence:** A large text area containing a multi-line DNA sequence: "ACTCCACTCCCTGCAGAGCTCAGRAGCCTGATCCCGGCTGCAGCCATGAGTGCCTCTGCTTGCCTGGCCCTCACCTGTGGCCCCAGGCCCTCATCGTCACCCAGACCATGAAGGGCCTGGATATCCAGAAGGTGGCGGGACTTGGTACTCCTTGGCCATGGCGCCAGCGACATCTCCCTGCTGGACGCCAGAGTGCCCCCTGAGAGTGTATGTGGAGGAGCTGAAGCCCCACCCCTGAGGGCGACCTGGAGATCCTGCTGCAGAAATGGGAGAATGATGAGTGTGCTCAGAAGAAGATCATTGCAGAAAAAACCAAGATCCCTGCGGTGTTCAGATCGATGCCTTGAACGAGAACAAAGTCCTTGTGCTGGACACCGACTAC".
- Pattern(Y/N):** A checkbox labeled "Y" is checked.
- Start:** A text input field containing "ACTCCA".
- End:** A text input field containing "TGGTAC".
- Pattern Name:** A text input field containing "ACTCCA to TGGTAC".
- Save:** A blue button labeled "Save".

## User Interface – Upload Sequence

User is navigated to the upload sequence screen upon selecting the upload sequence link from the menu. This screen works the same as the save sequence other then that it allows the user to save a sequence by selecting a file in Fasta format from the local file system. The duplicate check is performed the same way as in the save sequence screen.

The screenshot shows a web form titled "Upload Sequence File". It contains the following fields and elements:

- Please Select A File:** A text input field containing the file path "D:\Documents and Settings\KoparkarAM\My Documents\EDU\CECS-660\fastafiles\Copy (". To the right of this field is a "Browse..." button.
- Pattern(Y/N):** A checkbox labeled "N" is unchecked.
- Start:** An empty text input field.
- End:** An empty text input field.
- Pattern Name:** An empty text input field.
- Upload File:** A blue button labeled "Upload File".

## User Interface – Sequence Details

User is navigated to the sequence details screen upon selecting the sequence details link from the menu. This screen allows the user to view detailed information for any given sequence. The information includes sequence id, description, the sequence, any patterns associated to the sequence along with the information about each occurrence of the pattern and its location within the sequence.

Please Enter Sequence Id:

Sequence Id	Description	Length	Pattern (Y/N)
1916261	CLU86661 Canis latrans gamma fibrinogen gene, partial cds	581	Y

**Sequence**

MAAGATTGTAAGTAGTTTCATGTTCTCACTGCATGTGAAGTGGGAGGGGCGAGGAATATCATTGGCATTTTAAACTTTA  
 TTTTATTATTACTTATTTTCATTTRAACTTAAAAAACAATAATCGAGGTTTACCTGTTTTATTATGGATTTTAT  
 GCARGTTTTTTGTTTGTGTTGTTGTTGTTTTGAATCAAGATTCTAATCACTCTATTCTATTCTCAGGGTAGTTATTG  
 CCCGACTACCTGTGGAATTGCAGACTTCTGTYACTTACCAGACCGGAGTAGACAAATGATYTCAGGCTTGGAGACCTCCTA  
 AGACGAATTGAAAAAACAACAGCAGAAAGCCAAAGAAGTGAATCAATCCAATCACCTATAATCCCGATGAGCCACCAAAGC

Name	Start	Start Pos.	End	End Pos.	Length
AAATCA to CCCGA	AAATCA	382	CCCGA	406	24
AAATCA to CCCGA	AAATCA	392	CCCGA	406	14
ACTG to TCGA	ACTG	30	TCGA	135	105
AGACGAATT to TTGA	AGACGAATT	340	TTGA	347	7

## User Interface – Search Sequences

User is navigated to the search sequence screen upon selecting the search sequence link from the menu. A search operation is performed on the local database to search for the patterns entered by the user. If sequences are found that contain the selected patterns then they are displayed to the user. All those sequences that already have been identified to have the pattern specified by the user have their save checkbox option disabled while those that don't have an option to save identified pattern by selecting the checkbox and giving the pattern a name.

Please Enter Start Pattern:  End Pattern:

Sequence Id	Pattern Name	Start Pattern Pos.	End Pattern Pos.
<input checked="" type="checkbox"/> <a href="#">1916261</a>	ACTG to TCGA	31	136
<input checked="" type="checkbox"/> <a href="#">1916275</a>	ACTG to TCGA	42	278
<input checked="" type="checkbox"/> <a href="#">1916279</a>	ACTG to TCGA	42	244

Local intranet

## User Interface – Reports- Sequence List

User is navigated to the sequence list report upon selecting the sequence list report link from the menu. This is a very basic report that provides information about all the sequences stored in the database. The sequence id for each sequence displayed on the list can be clicked which navigates the user to the sequence details screen allowing the user to view all the extended details about the selected sequence.

Sequence Report			
Sequence Id	Description	Length	Pattern (Y/N)
<a href="#">1916253</a>	AAU86648 Alces alces gamma fibrinogen gene, partial cds	546	N
<a href="#">1916255</a>	BPU86653 Balaenoptera physalus gamma fibrinogen gene, partial cds	538	N
<a href="#">1916257</a>	CCU86660 Crocuta crocuta gamma fibrinogen gene, partial cds	560	N
<a href="#">1916259</a>	CDU86657 Camelus dromedarius gamma fibrinogen gene, partial cds	552	Y
<a href="#">1916261</a>	CLU86661 Canis latrans gamma fibrinogen gene, partial cds	581	Y
<a href="#">1916267</a>	GCU86649 Giraffa camelopardalis gamma fibrinogen gene, partial cds	546	N
<a href="#">1916269</a>	HLU86654 Hexaprotodon liberiensis gamma fibrinogen gene, partial cds	540	N

## User Interface – Reports- The Sequence Report

User is navigated to the sequence report criteria page upon selecting the sequence report link from the menu. User can select sequence id, description, max or min length of the sequence or pattern to filter out the results. Once user selects the run report button after providing one or more criteria options the query is executed and the results along with the criteria selected by the user are displayed.

Sequence Report Criteria			
Sequence Id	<input type="text"/>	Description	<input type="text" value="taurus"/>
Min. Length	<input type="text"/>	Min. Length	<input type="text"/> <input checked="" type="checkbox"/> Pattern
<input type="button" value="Run Report"/>			

Report Criteria			
<b>Sequence Id</b>	:		
<b>Description</b>	: gamma		
<b>Min. Length</b>	: 0		
<b>Min. Length</b>	: 0		
<b>Pattern</b>	: N		
Sequence Id	Description	Length	Pattern
<a href="#">1916253</a>	AAU86648 Alces alces gamma fibrinogen gene, partial cds	546	N
<a href="#">1916255</a>	BPU86653 Balaenoptera physalus gamma fibrinogen gene, partial cds	538	N
<a href="#">1916257</a>	CCU86660 Crocuta crocuta gamma fibrinogen gene, partial cds	560	N
<a href="#">1916267</a>	GCU86649 Giraffa camelopardalis gamma fibrinogen gene, partial cds	546	N
<a href="#">1916269</a>	HLU86654 Hexaprotodon liberiensis gamma fibrinogen gene, partial cds	540	N
<a href="#">1916271</a>	ODU86647 Ovis dalli gamma fibrinogen gene, partial cds	523	N
<a href="#">1916273</a>	PCU86652 Physeter catodon gamma fibrinogen gene, partial cds	538	N
<a href="#">1916281</a>	TNU86650 Tragulus napu gamma fibrinogen gene, partial cds	524	N

## User Interface – Reports- Sequence Pattern Report

User is navigated to the sequence pattern report criteria page upon selecting the sequence pattern report link from the menu. User can select sequence id, pattern name, start or end pattern to filter out the results. Once user selects the run report button after providing one or more criteria options the query is executed and the results along with the selected criteria options are displayed.

### Sequence Pattern Report Criteria

Sequence Id	<input type="text"/>	Pattern Name	<input type="text"/>
Start Pattern	<input type="text" value="AAATCA"/>	End Pattern	<input type="text"/>

### Sequence Pattern Report

**Report Criteria**

Sequence Id :	
Pattern Name :	
Start Pattern :	AAATCA
End Pattern :	

Sequence Id	Description	Pattern Name	Start Pattern	End Pattern
<a href="#">1916259</a>	CDU86657 Camelus dromedarius gamma fibrinogen gene, partial cds	AAATCA to CCCGA	AAATCA	CCCGA
<a href="#">1916261</a>	CLU86661 Canis latrans gamma fibrinogen gene, partial cds	AAATCA to CCCGA	AAATCA	CCCGA
<a href="#">1916275</a>	PTU86656 Pecari tajacu gamma fibrinogen gene, partial cds	AAATCA to CCCGA	AAATCA	CCCGA

Local intranet

## User Interface – Reports- Pattern Matrix Report

User is navigated to the pattern matrix report criteria page upon selecting the sequence pattern report link from the menu. User must provide a sequence id to run the report. Once user selects the run report button after providing the sequence id the query is executed to get the selected sequence id and all the patterns associated with that sequence id. A position matrix is built for all the patterns returned by the query. The result for this report displays the individual sequence along with all the patterns associated with the sequence along with position matrix for saved start and end patterns.

**Sequence Pattern Report Criteria**

Sequence Id

**Report Criteria**  
Sequence Id : 1916261

Sequence Id	Description	Pattern Name	Start Pattern	End Pattern
<a href="#">1916261</a>	CLU86661 Canis latrans gamma fibrinogen gene, partial cds	AAATCA to CCCGA	AAATCA	CCCGA
<a href="#">1916261</a>	CLU86661 Canis latrans gamma fibrinogen gene, partial cds	ACTG to TCGA	ACTG	TCGA
<a href="#">1916261</a>	CLU86661 Canis latrans gamma fibrinogen gene, partial cds	AGACGAATT to TTGA	AGACGAATT	TTGA

**Start Pattern Matrix**

<b>A</b>	3(1.0)	1(0.33)	2(0.66)	0(0.0)	0(0.0)	2(0.66)	1(0.33)	0(0.0)	0(0.0)
<b>T</b>	0(0.0)	0(0.0)	1(0.33)	1(0.33)	0(0.0)	0(0.0)	0(0.0)	1(0.33)	1(0.33)
<b>C</b>	0(0.0)	1(0.33)	0(0.0)	1(0.33)	1(0.33)	0(0.0)	0(0.0)	0(0.0)	0(0.0)
<b>G</b>	0(0.0)	1(0.33)	0(0.0)	1(0.33)	1(0.33)	0(0.0)	0(0.0)	0(0.0)	0(0.0)

**End Pattern Matrix**

<b>A</b>	0(0.0)	0(0.0)	0(0.0)	2(0.66)	1(0.33)
<b>T</b>	2(0.66)	1(0.33)	0(0.0)	0(0.0)	0(0.0)
<b>C</b>	1(0.33)	2(0.66)	1(0.33)	0(0.0)	0(0.0)
<b>G</b>	0(0.0)	0(0.0)	2(0.66)	1(0.33)	0(0.0)

## Business Logic – The DAO & Interface

To provide the web application with all the relevant information retrieved from the database. All the business logic required to process client requests is implemented using one independent DAO that provides easy access to the database tables by executing custom queries. The DAO definition is displayed below:

```
public interface SequenceServiceDAO {
    //Logger LOGGER = Logger.getLogger(this class);
    public SequencePojo findSequence(String id) throws BioinformaticsException;
    public ArrayList findPatternsForSequence(String id) throws BioinformaticsException;
    public ArrayList findAllSequence() throws BioinformaticsException;
    public ArrayList findSequencesForSubstrings(String sub1, String sub2) throws
BioinformaticsException;

    public boolean saveSequence(SequencePojo pojo) throws BioinformaticsException;
    public boolean savePatternForSequences(SequencePatternPojo patternPojo, List sequences)
throws BioinformaticsException;

    public ArrayList getSequenceReport(String seqId, String seqDesc, int minLen, int maxLen,
String hasPattern) throws BioinformaticsException;
    public ArrayList getSequencePatternReport(String seqId, String patternName, String
patStart, String patEnd) throws BioinformaticsException;
}
```

The Spring framework provides the interface between the client framework implemented as struts class and the DAO along with defining database properties for the project.

```
<bean id="mySqlDataSource" class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName">
        <value>com.mysql.jdbc.Driver</value>
    </property>
    <property name="url">
        <value>jdbc:mysql://localhost:3306/bioinformatics</value>
    </property>
    <property name="username">
        <value>root</value>
    </property>
    <property name="password">
        <value>password</value>
    </property>
    <property name="maxActive">
        <value>50</value>
    </property>
</bean>
```

## **Conclusion**

The project provided the author with an opportunity to delve a little deeper into the areas of DNA sequence searching and analysis. Although the time frame for the project was fairly small, it did provide an opportunity to understand a great many of the issues involved in efforts on those aspects of genetic research.

Despite the limited scope of the project the following changes can be made in the future to increase the scope and usability of this project.

- User management module.
- Graphical representation of matrix created by the application.
- Additional reports per user requirement.

## **Acknowledgments**

- Dr. Eric Rouchka – Director, Bioinformatics Laboratory, University of Louisville, KY
- Pingle Wang – Post Graduate Student , University of Louisville, KY

## **References**

1. Paper by Charlesworth A, Cox LL, MacNicol AM  
[http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list\\_uids=14752101&dopt=Abstract](http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=14752101&dopt=Abstract)
2. Struts - <http://struts.apache.org/>
3. Spring - <http://www.springframework.org/>
4. Tomcat - <http://tomcat.apache.org/>
5. Maven - <http://maven.apache.org/>
6. JUnit - <http://www.junit.org>
7. NCBI Homepage: <http://www.ncbi.nlm.nih.gov/>
8. PubMed Central: <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Pmc>
9. FASTA format: <http://www.ncbi.nlm.nih.gov/BLAST/fasta.shtml>
10. MySQL Homepage: <http://www.mysql.com>