UNIVERSITY OF THE PHILIPPINES MANILA
COLLEGE OF ARTS AND SCIENCES

DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS

Prediction of Polyketide Product from Module Organization of Enzymes Using Cumulative Tanimoto Fragment Scores

A Special Problem in Partial Fulfillment
Of the Requirements for the Degree of
Bachelor of Science in Computer Science

Submitted by:
Mendoza, John Althom Aduna
2009-42070

April 2013

ACCEPTANCE SHEET

The Special Problem entitled "***Prediction of Polyketide Product from Module Organization of Enzymes Using Cumulative Tanimoto Fragment Scores***" prepared and submitted by **John Althom Aduna Mendoza** in partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Science** has been examined and is recommended for acceptance.

Ma. Sheila A. Magboo, M.S.
Adviser

EXAMINERS:

		Approved	**Disapproved**
1.	Gregorio B. Baes, Ph.D. (candidate)		
2.	Avegail D. Carpio, M.S.		
3.	Richard Bryann L. Chua, Ph.D. (candidate)		
4.	Aldrich Colin K. Co, M.S. (candidate)		
5.	Perlita E. Gasmen M.S. (candidate)		
6.	Vincent Peter C. Magboo, M.D., M.S.		
7.	Geoffrey A. Solano,Ph.D. (candidate)		
8.	Bernie B. Terrado, M.S. (candidate)		

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Avegail D. Carpio, M.S.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences and Mathematics

Marcelina B. Lirazan, Ph.D.
Chair
Department of Physical Sciences and Mathematics

Alex C. Gonzaga, Ph.D.
Dean
College of Arts and Sciences

ABSTRACT

Polyketide is a major class of natural products possessing several pharmacological properties. Performing wet laboratory experiments to discover a functional polyketide is costly and difficult because of its trial-and-error nature. However, the analogous biosynthesis of these metabolites to fatty acids makes the resulting compound predictable. Through the use of information technology, a stand-alone computational tool –Predyketide – is created to observe the resulting structure per elongation, and to allow prediction and visualization of the most possible natural product compound. The list of all known building blocks (starter and extender) used in the system is gathered from ASMPKS, another polyketide-related system. With these functionalities, this application can help in the discovery of new drugs requiring lesser time and effort.

KEYWORDS

Polyketide, Tanimoto coefficient, Modular synthesis, Domain sequence

Table of Contents

I. Introduction

A. Background of the Study

The 2007 review of drug sources presents that only 30% are purely synthetic drugs as shown in Figure 1. The remaining portions are either biologically manufactured, natural products, derivatives of natural products, or obtained from natural sources before being artificially synthesized. And for the last 25 years, natural products have been a significant source of new drugs. [1]

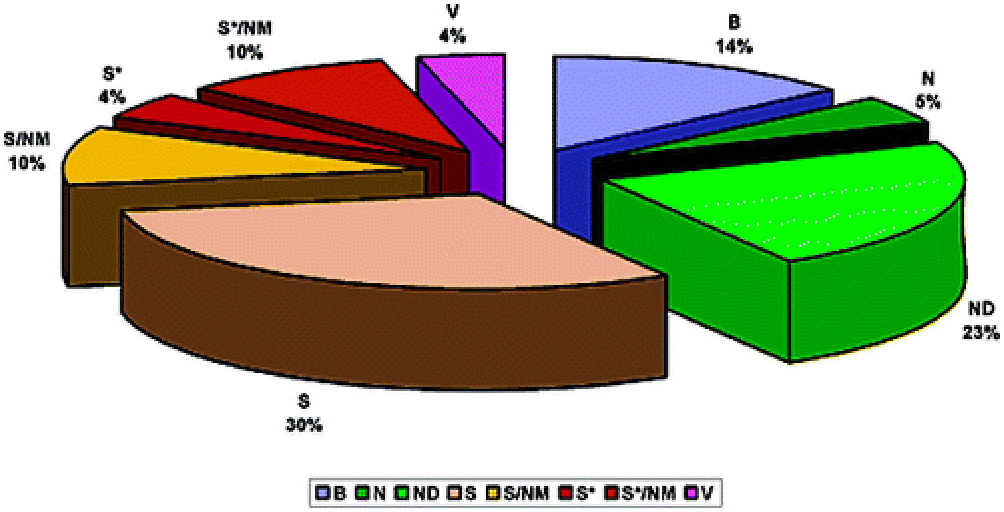


Figure 1 Sources of drug (2007)

One major class of natural products is polyketide. [2] Derived from bacteria, fungi, plants, insects, dinoflagellates, mollusks, and sponges, polyketides are structurally and functionally diverse class of secondary metabolites that provide a wide range of pharmacological activities such as antibacterial (e.g. erythromycin), anticancer (e.g. epitholene), anticholesterol (e.g. lovastatin), and immunosuppressant (e.g. rapamycin) properties. [2-6]

Analogous to fatty acid biosynthesis, these polyketides are being synthesized by sequentially catalyzed groups of enzymes called *polyketide synthases* (PKS). The only difference is that the former follows a fixed series of ketoreduction, dehydration, and enoyl reduction; and PKS undergoes even just a subset of these steps. The natural products produced after the succession of these processes are further classified into three – Type I, which consists of highly modular proteins; Type II, of mono-functional protein complexes; and Type III, of no acetyl carrier protein (ACP). [4, 7-8] Other than the mentioned classifications, these enzymes can further be grouped into two – modular, PKS with a non-iterative chain elongation process [9]; and linear, with an iterative one [10].

Polyketides dated back as early as 1893 at London University when James Collie studied the structure of *dehydroacetic acid* by boiling it with barium hydroxide. Instead of obtaining his desired result, the experiment yielded an aromatic compound – orcinol. [11] Since then, due to its importance in drug development, the staggering increase in the amount of published polyketide data demands computational tools such as database systems for data storage and retrieval, and polyketide structure determining tools. [12]

B. Statement of the Problem

Polyketide wet laboratory experiments are expensive [13]. In order to minimize the laboratory costs, the simulation before actual experimentation is performed; however, manually doing it is a tedious practice. So, computer-aided models may be helpful in this process. Despite the importance of polyketide research in development of new medicine, the existence of a single *natural product and polyketide structure determining systems* is scarce. Thus, it causes the scientists to access different systems for one single operation.

A (polyketide) scientist usually accesses chemical database systems for the collection of polyketide starter and extender units. The units gathered are then passed to a molecule structure drawing software, usually desktop applications (e.g. ChemDraw, ChemSketch). In some cases, the units retrieved from the databases are image files not readable by the aforementioned applications. The researcher is responsible for the recreation of the structure in the software. [14]

Although a number of polyketide computation analysis and database systems exist, several components including visualization and analysis, if not lacking, are not presented in a comprehensive and standard manner. Moreover, the existing systems which are found related to PKS deals with protein-protein interactions (PPI). PPI predictions only determine whether or not the engineered enzymes will be able to produce the desired product (i.e. it only provides the answer on whether two PKSs will be sequenced or not). [15]

A system capable of predicting a polyketide (PK) natural product (NP) structure given only the domain and module organization is still unavailable. An assembly system for modular PKS, the ASMPKS (Analysis System for Modular Polyketide Synthesis) [15], is close to the desired system; however, it lacks the actual natural product prediction. Moreover, the system is also closed source, preventing interested developers in the field from adding new functionalities.

The structure of a molecule is a useful tool in determining its pharmaco- and bioactivities. [16] Existing PKS and PK NP systems do not support the prediction of the structure. Researchers still have to use external, where some are proprietary, systems to determine such.

C. Objectives of the Study

The aim of this study is to create an open-source stand-alone desktop application – ***Predyketide*** – which allows the prediction of novel polyketide natural products given the enzyme sequence inputs, and an open-access web information system – ***Polykase*** – which provides a way to share PK NP predictions and contains a library of naturally occurring Type I PKS's. *Polykase* also provides functionalities for user and data moderation for the creation, deletion, or modification of PKS, PKNP, prediction, and user account records.

The specific objectives of this study are as follows:

1. *Predyketide* is a stand-alone desktop application that allows prediction of modular polyketide natural product given only the domain sequence of enzymes.

    a. To allow users to create a cartoon representation, similar to Figure 3, of the polyketide biosynthesis given only the enzyme or domain inputs on their respective work computers;

    b. To allow users to generate the structure predictions of the compound after the biosynthesis chain based on the NP-likeness score [17];

    c. To provide several molecular properties (e.g. weight, volume, aLogP) of each compound prediction;

    d. To allow users to export the workspace (polyketide chain) as a text file or as an image;

    e. To allow users to export the molecule prediction as a ZIP file containing the polyketide chain (as text and as image), the prediction's SMILES [18] and MolFile [19] representation format, the molecule's descriptors, and an image of the compound;

    f. To allow users to import a polyketide chain (in text format) for further chain elongation;

2. *Polykase* is a web application that allows registered users to share, download, and review predictions and to view known Type I modular PKNPs.

    a. To be able to provide forms and functions for the system administrator to allow the creation, modification, and deletion of PKNP, PKS, and prediction records;

    b. To be able to provide a review and ratings section on the web interface for each stored polyketide prediction, accessible to the registered users and moderated by the system administrator;

    c. To be able to create forms for guest users as a mean for account creation;

    d. To be able to create an interface for the system administrator to aid the polyketide data management;

    e. To provide the administrator a dashboard which displays all the pending account applications and all uploaded polyketide entries, and provides approval functionalities;

    f. To be able to provide functions allowing the administrator to add, edit, or delete a registered user's account and polyketide data.

g. To let the system administrator dump pre-existing records or data in the system gathered from ASMPKS; and

h. To be able to create a user interface for the structured and organized presentation of the stored data to all user levels to access this component.

D. Significance of the Study

Drug discovery has become a vital part of human existence. Knowing the fact that a number of therapeutics are derived from polyketides [20, 21], it is just proper to give this field more importance. Providing a way for recording and prediction of these natural products through *Predyketide* will help to lessen the repetition of work among the researchers, and to minimize the expensive and laborious laboratory experiments. The early detection of the structure and the organization of an undiscovered polyketide through predictive analysis will help to solve these problems. [22]

The significance of studying the interconnection of modular domains through linkers – short peptide sequences occurring between protein domains – has shown great importance in determining if a specific monomer can contribute to the growing polyketide [23, 24]. However, this means almost nothing in the drug development industry if the structure of the resulting natural product shows no promise of drug properties. Thus, it is important to determine the structure first through the use of the proposed system.

The stand-alone design of *Predyketide* allows the researchers to use the application even without internet connection. Unlike a similar system, ASMPKS [15], Predyketide works on the users' local computers and provides a way to save, export, and predict the possible PK NP structure.

These generated predictions can be uploaded and shared on a web interface which is a part of *Polykase*. The *Polykase* component also includes the reviews and ratings section for each novel polyketides where authorized users can help in the advancement of others' works by commenting on any discoveries about the prediction.

Moreover, this web component is also a reference of polyketide research materials since it includes a library of naturally occurring Type I modular polyketides. Instead of accessing pre-existing systems like ASMPKS [15] and PKSDB [25], the records are being duplicated to fuse both the prediction database and PKNP library into a single web domain. *Polykase* also includes a user management component, which allows the administrator to manage all the user accounts and polyketide data even without technical programming and search query backgrounds.

E. Scope and Limitations

This study focuses on the creation of a database and a prediction system for modular Type I polyketides. The records to be stored in the publicly accessible database will be gathered from the other existing polyketide systems [15, 25]. A single polyketide record contains the following: starter and extender units used, the enzyme where the PK was cultured (aka *host* *organism*) and where a PKS was obtained (aka *gene*), the chemical structure (after post-processing), the name of the natural product, and a short description about it. Other information not mentioned is not included. The chemical soundness of the applications is reviewed by a number of researchers in the field.

The source code for both systems will be available on a Sourceforge account upon completion to allow other interested developers to upgrade the existing systems. The executable Java Archive (JAR) file will be uploaded under the same account.

Predyketide, once downloaded, can be used by any user. However, it is assumed that the users are in the field of polyketide research to avoid system misuse. On the other hand, *Polykase* has three levels of user access: a guest user, a registered user, and an administrator. All user types can view the contents of the system, download the desktop application, and use it on their respective work computers. Only the registered users are given access to upload a predicted structure for the critique of fellow users. An administrator, a special kind of registered user, is responsible for the moderation of contents and registered users.

The only computational tool included in this project is the prediction of the natural product structure desktop application, *Predyketide*. A tool for PPI similar to ASMPKS' and others will not be included. Other than the database and the structure predictor components, related components are not included. *Predyketide* is capable of accepting domain sequences and generating a cartoon representation with a ranked list of predicted structure output.

Only the domain organization of the PK will be considered in the predictions. Some polyketides require a foreign macrolide structure before the last process of cyclization such as the rapamycin, which can be seen in Figure 2. It can be observed that before the rapamycin macrocycle is achieved, *pipecolate 27* is introduced to the structure using a pipecolate-incorporating enzyme. [4] This mechanism is not included in the system as well as other *post-processing* mechanisms or the introduction of external systems such as sugar.

Predyketide users can create any number of polyketide predictions that can be stored in his/her local repository (e.g. personal computer, work computers). However, a maximum of three predictions are allowed to be submitted in the web interface. This limitation prevents the possible database and web server congestion.

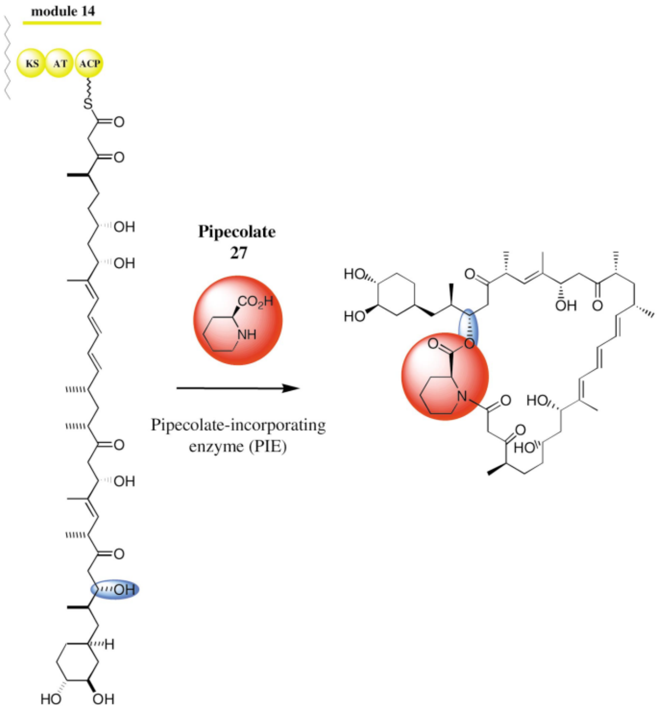


Figure 2 Formation of Rapamycin Macrocycle

Each stored prediction is available to other users for viewing and rating. In order to promote intellectual integrity, one prediction record is displayed together with the registered user who shares the sequence and the date of submission. However, all works or generated predictions are not annotated like in journals and not verified by wet lab experiments.

For the cartoon representation, each module is represented by a distinct column. The polyketide structure for every module is attached to the acetyl-carrier protein (ACP); PKS grouping is not included. The output cartoon is patterned from Figure 3. Also, a minimum of three (3) and a maximum of 15 modules (loading and terminating modules are not included) are allowed due to graphics and memory limitations. Also, the accuracy of the molecular structure is limited to the rendering capabilities of the libraries used – CDK [26] for the chain elongation and MarvinBeans [27] for the natural products.

Predictions and data storage for Type II and Type III PK's, though equally important, are not included due to the difference in the synthesis process from that of Type I [4, 28]. However, similar works or modifications in existing systems dealing with these two types can be done in the future.

On the other hand, despite the availability of the account creation form in public, only the administrator can approve a pending application. The said approval is subjected to the system manager's prejudice. The same is the case in a newly uploaded polyketide entry.

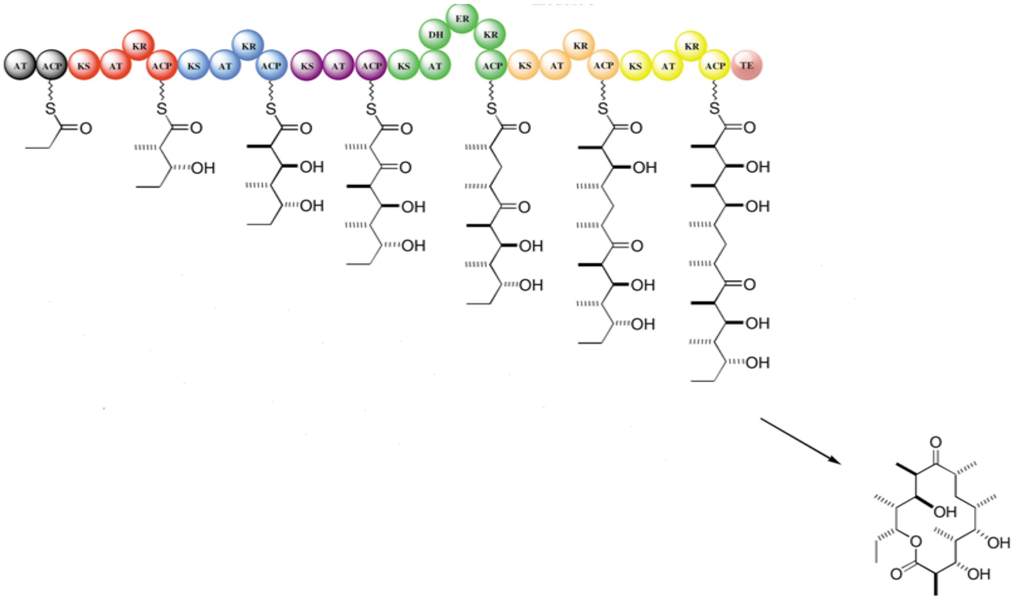


Figure 3 Erythromycin PKS Assembly

F. Assumptions

Several assumptions were made prior the development of this study:

a) The users of both *Predyketide* and *Polykase* are experts in area of polyketide or natural products research to avoid misusage;

b) The administrator should be knowledgeable in account and data management; and

c) The computer where *Predyketide* will be executed is installed with Java SE version 7.

II. Review of Related Literature

A. Polyketides

The study on polyketides can be traced back as early as 1893 when James Collie serendipitously discovered *orcinol*. [4, 11] Forty-five (45) years since this discovery, chemists were only able to isolate and discover new polyketides from fungi and bacteria. It was only in 1953 when mechanisms were initiated for the biosynthesis of polyketides. Polyketides are formed by the "*head-to-tail linkage of acetate units, followed by cyclization by an aldol reaction*", a synthesis similar to fatty acids. [29] During the mid-1980's, the application of recombinant DNA techniques allowed the analysis of genetic basis for polyketide production, giving way to the comprehension of *actinohordin(*from *Streptomyces coelicolor*) in both genetic and biological level. [4, 7]

PKS's share great similarities with fatty acid synthases (FAS) since both catalyze the condensation of *acetyl-CoA* and *metyl-CoA* to produce *ketoacetyl* linked to the enzyme by thioester bond. The only distinguishing characteristic between the two is that the former does not require ketoreduction, dehydration, and enoyl reduction to be all present in the process to form the polyketide; the fatty acid synthesis does. [30]

Ketoreduction is the process of reducing a ketone to an alcohol by using a ketoreductase. Dehydration involves the removal of water from the reacting molecule. Enoyl reduction hydrogenates the chain that results to a fully saturated methylene center. [31] Figure 3 illustrates how these processes are involved in the biosynthesis of erythromycin.

PKS are grouped into three (Type I, Type II, Type III) [4, 7-8] and classified according to mode of synthesis (linear or modular) [9, 10]. The distinction and other characteristics of each group and type are presented by Watanabe & Ebisuka (2004) on Frandsden (2010) in Table 1. [32, 33]

Group	Protein structure	Synthesis mechanism	Predictable	Resembles	Found in
Type I (modular)	Single protein with multiple modules	Linear (assembly-line style) in which each active site is used only once	Yes, to some extent.	na	Bacteria
Type I (iterative)	Single protein with one module	Iterative, in which the active sites are reused repeatedly	No	Vertebrate FAS	Fungi
Type II	Multiple proteins, each with a single mono-functional active site	Iterative, in which active sites may be used only once or repeatedly	No	Bacterial FAS	Bacteria
Type III	Single protein with multiple modules	Iterative, in which the active sites are reused repeatedly	No	na	Plants and Bacteria

Table 1 Comparisons of PKS types and groups

It can be seen on the fourth column of the table that only the Type I Modular PKS can be predicted; hence, it is included in the scope and limitations of this study. There are several researches which suggest the methods in exploring and forecasting for the other types, such as the PKMiner [34] and PKSIIIExplorer [35], through the use of genome analysis and support vector machines (SVM). [36] However, similar to existing Type I PKS prediction systems – SEARCHPKS [25] and ASMPKS [15] – these systems are able to predict not the polyketide natural product structure but the protein sequence linkages.

Genome is everything encoded in the deoxyribonucleic acid or DNA (i.e. everything about the organisms’ genetic information), and its segment transcribed into ribonucleic acid or RNA is called gene. These genes are responsible for determining amino acid sequence in proteins. Yet, there are still some components of the DNA that cannot be expressed as proteins called non-coding sequences. A genome is made up of genes and non-coding sequences. [37, 38]

Techniques such as DNA sequencing, genome-wide association studies, and improved informatics tools to organize and analyze these data constitute genome analysis. All of these are methods in determining and comparing an organism’s genetic sequence. [37, 38] On the other hand, genome mining deals with the extraction and the prediction of DNA sequence that is beneficial for novel natural product discovery consequently to drug discovery. [22, 39-40] This method utilizes the rapidly increasing DNA sequence data from an extensive diversity of organisms in publicly accessible database. [39, 40]

Automated genome mining and analysis has gained much attention in the exploration of polyketide synthases. [22, 41-42] It is not surprising for the researchers to exploit modern computational techniques and information technology to improve their works. Some of the existing applications that are web-based, if not employing genome analysis or mining, are using protein sequence prediction to determine PKS. Listed below are some of the retrieved PKS database and systems over the internet with a brief background and if available, its design and architecture.

1. PKSDB

The Polyketide Synthases Database (PKSDB) is a searchable database of modular Type I PKS that provides an interface for correlating a polyketide structure to its respective domains and modules. The database currently hosts 20 experimentally characterized modular PKS clusters such as *erythromycin*, *rapamycin*, and *epothilone*. [3, 25]

It can be seen on Figure 4 - 5 that the system has been designed so that the data mining and the search facilities are independent with one another. The first section allows the extraction of various domain sequences, linker sequences, and chemical structure of the natural product of the available PKS domains. The search function, on the other hand, requires an input of a test protein sequence to be analyzed for possible polyketide domain. [3, 25]

The system is developed by Yadav, Gokhale, and Mohanty of the National Institute of Immunology of India and is accessible through the link http://linux1.nii.res.in/~pksdb/DBASE/page.html.

2. SEARCHPKS

SEARCHPKS, created by the same authors, is a web-based program that utilizes the PKSDB. It is a program for identification and analysis of PKS domains from a polypeptide sequence input that uses knowledge-based approach. The system also provides, from the same input, the acyl-transferase residue, the substrate specificity, the sequences of domains and linker regions, and its relationship to and alignment with other PKS domains. [25] The input and output sequence are expressed in FASTA format. SEARCHPKS can be accessed at http://linux1.nii.res.in/~pksdb/DBASE/pagesearchpks.html .

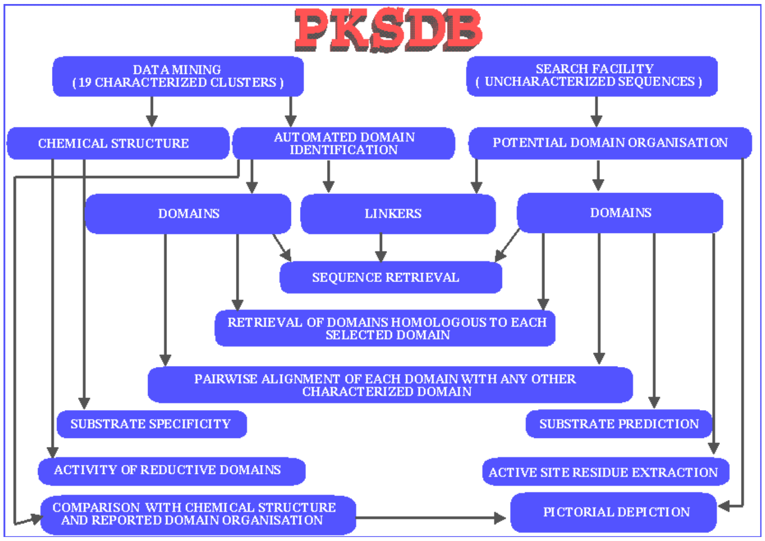
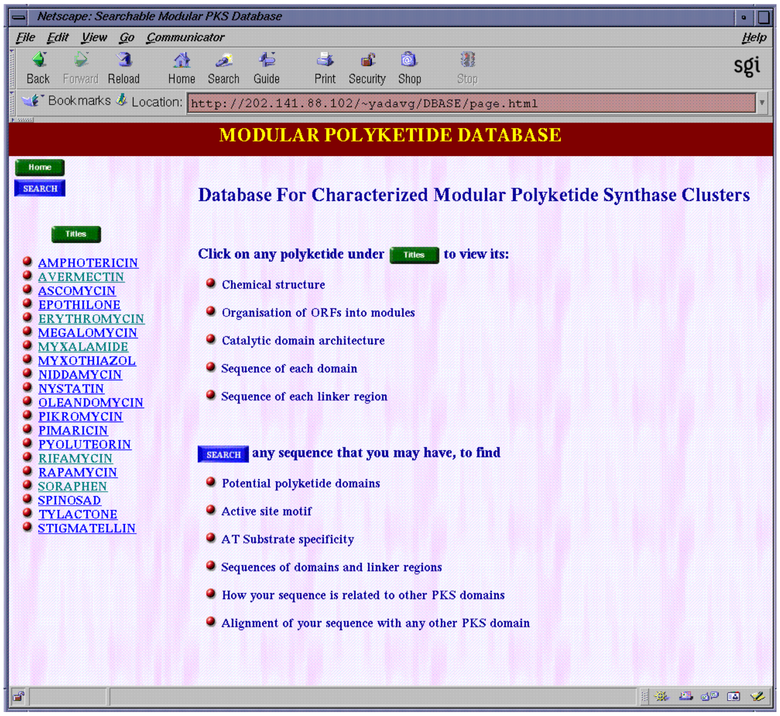


Figure 4 PKSDB Organization Chart (Source: http://linux1.nii.res.in/~pksdb/nar.gif)



MODULAR POLYKETIDE DATABASE

Database For Characterized Modular Polyketide Synthase Clusters

- AMPHOTERICIN
- AVERMECTIN
- ASCOMYCIN
- EPOTHILONE
- ERYTHROMYCIN
- MEGALOMYCIN
- MYXALAMIDE
- MYXOTHIAZOL
- NIDDAMYCIN
- NYSTATIN
- OLEANDOMYCIN
- PIKROMYCIN
- PIMARICIN
- PYOLUTEORIN
- RIFAMYCIN
- RAPAMYCIN
- SORAPHEN
- SPINOSAD
- TYLACTONE
- STIGMATELLIN

Click on any polyketide under Titles to view its:

- Chemical structure
- Organisation of ORFs into modules
- Catalytic domain architecture
- Sequence of each domain
- Sequence of each linker region

SEARCH any sequence that you may have, to find

- Potential polyketide domains
- Active site motif
- AT Substrate specificity
- Sequences of domains and linker regions
- How your sequence is related to other PKS domains
- Alignment of your sequence with any other PKS domain

Figure 5 Screenshot of the PKSDB Homepage

3. ASMPKS

The Analysis System for Polyketide Synthase (ASMPKS) is another web-based application (http://gate.smallsoft.co.kr:8008/~hstae/asmpks/pks_prediction.pl) using the PKSDB data mentioned earlier. Through the use of genome sequences, it searches the database for domain and module data. It also provides known polyketide structures and predicts the PKS of unknown novel polyketide candidates. Every successful analysis is stored for future references. The PKS Navigation component presents the chemical structure of the polyketide and the PKS composition by displaying the arrangement of the PKS' with its respective domain and modules. [15]

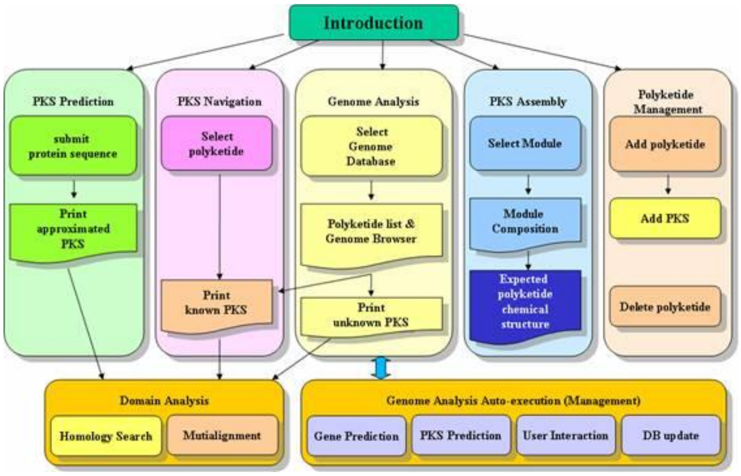


Figure 6 ASMPKS Web Interface Structure

Another component, the PKS Assembly, assembles sets of module and constructs the expected polyketide. [15] This component is the desired result of this study. However, it does not provide the final process of cyclization and does not present the assembly in a linear manner. Also, it does not display the chemical structure in the classic sketch schema where the molecule's orientation is being depicted. Figure 7 shows a screenshot of a sample PKS assembly. From the image, it can be said that it only displays the latest structure and not the whole modular linear synthesis similar to Figure 3.

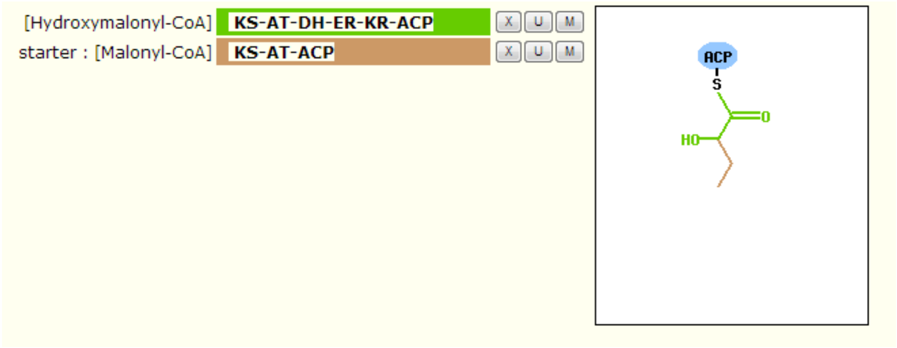

[Hydroxymalonyl-CoA] KS-AT-DH-ER-KR-ACP
starter : [Malonyl-CoA] KS-AT-ACP

Figure 7 PKS Assembly Screenshot

4. PKMiner

Despite the limitation of predicting polyketide biosynthesis to Type I modular PKS, Yi & Kim (2012) provide a way to identify novel Type II PKS gene clusters through the creation of PKMiner – a web-based system that also provides a comprehensive collection of such type. The system also provides analysis of protein or genome sequences. It currently contains 42 known bacterial aromatic polyketide, 280 reported Type II PKSs, 40 Type II PKS gene clusters, 231 predicted Type II PKSs, and 319 actinobacterial genomes. [34]

The system makes use of the *combination of Hidden Markov Model, and sequence pairwise alignment based support vector machine* in the construction of Type II PKS classifiers. [34] The prediction rules for the aromatic polyketide chemotype are derived from the following criteria:

a. *for type II PKS gene cluster mapped onto aromatic with unique domain combination, assign corresponding polyketide chemotype into type II PKS gene cluster;* [43] and

b. *for type II PKS gene cluster mapped onto aromatic polyketide chemotype with overlapped domain combination, assign the most abundant polyketide chemotype of homologs of type II PKS ARO and CYC onto type II PKS gene cluster.* [43]

The system can be accessed with this URL: http://pks.kaist.ac.kr/pkminer/.

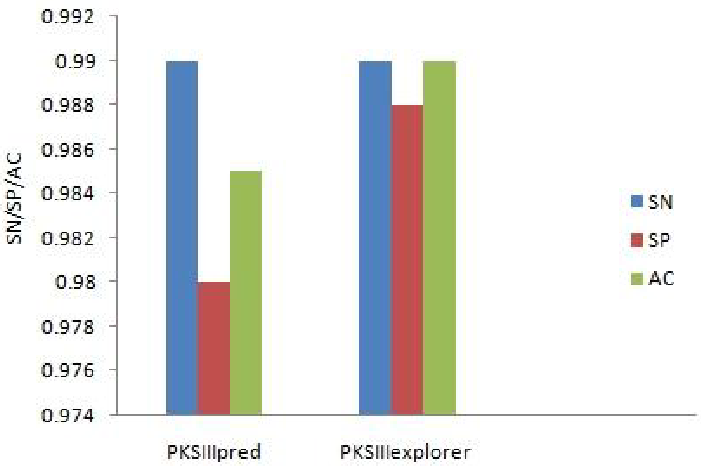
5. PKSIIIExplorer

The PKSIIIExplorer is another web-based system that provides prediction of whether a certain sequence belongs to a synthase; however, it focuses on Type III polyketide synthases. Similar to PKMiner, it is also based on transductive Support Vector Machines (SVM). The user interface allows user to either upload a text file or type-in the protein sequence in FASTA format. PKSIIIExplorer provides both plant proteins and type III PKS from bacteria, fungi, and bryophytes in the training set of the sequence learning algorithm. [35]

PKSIIIExplorer is compared to another Type III PKS prediction server, PKSIIIPred. Figure 8 illustrates the statistical differences of the two systems that support the efficiency of SVM being used in the system. Accuracy (AC), specificity (SP), and sensitivity (SN) are the criteria for such conclusions. The two systems can be accessed at the following URL's :

(1) PKSIIIExplorer - http://type3pks.in/tsvm/pks3/index.php; and

(2) PKSIIIPred - http://type3pks.in/prediction/index.php



.

Figure 8 Comparisons of PKSIIIPred and PKSIIIExplorer

B. Natural Product Likeness

There are only few notable studies done with respect to the calculation of natural product-likeness (NP-likeness) with the earliest being in 2008. [17, 44] NP-likeness, though no formal definition was provided, could be pertained to as the similarity of a molecule to pre-existing natural products. The first NP-likeness computational tool was implemented in a closed source application [17] , but was re-made by Jayaseelan in 2012 to allow access to the source codes. [44]

The application is implemented as CDK-workflows to allow users with no programming background to use the system. It is also available as a stand-alone Java Archive package, which can be used as a library for other stand-alone and web applications. The source codes and the application can be accessed at http://sourceforge.net/projects/np-likeness/ . [44]

III. Theoretical Framework

A. Biosynthesis of Polyketides

The biosynthesis of polyketides is almost similar to the fatty acid synthesis (see Figure 9). [27] Both processes involve the following:

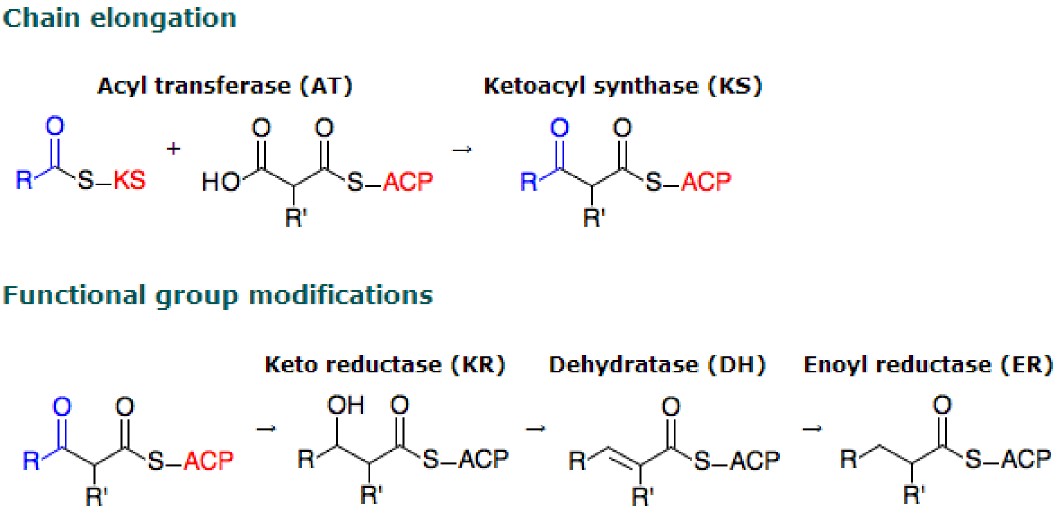


Figure 9 Reaction of Enzymes (Source: http://www.genome.jp/kegg/compound/bs_pk.html)

- Ketosynthase (KS) is where the starter unit is being attached;
- Acetyl-carrier Protein (ACP) holds the unit for every cycle or module;
- Acetyl Transferase (AT) transfers the working molecule to the next module or cycle which will be attached to the KS;

- Ketoreductase (KR) reduces a ketone to an alcohol;

- Dehydratase (DH) removes water from the component;

- Enoyl Reductase (ER) hydrogenates the chain to from a fully saturated methylene center; and

- Thioesterase (TE) catalyzes the off-loading and cyclization of the fully-formed polyketide. [4, 31]

Figure 10 illustrates the fatty acid synthesis (FAS) and the activity and reactions of the mentioned enzymes. The only difference between PKS and FAS is that the former does not require the DH, the ER, and the KR to be present in one cycle.

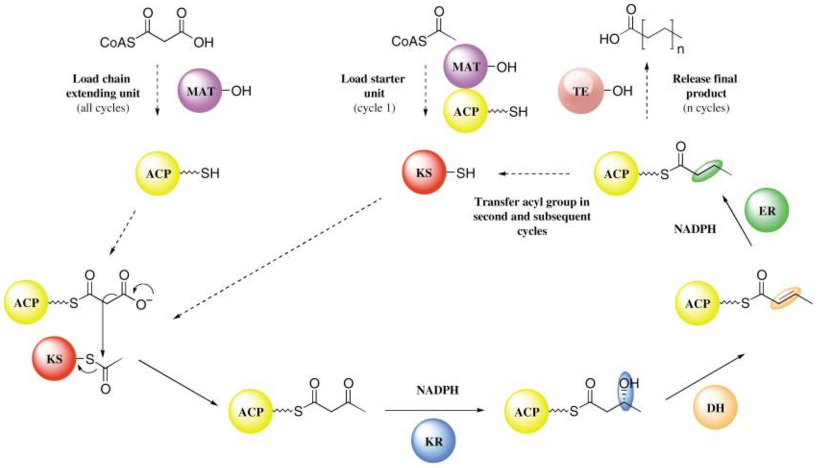


Figure 10 Fatty Acid Biosynthesis Cycle

The basic unit of a PKS is called the module. These modules are composed of domains mentioned earlier – ACP, KS, etc. It is mentioned earlier that not all these domains are required in the polyketide biosynthesis, but must always contain the KS, AT, and ACP since these three enzymes are responsible to catalyze a chain extension cycle. The visual representation of a PKS is illustrated in such a way that a module starts with a KS and ends with an ACP. [4, 31]

The growing chain undergoes three different stages which correspond to three types of polyketide modules. First, the loading stage allows the starter unit, usually an acetyl-CoA or malonyl-CoA, to be loaded and catalyzed on the two domains present, ACP and AT. Next, the elongation stage includes the following domain sequence –KS-AT-{DH-ER-KR}-ACP- , where the enzymes inside the braces are optional *processes*. This stage permits the biosynthesis to add extender units in the polyketide chain. The last stage is the termination stage containing the TE. [4, 31]

Groups of these modules form the enzyme complexes called the polyketide synthase. [33] In the case of our example, Figure 11, there are three PKSs – DEBS1, DEBS2, and DEBS3. Each erythromycin PKS is composed of two modules, but it must be noted that it is not the same case for the others. These PKSs are then *merged* together by linker enzymes. [4] This process and the prediction of whether a PKS will be combined with another PKS is the focus of the reviewed literature.

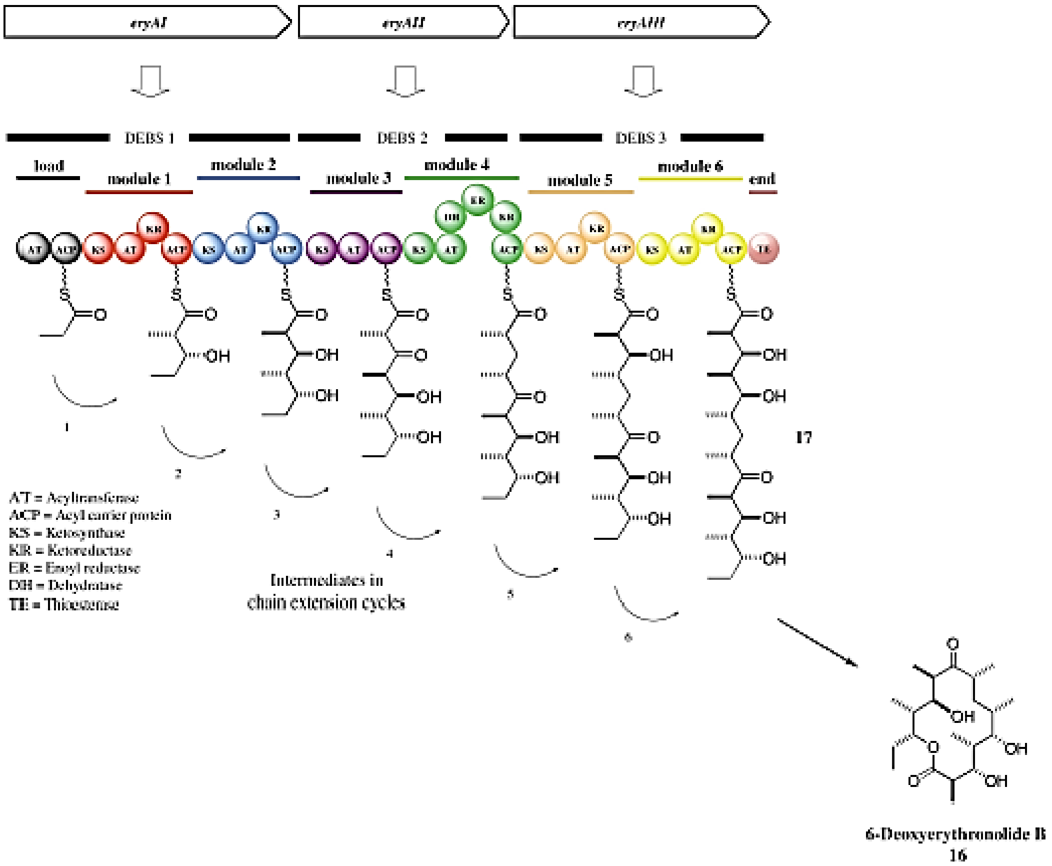


Figure 11 Parts of the erythromycin PKS

B. Natural Product Likeness and Predictions

The natural product (NP) likeness of a molecule (i.e. the similarity of the molecule to existing natural products) is a useful gauge in designing novel bioactive and pharmaco-active compounds. The estimation of the NP-likeness, however, requires prior knowledge of a molecule such as the physiochemical and structural properties. [44]

The data on the experiment was gathered on several online natural products database and synthetic drug database, where Ertl et al. [17] used the CRC Dictionary of Natural

Products for the former. Before the scoring process took place, several molecule curation was done. First, all metallic atoms were removed. More specifically, the only atoms preserved in the molecule are C, H, N, O, P, S, F, Cl, Br, I, As, Se, and B. The next process involves the removal of sugar components leaving only the core molecular structure. [17, 44] After the curation process, the molecule underwent fragmentation. The fragmentation process was the generation of atom signature by Faulon et al. (2003).

A signature is an acyclic subgraph of a molecular graph. An atomic signature is a tree of height *h* rooted from a particular atom, and a molecular signature is the set of all the atomic signatures. [45] Figure 12 shows an example of this kind of molecule fragmentation.

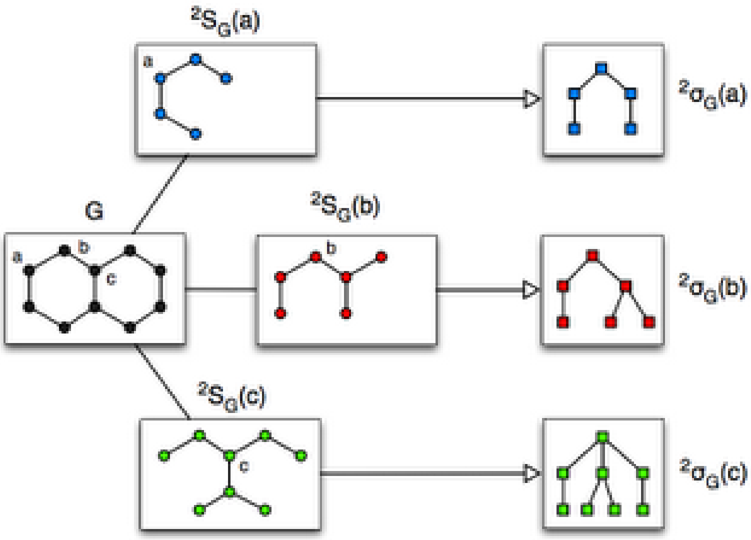


Figure 12 Atomic Signatures

An atomic signature of height 2 is considered as a fragment and quantized by the equation below, where NP_i is total number of molecules in the natural product set where $fragment_i$ occurs; SP_i, total number of molecules in the synthetic chemical set where

$fragment_i$ occurs; and NP_t and SP_t are the total number of molecules from the respective sets. [44] The equation is, in principle, an application of the Naïve Bayes classifier since contribution of each fragment is independent with one another. [44]

$$fragment_i = \log\left(\frac{NP_i}{SM_i} * \frac{SM_t}{NP_t}\right)$$

The NP-likeness score will then be provided by the sum of all the fragments of the molecule, and will be further normalized by the number of atoms in the molecule. The equation below shows this normalization process. [44]

$$NP.likeness = \frac{\sum_{i=0}^{N} fragment_i}{N}$$

C. Tanimoto/Jaccard Coefficient

One of the most useful concepts in graph theory is the substructure similarity search of a query graph. The most popular application is on the field of chemistry – determining whether a molecule contains a desired group (e.g. functional groups). However, this problem is classified as a Non-Polynomial-Complete (NP-Complete) problem which means that its worst-case performance will be an $O(k^{kN})$. [46, 47]

This computer science is actually not useful considering the existence of huge molecules. In fact, a 30-atom structure may take around 17 minutes for a simple substructure search and this duration is inefficient since a professional chemist can determine subgroups in lesser time. [46, 47]

However, recent studies are more interested in the similarity of two structures since it also tells that two substructures are part of one or the other. And unlike, graph search, similarity measures are represented as bits (currently limited to 32 to 2^{30}) making comparisons easier by just determining the intersection of the two sets. Among these mterics, the Tanimoto coefficient is considered the most appropriate, and in fact is the most used. [47]

The Tanimoto coefficient formula shown in Figure 13 is computed as the quotient between the number of bits present in the two molecules (Molecule a and Molecule b), and the sum of number of bits present in both molecules less their intersection. [47]

$$T(a, b) = \frac{N_c}{N_a + N_b - N_c}$$

Figure 13 Tanimoto coefficient

D. Data Sources

Jayseelan et al. made use of the ChemBL database as the source of data for the natural product molecules set, and the ZINC database for the synthetic drug-like reference set.

1. ChemBL Database

The ChemBL database is an open-access database developed by Gaulton et al. containing the structure and information of drug-like and bioactive compounds. The database can be accessed through a web interface at https://www.ebi.ac.uk/chembldb. [48]

2. ZINC Database

The ZINC database is another open-access database but contains purchasable (synthetic) compounds. Similar to ChemBL, it allows free download of the compounds in popular chemical formats [49]. In the case of the mentioned study, the SMILES format is gathered. The system can be accessed at http://zinc.docking.org.

E. Chemical Structure Models and Format

The Chemical Development Kit (CDK) library is a chemistry and cheminformatics Java collection which was originally authored by Christoph Steinbeck, Egon Willighagen and Dan Gezelter. The said Java library is capable of producing 2D graphics of chemical structures, reactions, and spectra, and performing quantitative structure activity relationship (QSAR) descriptor calculations. The library is an open-source implementation, and is being used extensively in programs like JChemPaint, NMRShiftDB, and others. [26]

One of the reasons of using this library for this study is its open-source licensure. Unlike web-based chemical libraries (ChemDoodle, being the most prominent in cheminformatics [50]), a locally available plugin does not require access to cloud service subscriptions. Implementation of CDK in the prediction component mitigates the possibility of having a chemistry cloud service subscription withdrawal. Other factors that caused the usage of this library are small memory size, mobility and compatibility, thorough testing and wide development and support team, and security. [26]

Another reason of the utilization of the CDK library is its capability to create 2D chemical structures using native Java canvas, calculate basic information about the designed structure such as the molecular formula, molecular mass, etc.; and interpret and store structures using the SMILES format. [26] The SMILES format is a de facto format of transforming molecules into traditional character strings. [18] The library also allows conversion of a molecule to MolFile format, another molecule representation format standard. [19]

The Simplified Molecular Input-Line Entry Specification or more commonly known by its acronym **SMILES**, is a format to describe chemical structures using simple ASCII characters. It includes the connectivity of two atoms. A single bond is represented by two adjacent atom symbol, a double bond with a '=' in between, and a triple bond with a '#'. Though this format allows 2D and 3D representation, it does not provide coordinates for these. Another thing, all atoms are represented in this format except for the H-atom. [18] A tutorial on how to use this format is presented in http://www.daylight.com/dayhtml_tutorials/languages/smiles/index.html.

On the other hand, **MolFile** is another chemical structure representation format by MDL. This format maps a compound with two matrices. The first matrix has a number of rows equal to the number of atoms. The first four columns of this matrix are the x, y, and z coordinates, and the atom concerned. The second matrix is the bond matrix with number of rows equal to the number of bonds. The first two columns are the two bonded atoms, and the third column indicates the type of bond – 1 for single bond, 2 for double, 3 for triple, and 4 for aromatic. [19]

Figure 14 shows the molecular structure of the compound C_4H_{10}. The SMILES representation of the given example is CCCC signifying the four carbon atoms in the system – the first C for the H_3C in the figure, the second and third are the implicit C-atoms (the "corners"), and the last C as the CH_3. The molFile representation of the same compound is given by Figure 15.

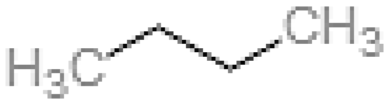
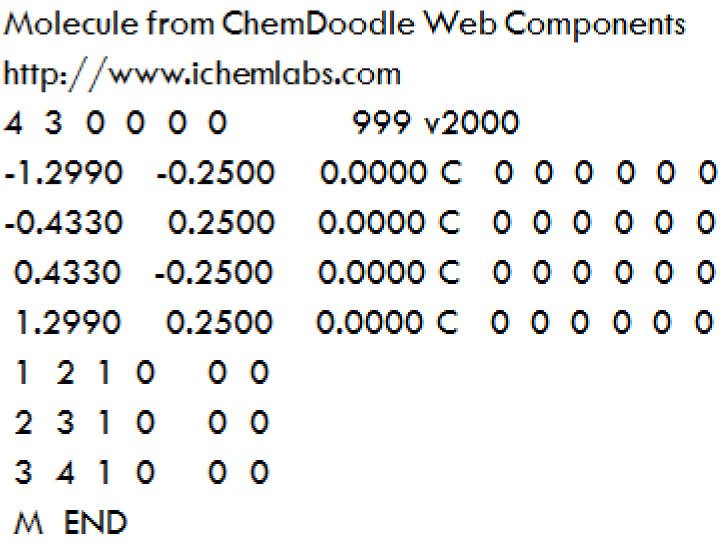


Figure 14 C14H10 molecule structure



```
Molecule from ChemDoodle Web Components
http://www.ichemlabs.com
4  3  0  0  0  0          999 v2000
-1.2990   -0.2500   0.0000 C   0  0  0  0  0  0
-0.4330    0.2500   0.0000 C   0  0  0  0  0  0
 0.4330   -0.2500   0.0000 C   0  0  0  0  0  0
 1.2990    0.2500   0.0000 C   0  0  0  0  0  0
 1  2  1  0    0  0
 2  3  1  0    0  0
 3  4  1  0    0  0
 M  END
```

Figure 15 C4H10 MolFile representation

F. MVC Design Pattern and CodeIgniter

The most prevalent server-side scripting language is PHP. As compared to its competitors, the language allows a cheaper development cost due to the abundance of web servers where it can be deployed; unlike ASP, which requires Windows with IIS components. Other than the freedom of choosing its server host (Windows or the most common, Linux are just some), PHP also allows developers to implement their web applications using procedural programming, object-oriented programming (OOP), and a mixture of both. [51, 52] The current version as of October 2012 is version 5.4.7. (php.net)

Since PHP version 3, basic OOP functionalities are added but are fully implemented since the fourth version. However, it was only in PHP v5 that the language was rewritten to allow full object model. This latest version includes the following features: visibility, abstract and final classes and methods, magic methods, interfaces, cloning, and type hinting. [53, 54]

This development led to the implementation the Model-View-Control model in PHP, thus the creation of several frameworks.

The MVC design pattern, in general, is a paradigm that tries to separate user experience and information representation; breaking an application or even a single component of the software into three entities – the model, the view, and the controller. The primary reason to its development is the mapping of the traditional input/output (I/O) processing to the Graphical User Interface (GUI) as illustrated in Figure 16. The paradigm also promotes code reusability and modularity. [55, 56]

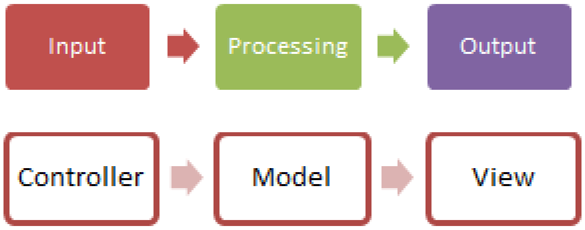


Figure 16 MVC Design Pattern

The model component of this paradigm is its knowledge representation. In most cases, a model is a structure of objects and in some; it can be almost useless as a single object. It should also represent and should have a one-to-one correspondence with the real world objects in which it was modeled from. The view deals with the visual presentation and representation of the models. It gets data from the model and decides how these data will be displayed. It either enhances or suppresses attributes of the said model. Last, the controller serves as the link between the system and the users. It facilitates the changes in the state of the model, and acts as the *listener* of the application to certain events such as key, and mouse. [55, 57]

There are several PHP frameworks that implement the MVC paradigm, Zend [58], CakePHP [59], Yii [60], and CodeIgniter [61] to name some. On a survey done by phpframeworks.com, both Yii and CodeIgniter made the Top 3 on both the Top Hottest PHP Frameworks, and the Top Ranking PHP Frameworks lists; Yii ranking first on both. [62] However, other studies comparing PHP frameworks claim that CodeIgniter (CI) is the best framework due to its support to PHP v4, which is not available on other systems. CI's major disadvantage is the absence of AJAX. But this deficiency can be fixed. [63, 64] Figure 17 shows CI's architecture and its implementation of the MVC paradigm.

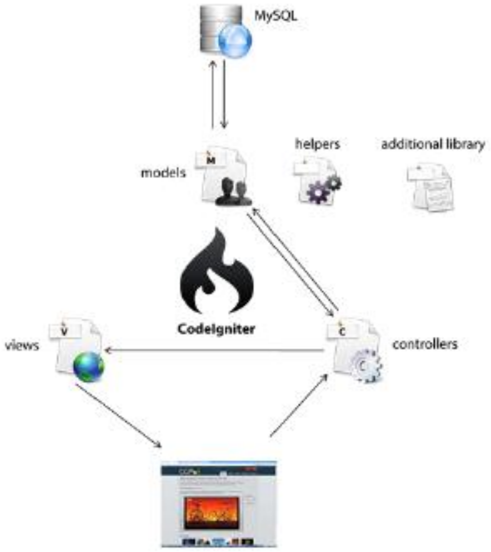


Figure 17 CodeIgniter

IV. Design and Implementation

A. System Design

The project is composed of two computer systems – *Predyketide* and *Polykase* – interrelated only by the prediction zip file. *Predyketide* is a stand-alone desktop Java application for novel polyketide natural product prediction, while *Polykase* is a web information system for naturally occurring Type I modular PKNP and a researchers' community. The context diagram of these two systems is shown in Figure 18.

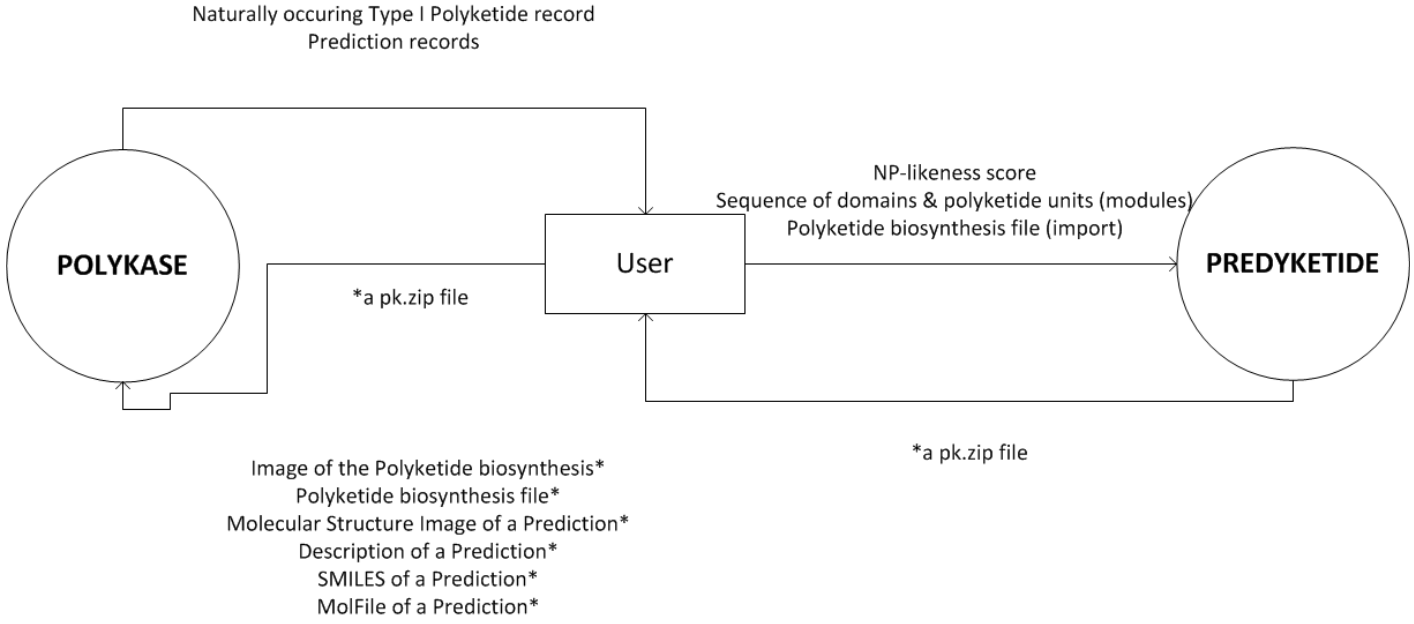


Figure 18 Context Diagram of *Predyketide* and *Polykase*

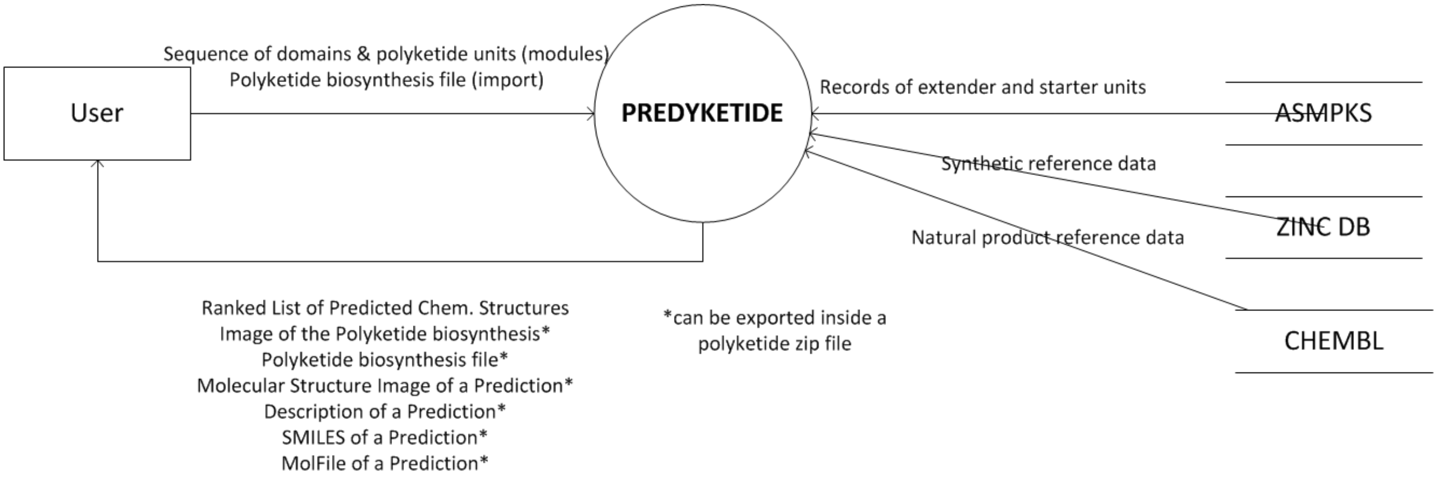


Figure 19 Context Diagram, *Predyketide*

The context diagram in Figure 19 shows the interaction of *Predyketide* to external objects. For every addition of module in the biosynthesis by the user, the application replies with the updated cartoon of the polyketide chain structure. He/She is also capable of importing a saved synthesis file to continue a polyketide elongation. After sending a series of commands, the user is provided with a polyketide zipped (.pk.zip) folder containing the files in asterisk. Due to limitations of the CDK library, the synthesis cartoon is rendered similar to Figure 20 , instead of Figure 3.

Three external systems are used by *Predyketide* – the CHEM BL for the natural product reference data, the ZINC DB for synthetic compound reference data, and the ASMPKS for the records of known polyketides, and starter and extender units. Reference data refers to the records where the predicted natural product is compared for the NP-likeness scoring using Tanimoto coefficient. [44]

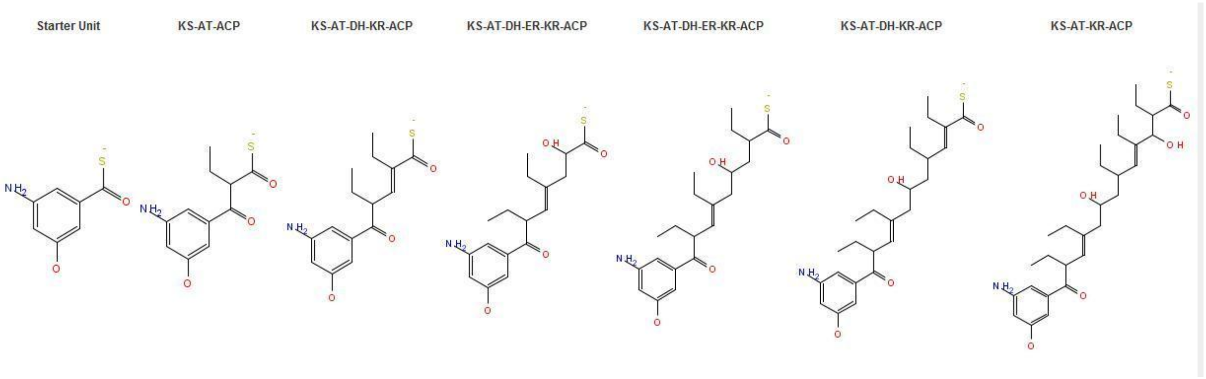


Figure 20 Actual biosynthesis cartoon, *Predyketide*

On the other hand, *Polykase* is the only module of the project accessible to public users. However, this privilege is only limited to viewing of the PKNP data. Only the system administrator is capable of adding, modifying, and deleting records of the system. This sub-system also exploits the ASMPKS databases for the purposes mentioned earlier. This component also allows guest users to create an account and provides the administrator and registered users different dashboard functionalities for better data management.

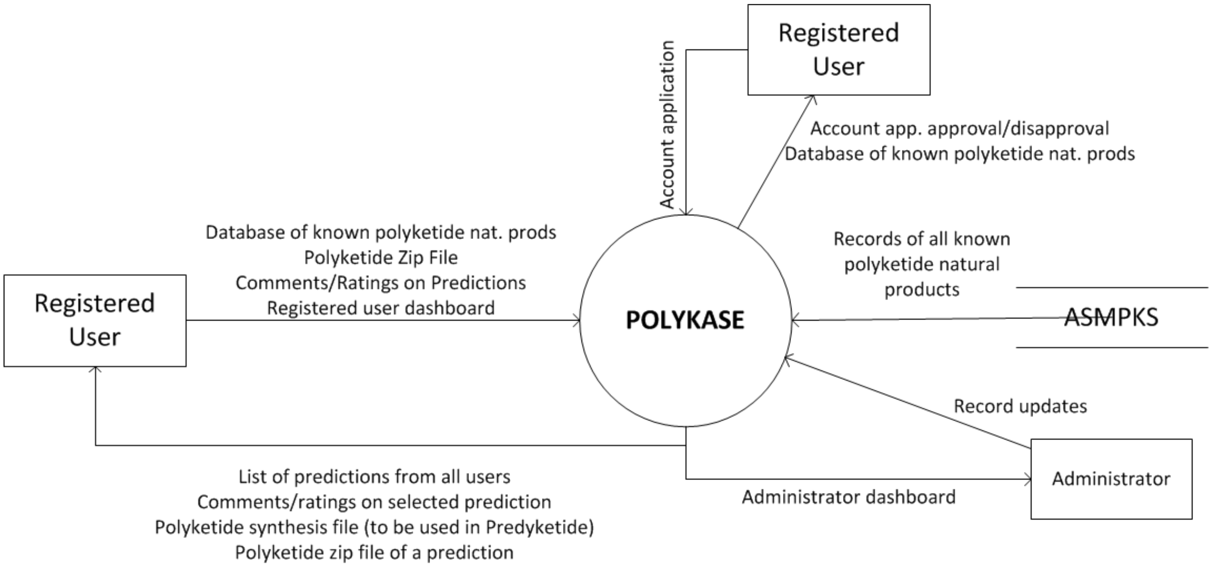


Figure 21 Context Diagram, *Polykase*

1. Predyketide

The most critical factor for the success of this project is the creation of this prediction component. This section focuses on the flow of data, both input and output, to and from *Predyketide* and related external systems. The topmost level Data Flow Diagram (DFD) of this component is illustrated in Figure 19. Figure 22, on the other, shows the level 1 DFD with the commands the users can give to the system.

An application user is permitted to initiate a biosynthesis by choosing the starter unit from a list of 16 which he/she can extend by adding a minimum of three and maximum of 15 extender units. The chain will be elongated for every domain sequence and extender unit input, and will halt once the number of modules mentioned earlier is met or when the user clicks the **Predict** button. All the process mentioned will trigger a change in the biosynthesis cartoon, which is dynamically being updated and displayed to the user.

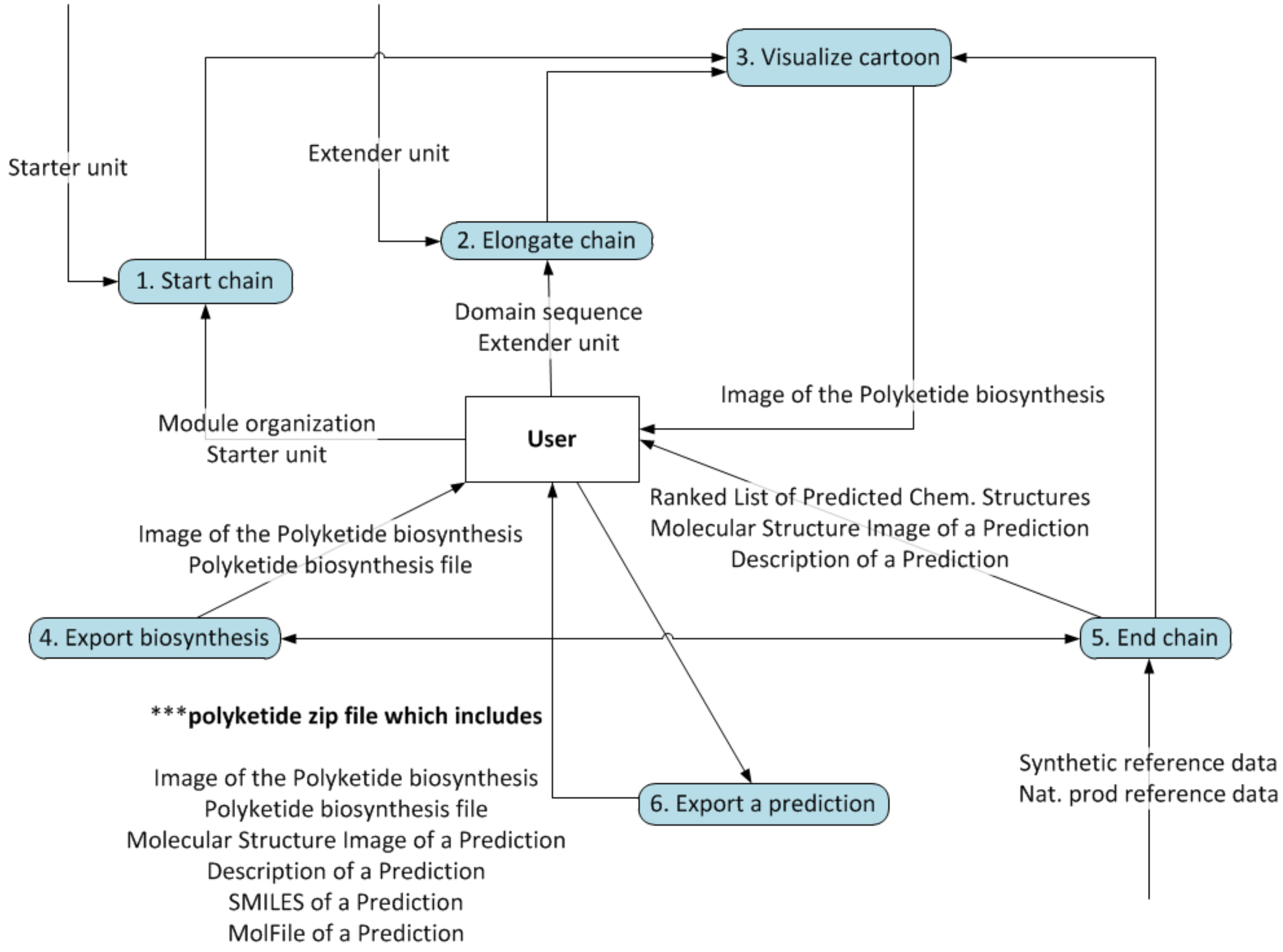


Figure 22 Level 1 DFD, *Predyketide*

Figure 23 and Figure 24 represents the XML tree structure of the starter and extender units, respectively. The two files are designed into two separate files for ease of data update. However, it can be seen that the two have similar structures – both have the (polyketide unit) name, SMILES representation, image file name, and description as children elements.

Instead of introducing a thioesterase (TE) in the chain, a command button is provided to end the chain (i.e. the PREDICT button). The system returns a ranked list of novel natural product predictions based from the NP-likeness scores. After this process, the user is given an option to export a prediction in a zipped folder containing the biosynthesis file (.pksf) , biosynthesis cartoon image (.jpg), the predicted molecule image (.jpg), SMILES string, MolFile representation, and a text file containing some QSAR properties. The polyketide synthesis file can be exported if the user feels continuing a saved work.

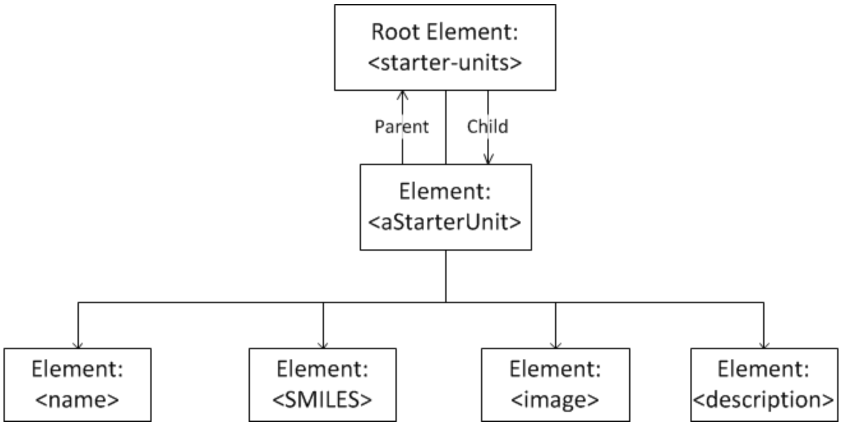


Figure 23 XML Tree of Starter Units, *Predyketide*

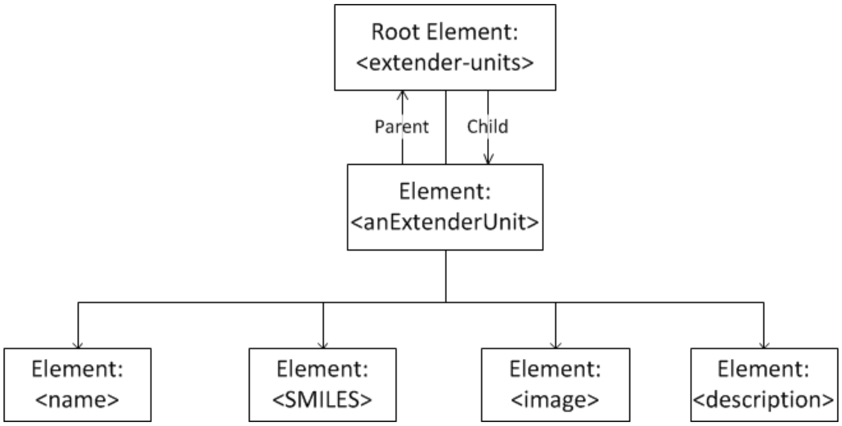


Figure 24 XML Tree of Extender Units, *Predyketide*

The sub-explosion of Process 2 (*Elongate chain*) is described in Figure 25. The domain sequence is parsed and is searched for the strings “DH”, “ER” and “KR”. A boolean is returned to indicate the string’s presence. A “DH”, or dehydratase, removes water from the component; an “ER”, or enoyl reductase forms a methylene center; and a “KR”, transforms a ketone to an alcohol. These reduction operations are applied to the extender unit parameter.

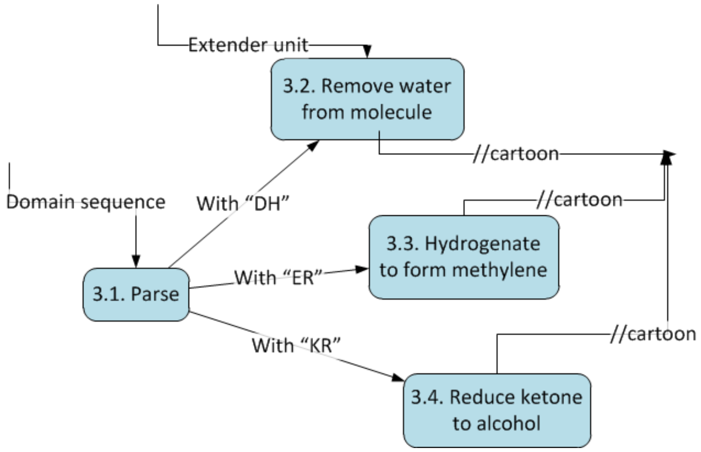


Figure 25 Sub-explosion of Elongate chain process, *Predyketide*

Once the user decides to terminate the elongation chain by clicking the command button mentioned earlier, the system receives a vector of all possible structures. Since the final domain is a TE, which initiates the esterification process, the S (refer to the biosynthesis figures, Figures 2 and 3) on the top of the chain is removed, and the lone atom finds a new atom to pair; in this case, it is the O in the OH. This process produces water by-product. [4, 9]

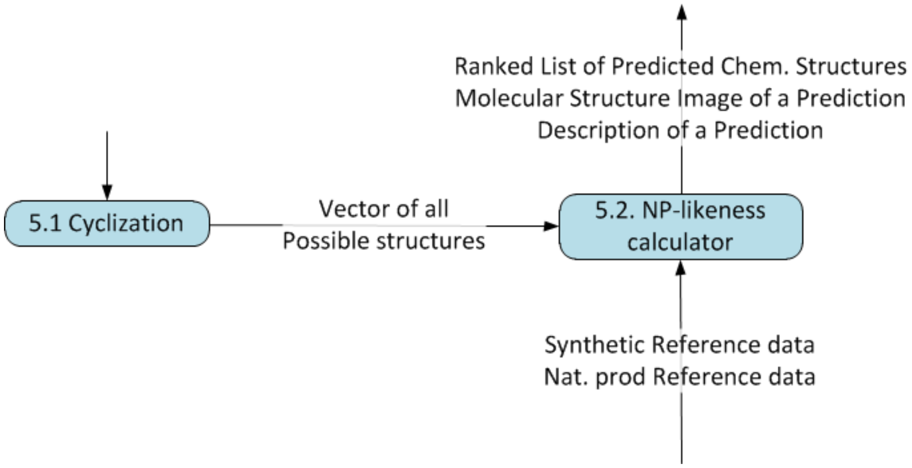


Figure 26 Sub-explosion of End chain process, *Predyketide*

This vector of possible structures is then ranked according to their NP-likeness scores. A similar implementation to the NP-likeness calculator [17, 44] is implemented in the *Predyketide* system except for the curation process. Since no metallic atoms or sugar components are introduced in the chemical substance, this process is disregarded. The computation process starts with fragmentation by extraction of atomic signatures. Instead of using a similar formula for the fragment score, the Tanimoto coefficient [46, 47] is used (Equation 1). The reference data is gathered from the external database systems mentioned earlier. The process results to a ranked list of the provided structure vector with the corresponding NP-likeness scores. The sub-explosion DFD in Figure 27 shows this process.

(Equation 1) $$fragmentScore(a,b) = \frac{N_c}{N_a + N_b - N_c}$$

(Equation 2) $$NP.likeness = \sum_{i=0}^{N} fragmentScore$$

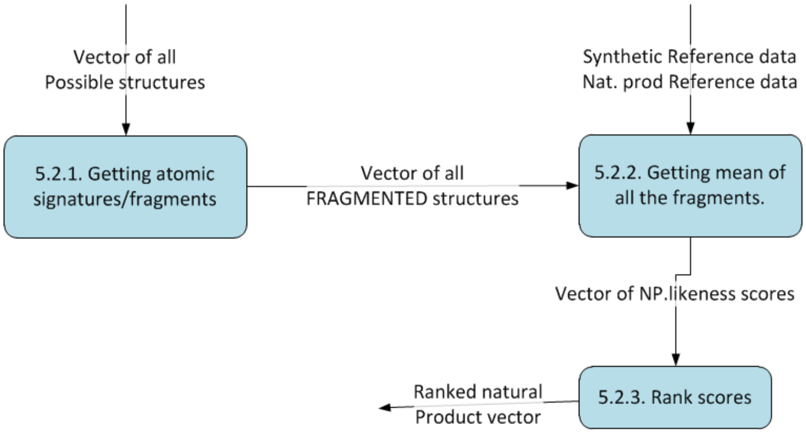


Figure 27 Sub-explosion of Process 5.2. NP-Likeness calculator, *Predyketide*

2. Polykase::Use Case Diagram

Polykase has three levels of data access. First, the unregistered or *guest users* are only capable of viewing the contents of the component and requesting an authorized account access. The registered users and the administrator should log-in first with their login credentials in order to access their respective account privileges.

The registered users are allowed to submit predictions (the exported .pk.zip file from *Predyketide*), and comment and rate others works. However, a maximum of three uploads are given to each user to avoid web server data congestion.

The system administrator is responsible for the approval of account applications, and the moderation of the polyketide prediction submissions and peer reviews.

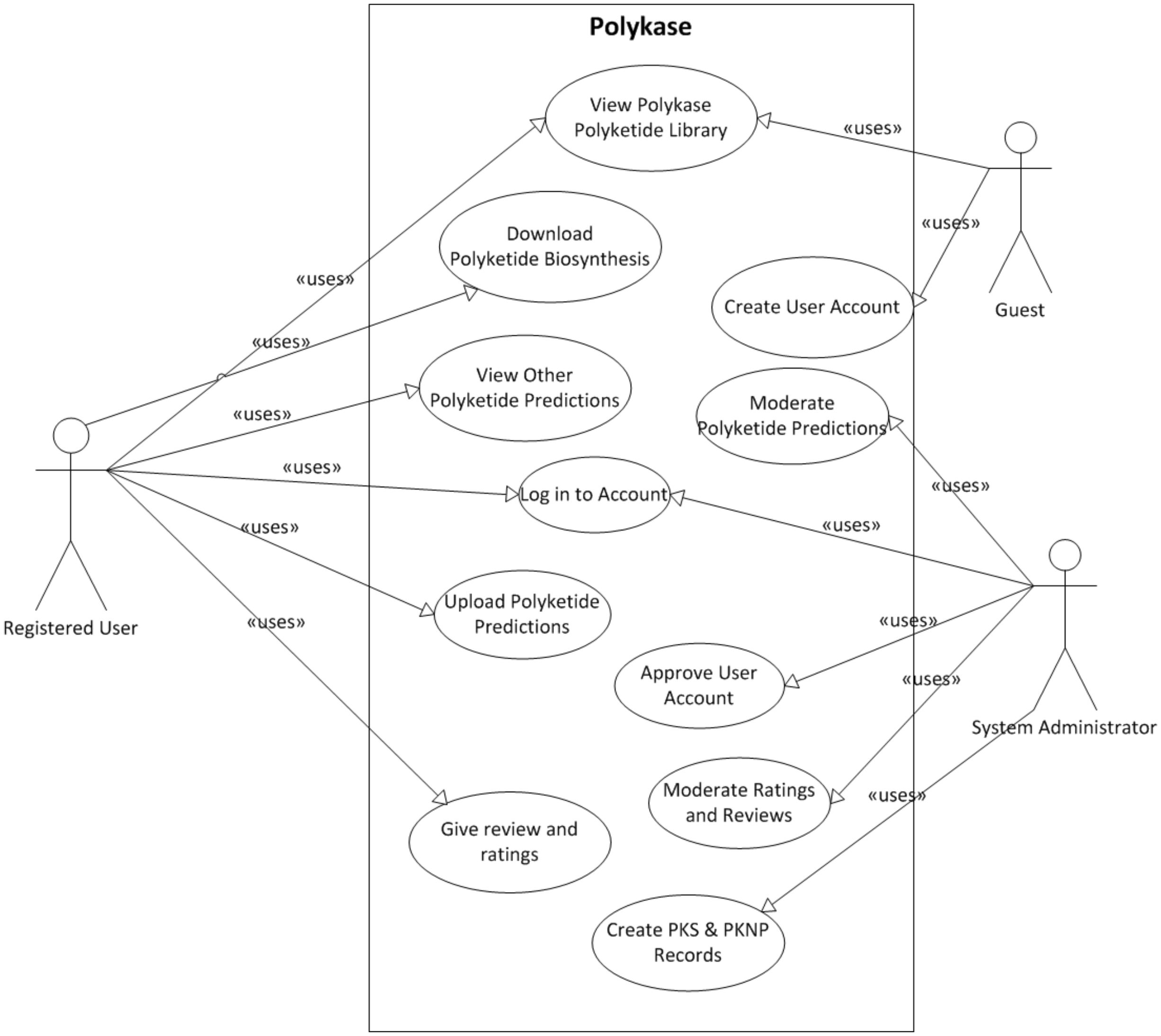


Figure 28 Use-Case Diagram, *Polykase*

3. Activity Diagrams

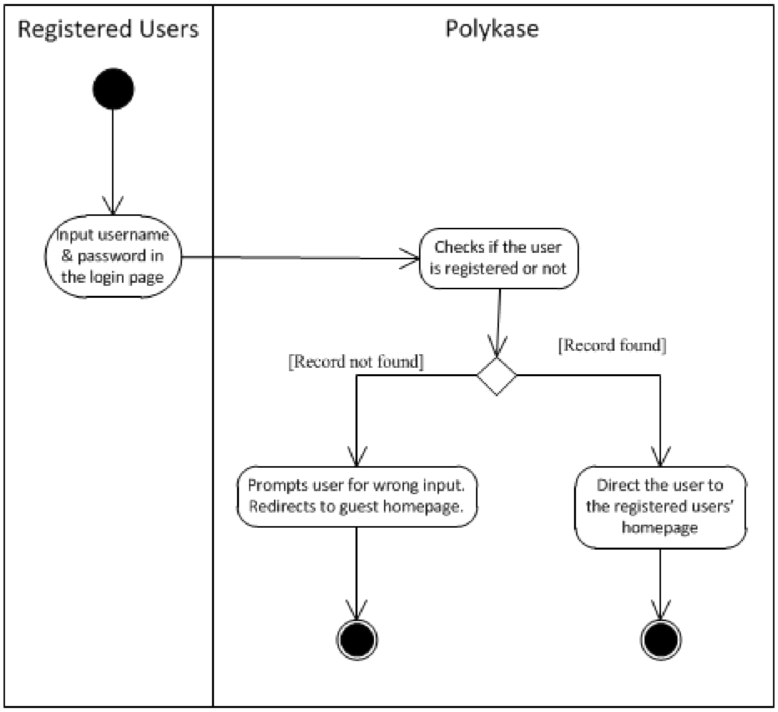


Figure 29 Activity Diagram for the Login Process, *Polykase*

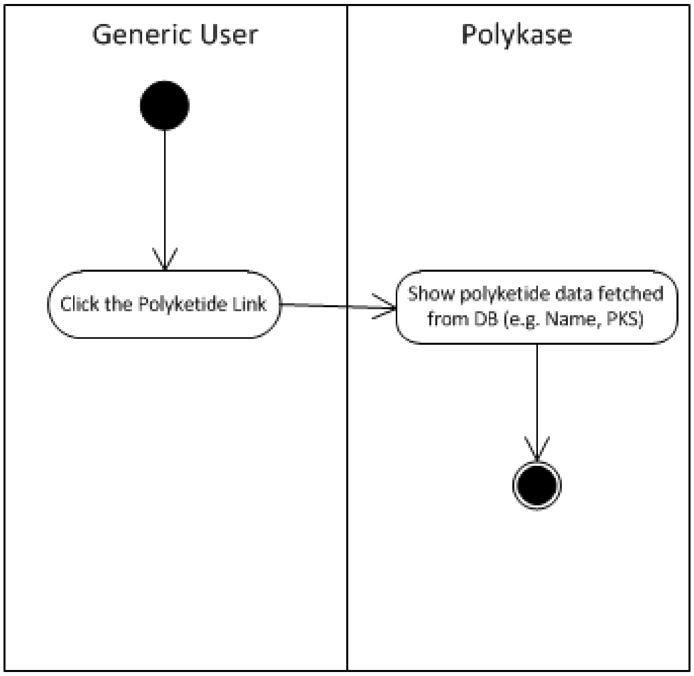


Figure 30 Activity Diagram for viewing the contents PKDB Component process, *Polykase*

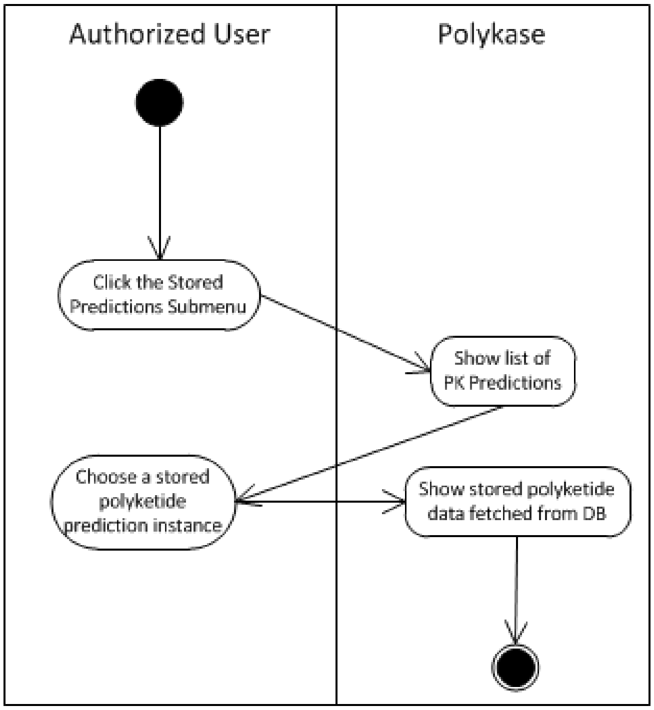


Figure 31 Activity Diagram for the Viewing of Submitted Predictions Process, ***Polykase***

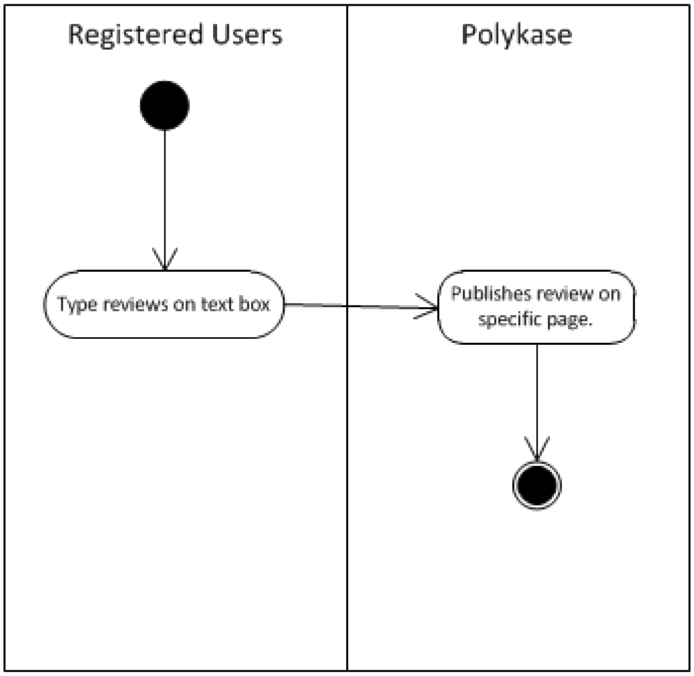


Figure 32 Activity Diagram for the Review Publishing Process, ***Polykase***

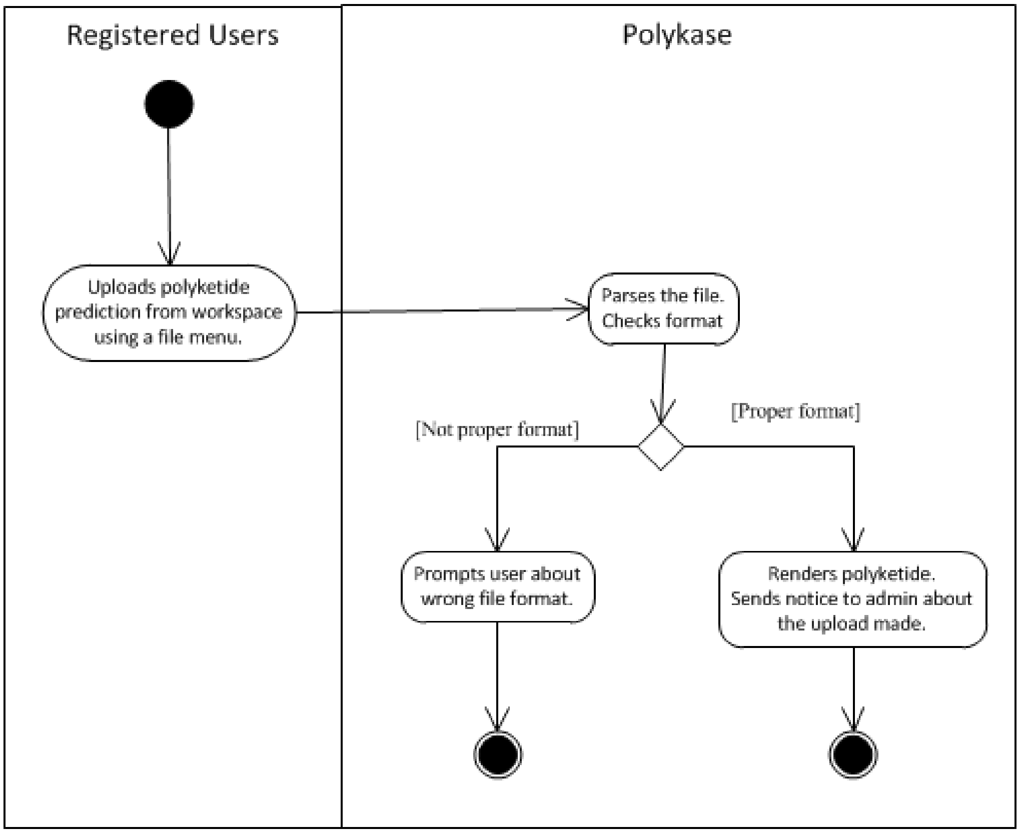


Figure 33 Activity Diagram for the Upload Polyketide Prediction Process, ***Polykase***

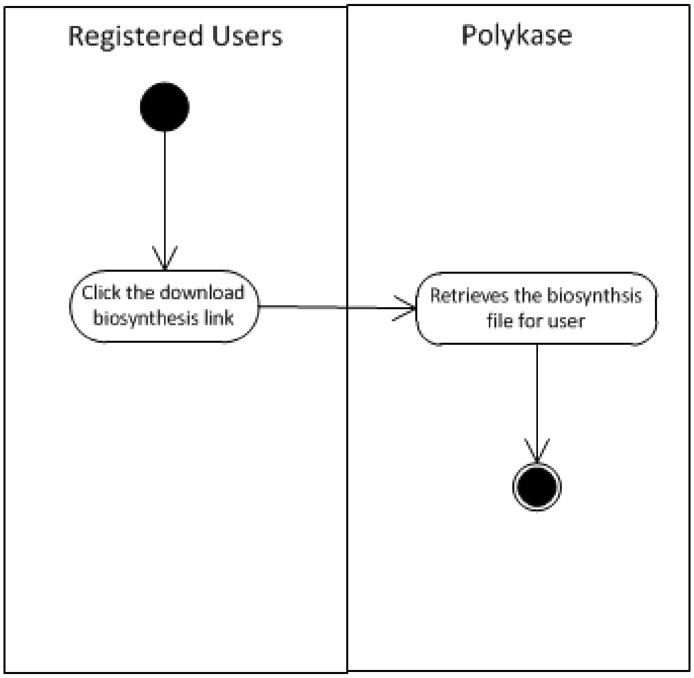


Figure 34 Activity Diagram for the Export Polyketide Biosynthesis Process, ***Polykase***

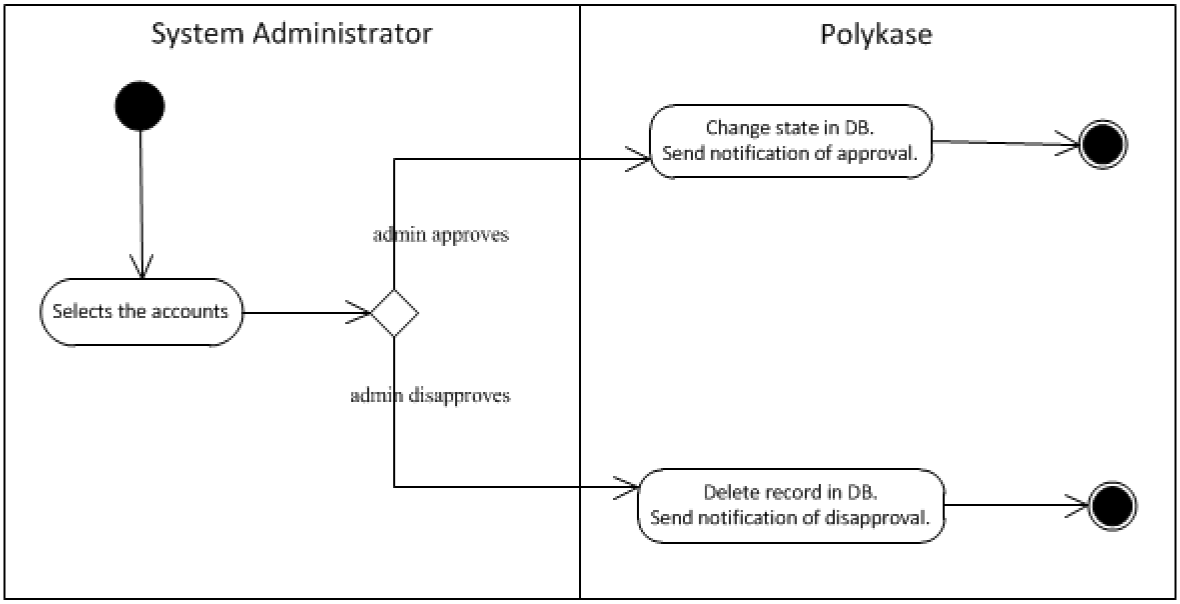


Figure 35 Activity Diagram for the Account Approval Process, *Polykase*

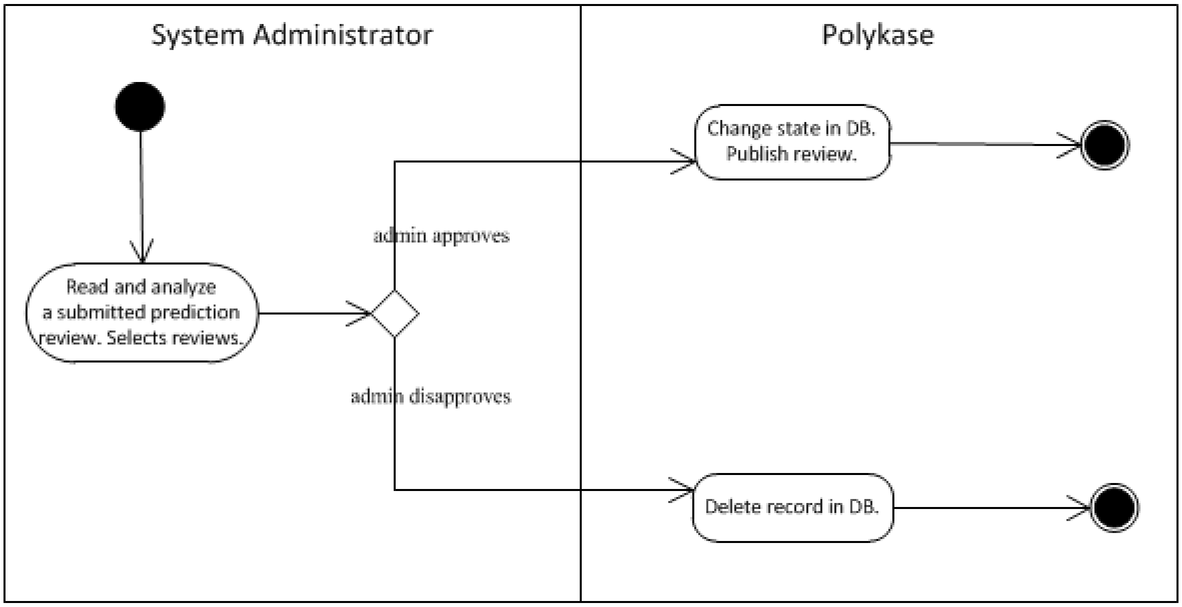


Figure 36 Activity Diagram for the Moderate Reviews Process, *Polykase*

4. Entity Relationship Diagram and Data Dictionary

Polykase's database consists of nine (9) tables. The relationship of each table is presented in the Entity Relationship Diagram (ERD) in Figure 37. Listed below are the said tables with a short description.

a. *users* – a list that includes both the administrator and other registered users. The admin holds a userID 1.

Field	Type	Description
<u>userID</u>	INT	Unique indentifier of a registered user.
<u>uname</u>	VARCHAR(20)	Unique name used for login.
password	VARCHAR(32)	Encrypted user password
surname	VARCHAR(20)	Surname of the user
fname	VARCHAR(50)	First name of the user
occupation	VARCHAR(50)	Occupation of the user
approved	INT	1 if approved; 0, otherwise
email	VARCHAR(3)	The email address of the user
home_address	VARCHAR(100)	Mailing address of the user
contact_num	VARCHAR(25)	Contact number of the member
short_bio	TEXT	Short bio of the member

Table 2 Data Dictionary of AuthUsers table, *Polykase*

b. *rating*– a user can give only one score to a prediction.

Field	Type	Description
<u>pred_id</u>	INT	Unique indentifier of a submitted polyketide prediction
<u>uname</u>	INT	Foreign key. Author of the prediction
<u>user_id</u>	VARCHAR(30)	Foreign key. Author of the prediction
rating_score	INT	The score provided by the user

Table 3 Data Dictionary of Starters table, *Polykase*

c. *prediction* – records of the shared predicted polyketide natural product.

Field	Type	Description
pred_id	INT	Unique indentifier of a submitted polyketide prediction
uname	INT	Foreign key. Author of the prediction
user_id	VARCHAR(30)	Foreign key. Author of the prediction
pred_name	VARCHAR(32)	A one-word name of the prediction
natprod_name	VARCHAR(120)	Weak relationship between the predictions and the known natural products
submission_date	TIMESTAMP	Date when the prediction was shared

Table 4 Data Dictionary of NovelProduct table, *Polykase*

d. *comment*– a table for the reviews to a novel natural product. This is accessed whenever a registered user comments on a colleague's work.

Field	Type	Description
pred_id	INT	Unique indentifier of a submitted polyketide prediction
uname	INT	Foreign key. Author of the prediction
user_id	VARCHAR(30)	Foreign key. Author of the prediction
comment_text	INT(2)	Comment text
comment_date	TIMESTAMP	Date of the comment

Table 5 Data Dictionary of Feedback table, *Polykase*

e. *pks_module* – The children of the module_unit and pk_synthase tables.

Field	Type	Description
module_unit	INT	The module unit used (e.g. starters or extenders)
involved_synthase	INT	The synthase it is included
domain_seq	INT	(e.g. KS-AT-ACP)

module_order	INT	The order on which the module appeared in the synthase

Table 6 Data Dictionary of Extenders table, *Polykase*

f. *module_unit* – table that contains all known and accepted starter and extender units. Data were gathered from ASMPKS.

Field	Type	Description
module_id	INT	The unique identifier of the unit
module_name	VARCHAR(20)	The name of the module unit
module_type	VARCHAR(20)	Extender or Starter

Table 7 Data Dictionary of ExtDomain table, *Polykase*

g. *pknatprod*– records of naturally occurring Type I modular PK NP. This is accessible through the PKDB component.

Field	Type	Description
natprod_name	VARCHAR(30)	Name of the naturally occurring polyketide natural product (e.g. Erythromycin)
host_organism_id	INT	The identifier of the organism where the polyketide was cultured.
pknpImage	VARCHAR(20)	File name of the natural product
NP_Desc	VARCHAR(120)	Short description of the natural product

Table 8 Data Dictionary of PolyketideNaturalProduct table, *Polykase*

h. *Organism* – a table for the host and source organisms.

Field	Type	Description
organism_id	INT	Unique identifier for a certain organism.
gene	VARCHAR(20)	If source, the gene where it is retrieved
organism_specie	VARCHAR(20)	If host, the organism it is cultured
organism_type	VARCHAR(20)	“Host” or “Source”

Table 9 Data Dictionary of Organism table, *Polykase*

i. *pk_synthase* – a registry of polyketide synthases (e.g. DEBS, RAPS).

Field	Type	Description
PKS_id	INT	The unique indetifier of the polyketide synthase
PKS Name	VARCHAR(30)	Unique name of a polyketide synthase (e.g. DEBS1, RAPS1)
source_organism_id	INT	Identifier of the organism where the PKS was obtained
natprod_name	VARCHAR(30)	Name of the PK NP it was used.
resources	TEXT	The resource materials
pks_order	INT	The order on which the pks appeared in the natural product

Table 8 Data Dictionary of PolyketideNaturalProduct table, *Polykase*

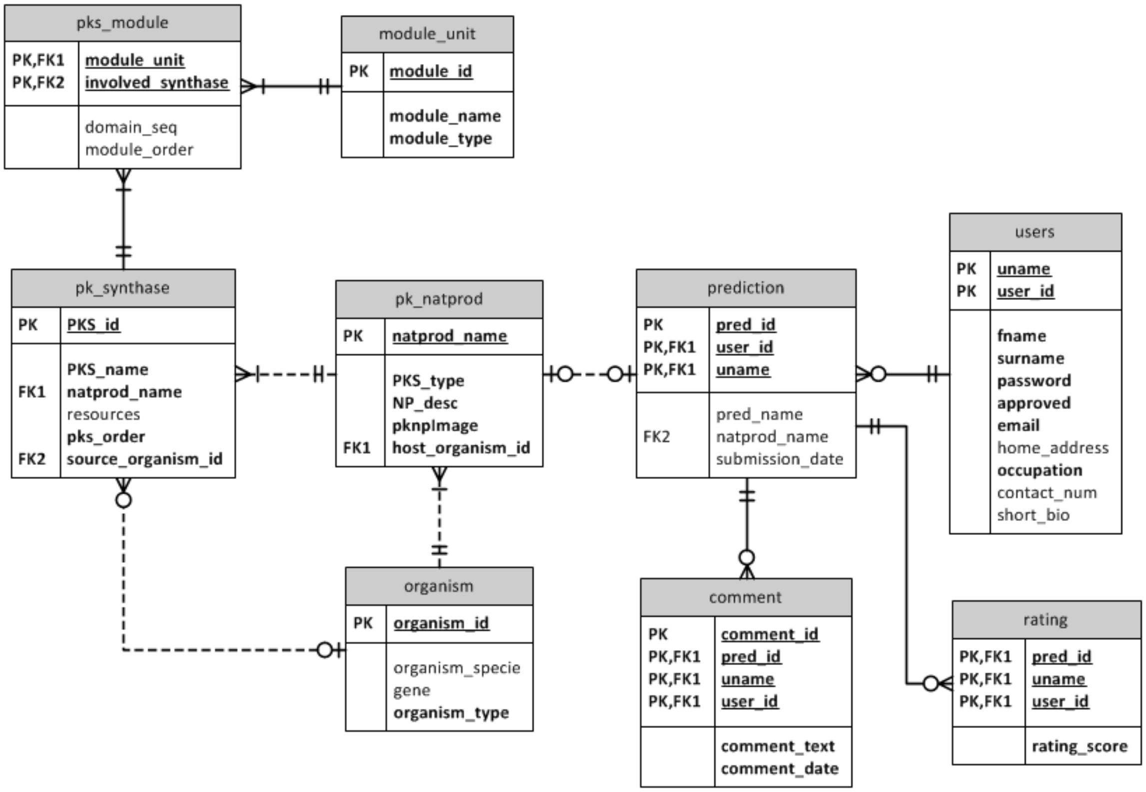


Figure 37 Entity Relationship Diagram, *Polykase*

B. System Requirements

Predyketide is implemented in Java version 1.6 and uses the following external libraries: cdk-1.5.0 and cdk-jchempaint-26. This executable needs no installation nor internet access for usage. Once downloaded, users can store any number of predictions in his/her unit.

Polykase is coded in PHP v.5 using the CodeIgniter v.2.2.1 framework. It uses a MySQL v.5.1 database. These require the system to be deployed on an Apache 2.2 web server with a MySQL5.1 database server.

The end-users of the system need to have a stable internet connection to access the information system. Any browser can be used but Mozilla Firefox and Google Chrome are highly suggested.

V. Results

A. Predyketide

Predyketide is available for viewing at http://sourceforge.net/projects/predyketide/ under the Creative Commons Attribution License and *Chemisty*, *Simulations,* and *Visualizations* categories. It is developed using the Java programming language with Java Swing user interface. The intended users of this application are drug researchers.

For interested developers, the source code can be checked-out using the command:

```
svn checkout svn://svn.code.sf.net/p/predyketide/code/trunk predyketide-code
```

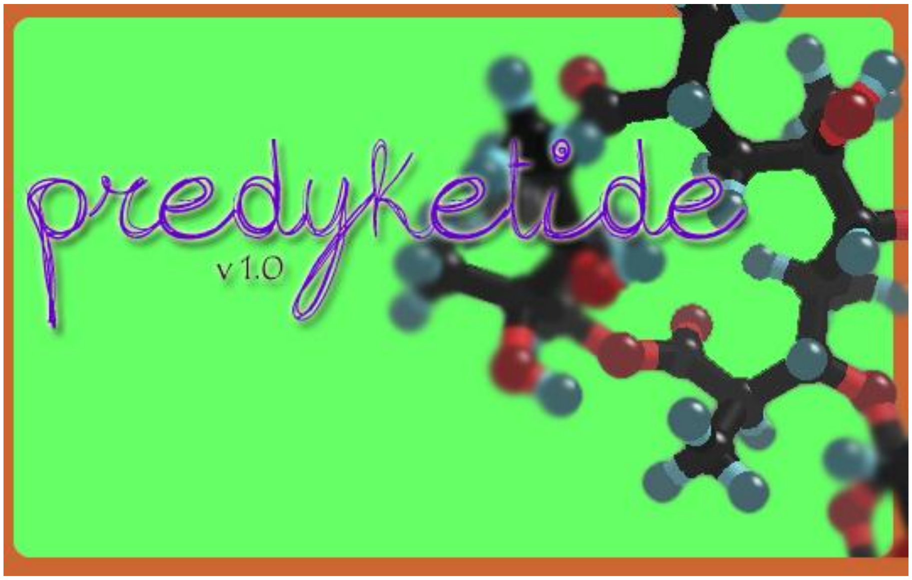


Figure 38 Welcome screen, *Predyketide*

As soon as the application is executed, a welcome screen (Figure 38) is loaded. After all the variables have been successfully initialized, the splash screen disappears and is replaced by an empty workspace (Figure 39). The said workspace contains a blank canvas and only the **Starter** button is enabled. This is done to assure that the first module submitted by the user is indeed a starter polyketide unit. The available items on the two menus, **File** and **Options**, are shown in Figure 40 and Figure 41. The first menu contains the following commands: (1) create of a new biosynthesis project, (2) import a saved polyketide synthesis in .pksf format, (3) export project as text file, (4) render biosynthesis cartoon as JPG image, (5) change reference data – both synthetic and natural product, and (6) exit the application. On the other hand, the **Options** menu allows users to display the help (Figure 42) and about (Figure 43) windows.

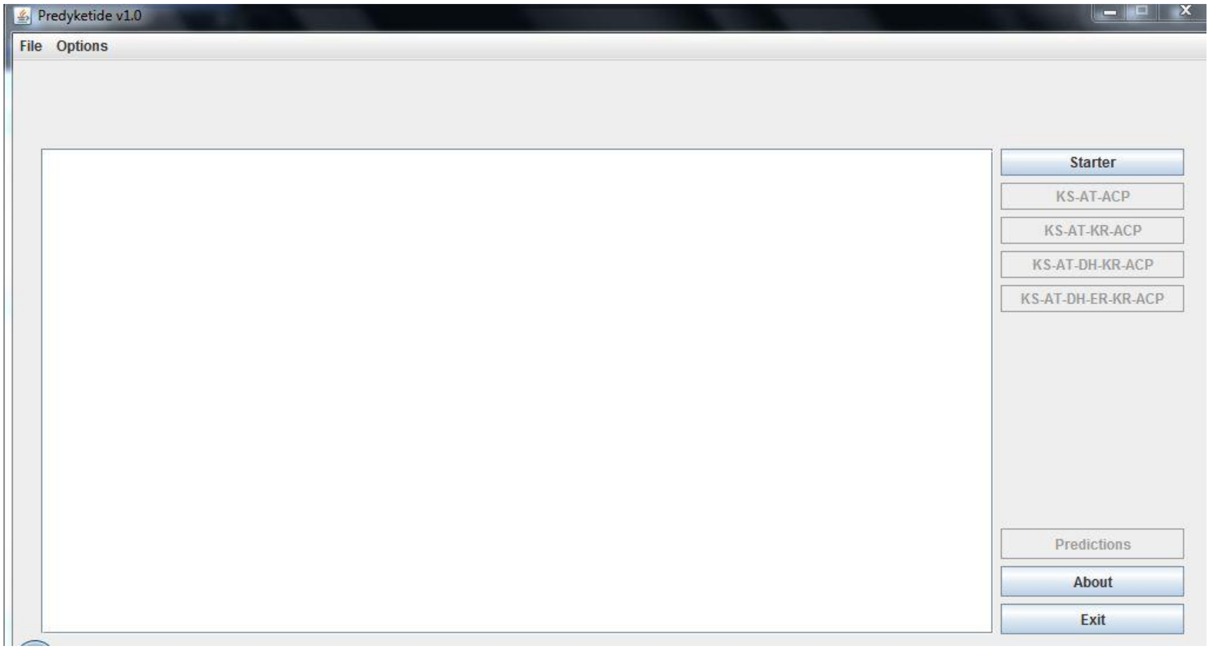


Figure 39 An empty workspace, *Predyketide*

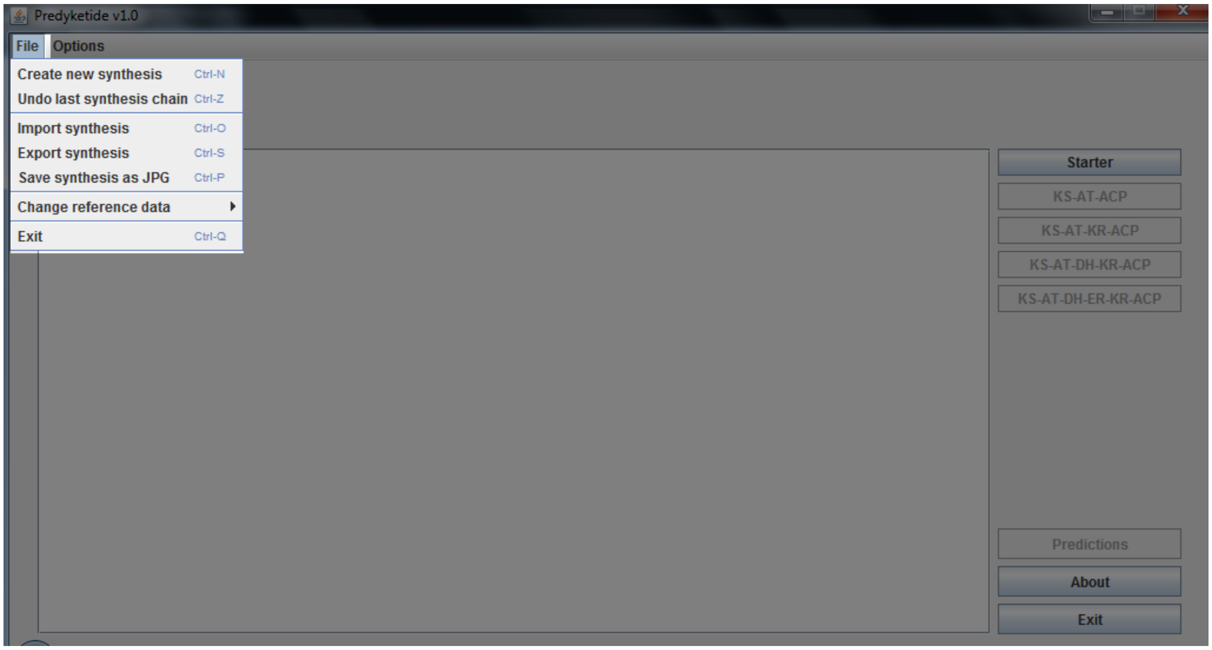


Figure 40 Items on the File menu, *Predyketide*

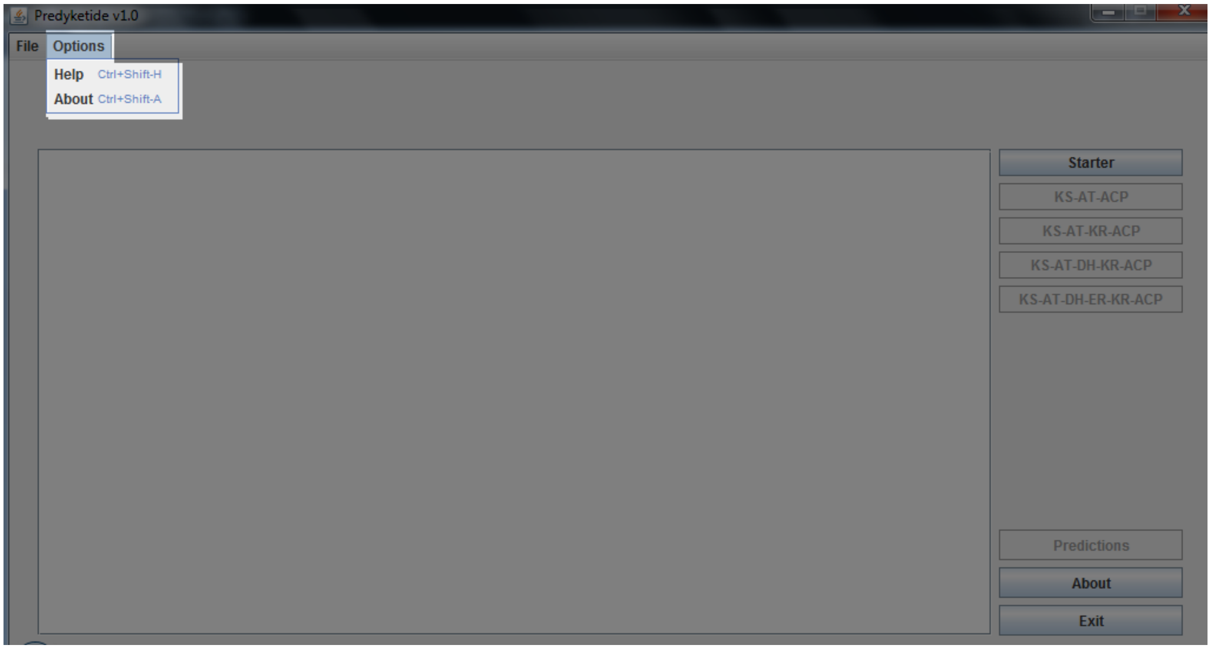
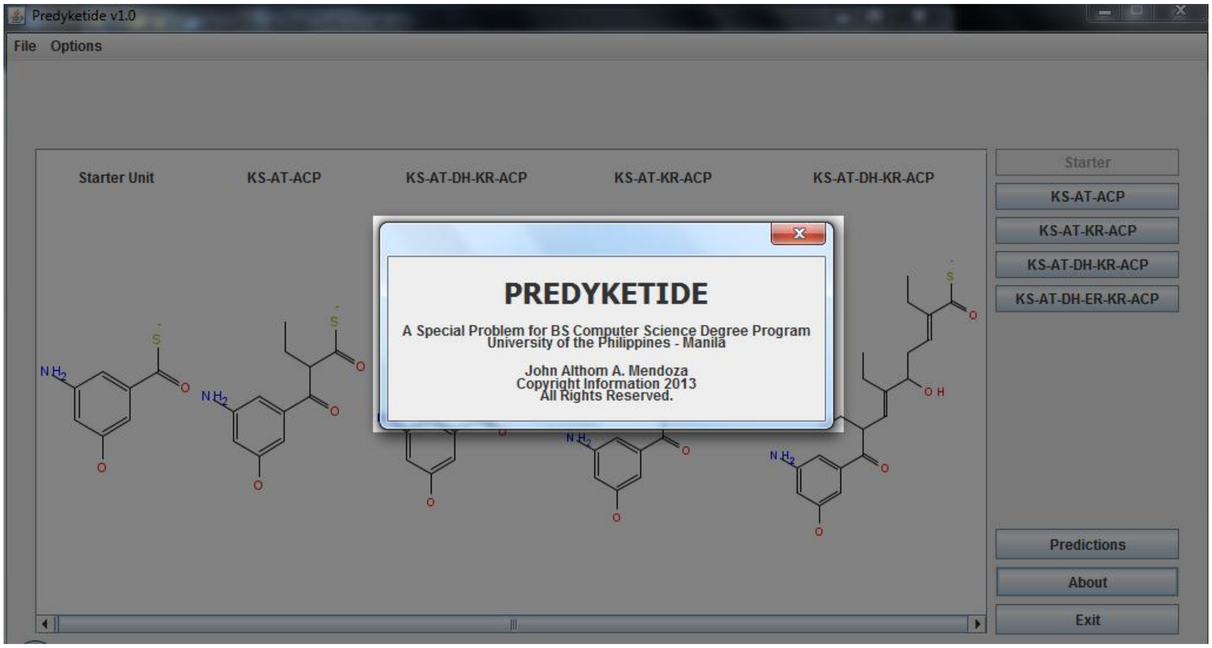


Figure 41 Items in the Options menu, *Predyketide*

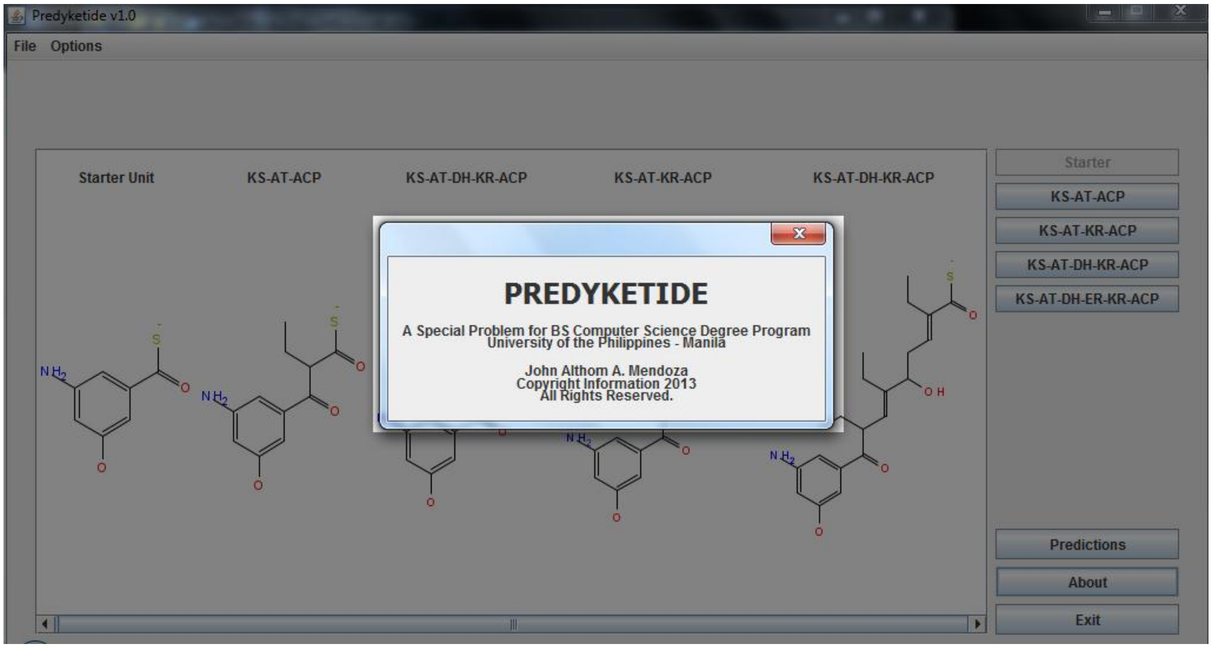


PREDYKETIDE

A Special Problem for BS Computer Science Degree Program
University of the Philippines - Manila

John Althom A. Mendoza
Copyright Information 2013
All Rights Reserved.

Figure 42 Help window, *Predyketide*



PREDYKETIDE

A Special Problem for BS Computer Science Degree Program
University of the Philippines - Manila

John Althom A. Mendoza
Copyright Information 2013
All Rights Reserved.

Figure 43 About window, *Predyketide*

Clicking the **Starter** button triggers an event to display all available loading units (Figure 44). Table 3 presents all the known loading units [15] included in the system.

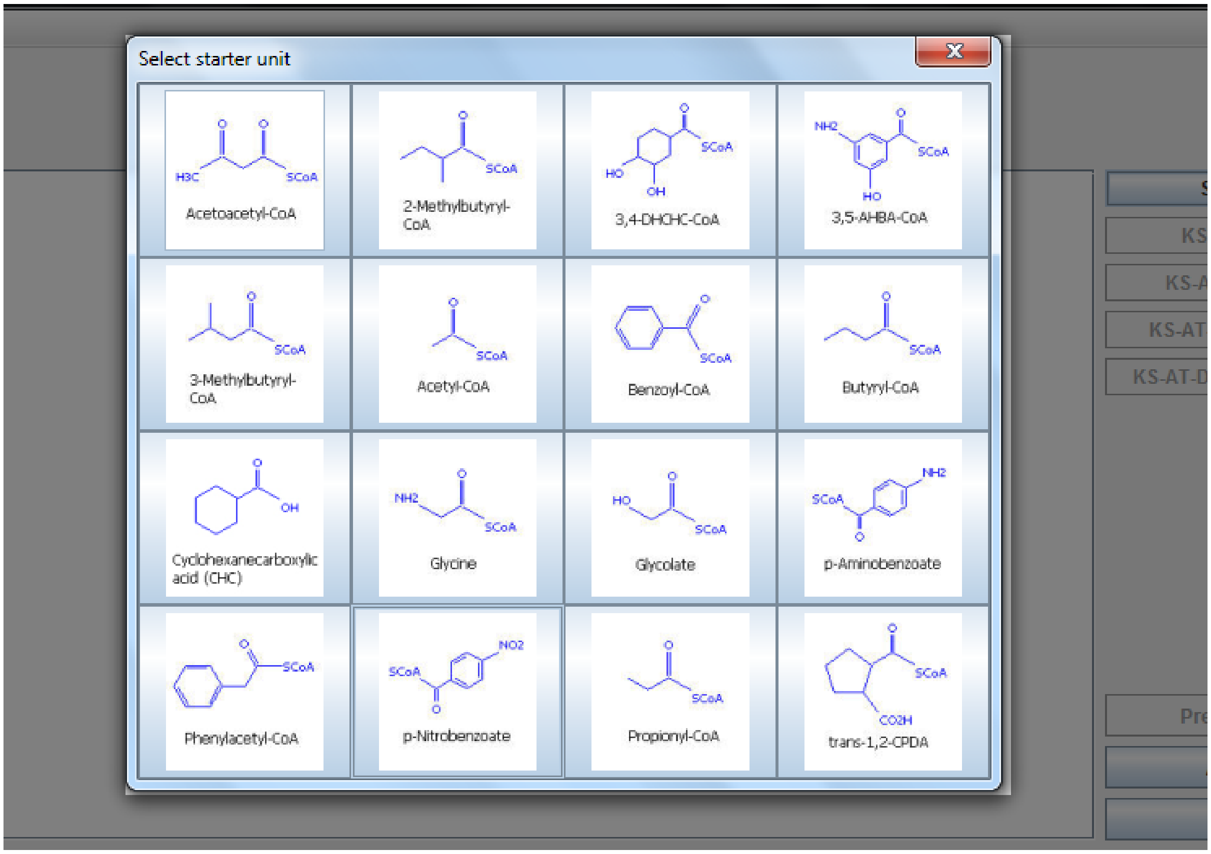
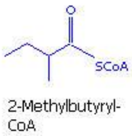
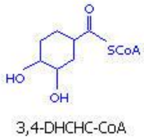
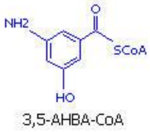
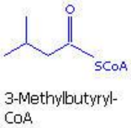
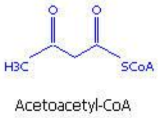


Figure 44 Selection of Starter Units, *Predyketide*

	Starter Unit Name		Starter Unit Name
	2-Methylbutyryl-CoA		3,4-DHCHC-CoA
	3,5-AHBA-CoA		3-Methylbutyryl-CoA
	Acetoacetyl-CoA		Acetyl-CoA

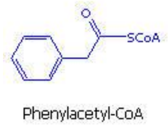
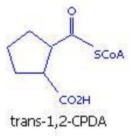
 Benzoyl-CoA	Benzonyl-CoA	 Butyryl-CoA	Butyryl-CoA
 Cyclohexanecarboxylic acid (CHC)	Cyclohexane carboxylic acid (CHC)	 Glycine	Glycine
 Glycolate	Glycolate	 p-Aminobenzoate	p-Aminobenzoate
 Phenylacetyl-CoA	Phenylacetyl-CoA	 p-Nitrobenzoate	p-Nitrobenzoate
 Propionyl-CoA	Propionyl-CoA	 trans-1,2-CPDA	Trans-1,2-CPDA

Table 3 Listing of Known Starter Units

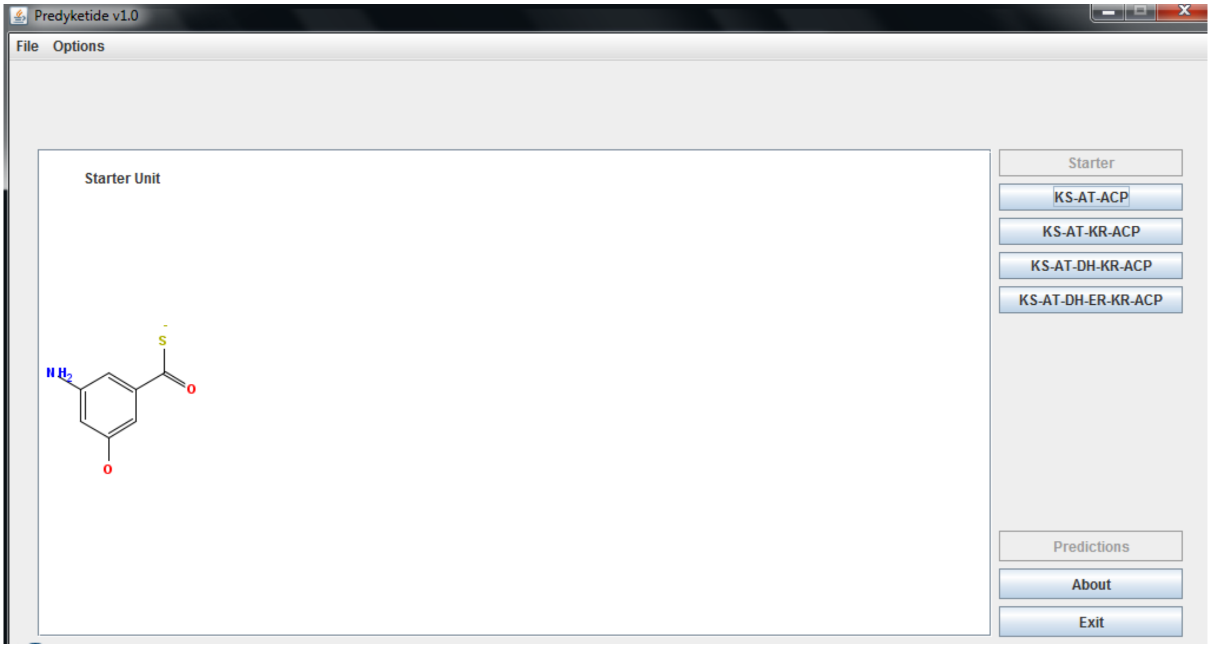


Figure 45 A starter unit is selected, *Predyketide*

The biosynthesis canvas is updated immediately after a unit is selected. An instance of the event is shown in Figure 45. The **Starter** button is then disabled to ensure a single loading unit. The extension buttons – **KS-AT-ACP**, **KS-AT-KR-ACP**, **KS-AT-DH-KR-ACP** and **KS-AT-DH-ER-KR-ACP** – are activated signaling permission for chain elongation. The modifications of these enzymes are summarized in Figure 9. Similar to the **Starter** button, any **extension** button displays a window of recognized extender units (Figure 46). These units exploited from ASMPKS [15] are listed in Table 4. The elongation process is required to be at least three extenders to enable the **Predict** button, and limited to 15. A sample biosynthesis chain cartoon is presented in Figure 20.

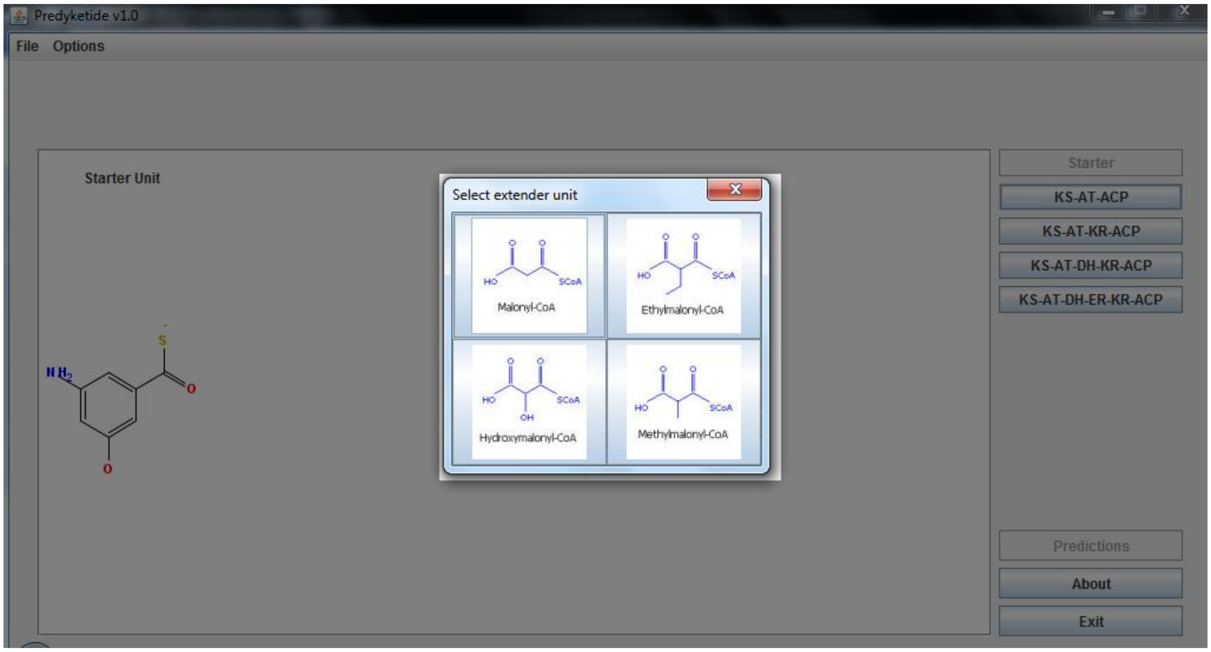


Figure 46 Extender Unit Window, *Predyketide*

	Extender Unit Name		Extender Unit Name

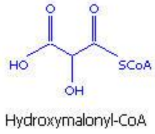
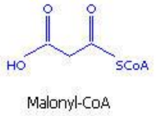
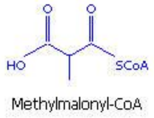
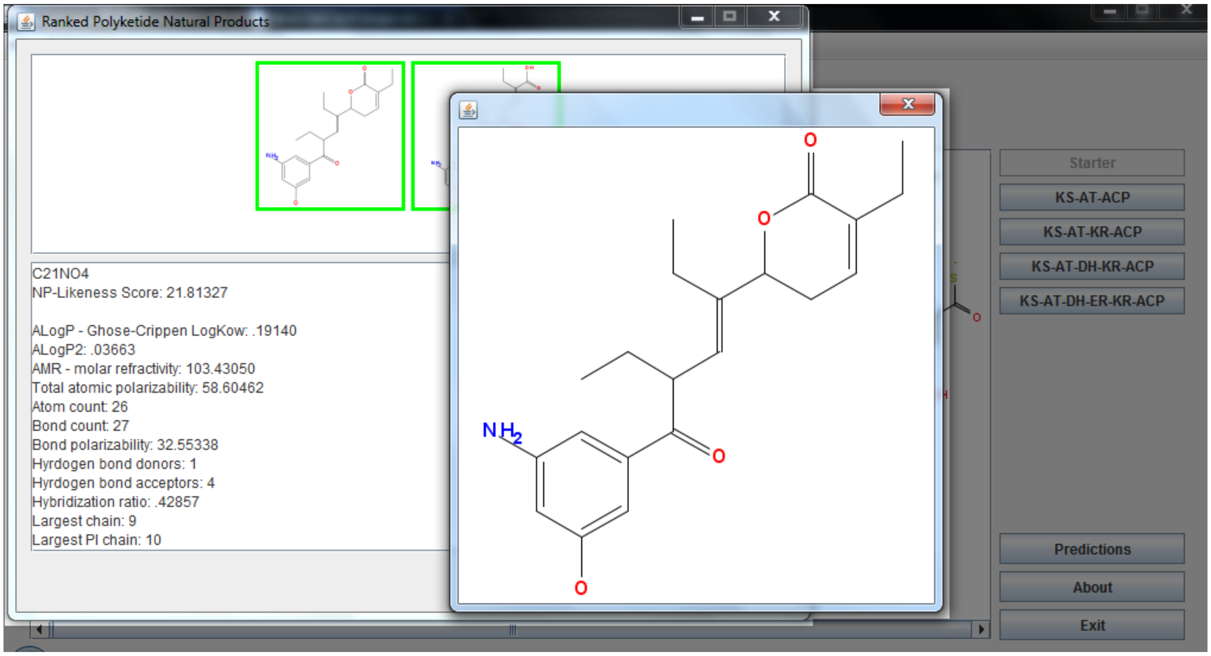
	Ethylmalonyl-CoA		Hydroxymalonyl-CoA
	Malonyl-CoA		Methylmalonyl-CoA

Table 4 Listing of Known Extender Units

Once the minimum units are included in the synthesis, clicking the **Predict** button shows the ranked list of predicted polyketide products along with some properties. A single click on a structure image displays these properties; two clicks open a window with its enlarged image. (See Figure 47)

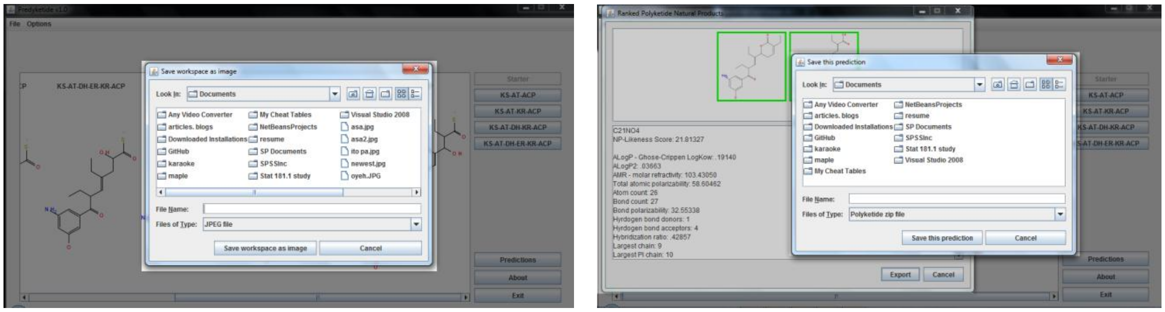


C21NO4
NP-Likeness Score: 21.81327

ALogP - Ghose-Crippen LogKow: .19140
ALogP2: .03663
AMR - molar refractivity: 103.43050
Total atomic polarizability: 58.60462
Atom count: 26
Bond count: 27
Bond polarizability: 32.55338
Hyrdogen bond donors: 1
Hyrdogen bond acceptors: 4
Hybridization ratio: .42857
Largest chain: 9
Largest PI chain: 10

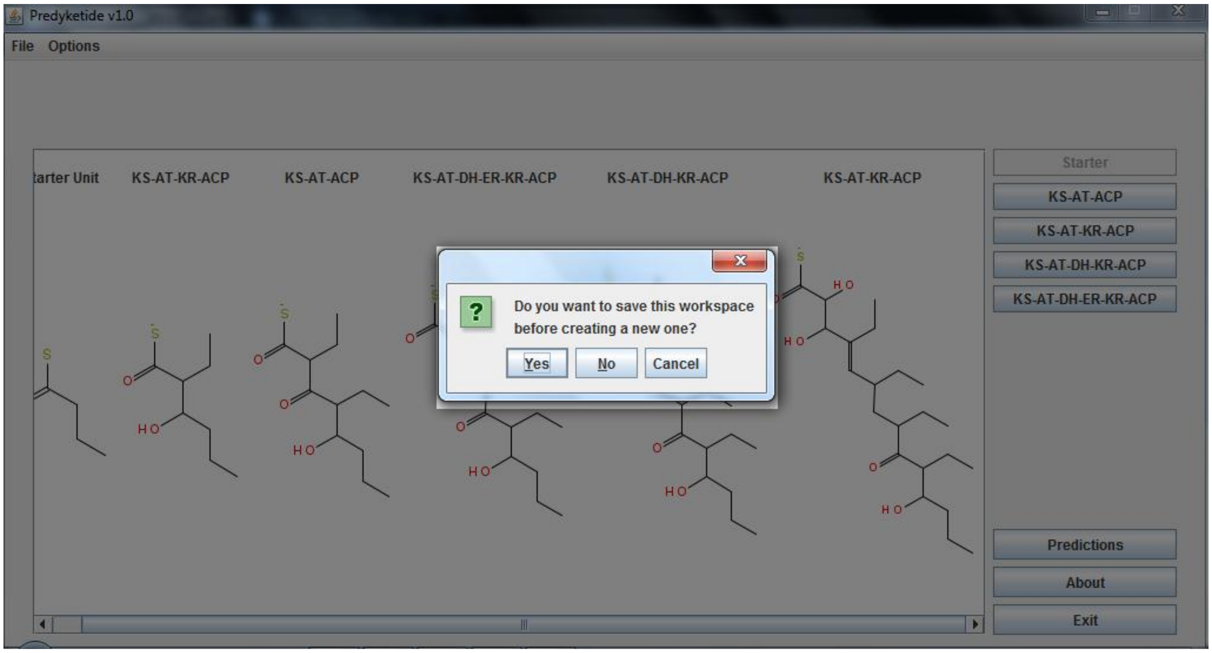
Figure 47 Ranked Prediction Panel, *Predyketide*

The biosynthesis cartoon can be exported as a text file format with extension ***pksf*** and a **JPG** image. A prediction can be saved as a zipped folder containing the two previously mentioned files, SMILES and MolFile representation, a text file of the descriptors, and the natural product structure image. These functionalities are shown in Figure 48. A notification message (Figure 49) is issued on any of the following events: success of exporting the synthesis or a novel structure, addition of more than 15 extender units, and closing the application.



```
C21NO4
NP-Likeness Score: 21.81327

ALogP - Ghose-Crippen LogKow: .19140
ALogP2: .03663
AMR - molar refractivity: 103.43050
Total atomic polarizability: 58.60462
Atom count: 26
Bond count: 27
Bond polarizability: 32.55338
Hyrdogen bond donors: 1
Hyrdogen bond acceptors: 4
Hybridization ratio: .42857
Largest chain: 9
Largest Pi chain: 10
```

Figure 48 Export functions. (Left) for the synthesis. (Right) for a prediction, *Predyketide*



Do you want to save this workspace before creating a new one?

Figure 49 Example of a notification message event, *Predyketide*

Predyketide uses the ChemAxon MarvinBeans [27] in the rendering of the predicted natural product. It also exploits the library's legacy functionalities such as 3D-clean rendering and rotation (Figure 50), ball-and-stick/wireframe/space fill display (Figure 51), and others.

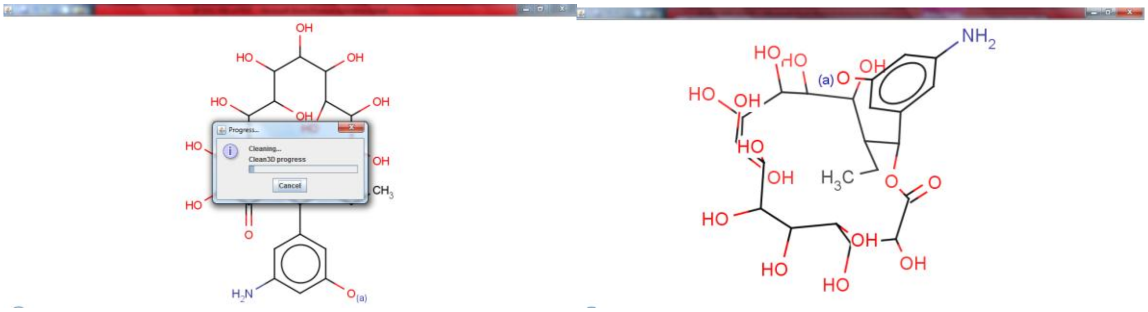


Figure 50 Before and after 3D Clean of MarvinBeans, *Predyketide*

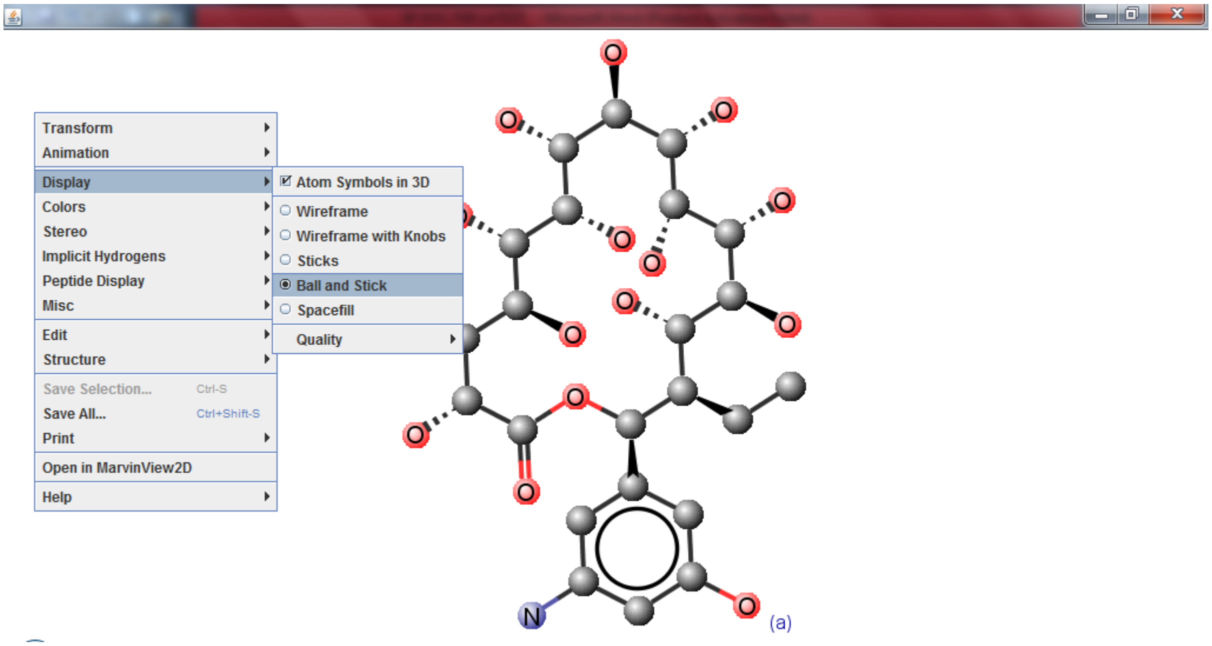
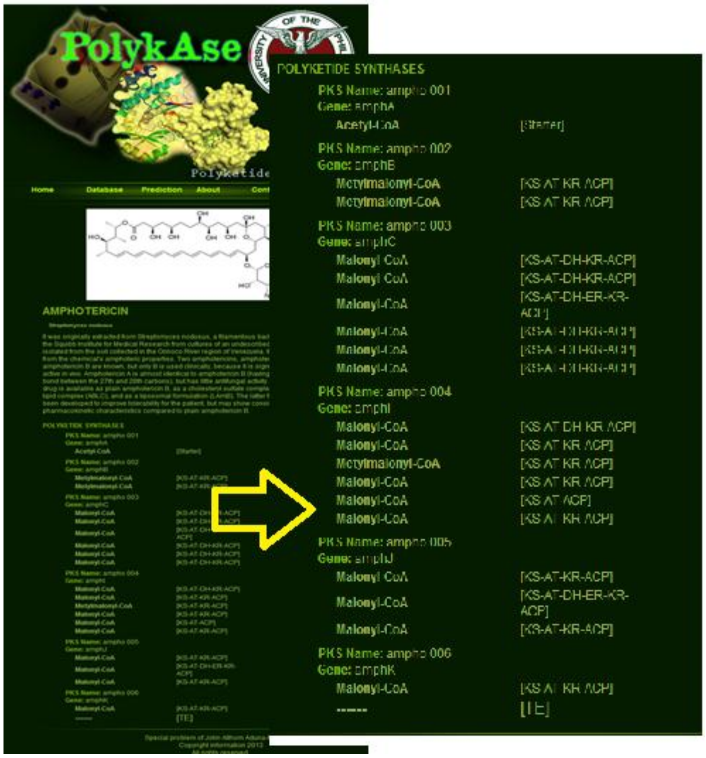


Figure 51 Ball and Stick Display Function of MarvinBeans, *Predyketide*

B. Polykase

Polykase is composed of two main parts: (1) a recreated database of naturally occurring Type I modular polyketide natural product, and (2) a novel polyketide research community. The system can be browsed at the web link agila.upm.edu.ph/~jamendoza/polykase.

The system is currently at beta mode (i.e. the system is made available before it is finished to allow testing and feedback). It currently holds 10 PKNPs but records will be added continuously to keep it up to date. A sample PKNP record page is shown in Figure 52.



POLYKETIDE SYNTHASES

PKS Name: ampho 001
Gene: amphA

Acetyl-CoA	[Starter]

PKS Name: ampho 002
Gene: amphB

Metylmalonyl-CoA	[KS AT KR ACP]
Metylmalonyl-CoA	[KS AT KR ACP]

PKS Name: ampho 003
Gene: amphC

Malonyl CoA	[KS-AT-DH-KR-ACP]
Malonyl CoA	[KS-AT-DH-KR-ACP]
Malonyl-CoA	[KS-AT-DH-ER-KR-ACP]
Malonyl-CoA	[KS-AT-DH-KR-ACP]
Malonyl-CoA	[KS-AT-DH-KR-ACP]
Malonyl-CoA	[KS-AT-DH-KR-ACP]

PKS Name: ampho 004
Gene: amphI

Malonyl-CoA	[KS AT DH KR ACP]
Malonyl-CoA	[KS AT KR ACP]
Metylmalonyl-CoA	[KS AT KR ACP]
Malonyl-CoA	[KS AT KR ACP]
Malonyl-CoA	[KS AT ACP]
Malonyl-CoA	[KS AT KR ACP]

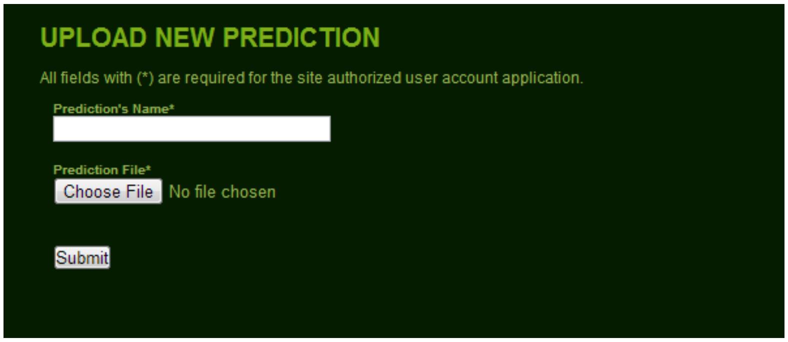
PKS Name: ampho 005
Gene: amphJ

Malonyl CoA	[KS-AT-KR-ACP]
Malonyl-CoA	[KS-AT-DH-ER-KR-ACP]
Malonyl-CoA	[KS-AT-KR-ACP]

PKS Name: ampho 006
Gene: amphK

Malonyl-CoA	[KS AT KR ACP]
------	[TE]

Figure 52 Screenshot of a PKNP record, with the PKS sequence magnified, *Polykase*


UPLOAD NEW PREDICTION

All fields with (*) are required for the site authorized user account application.

Prediction's Name*

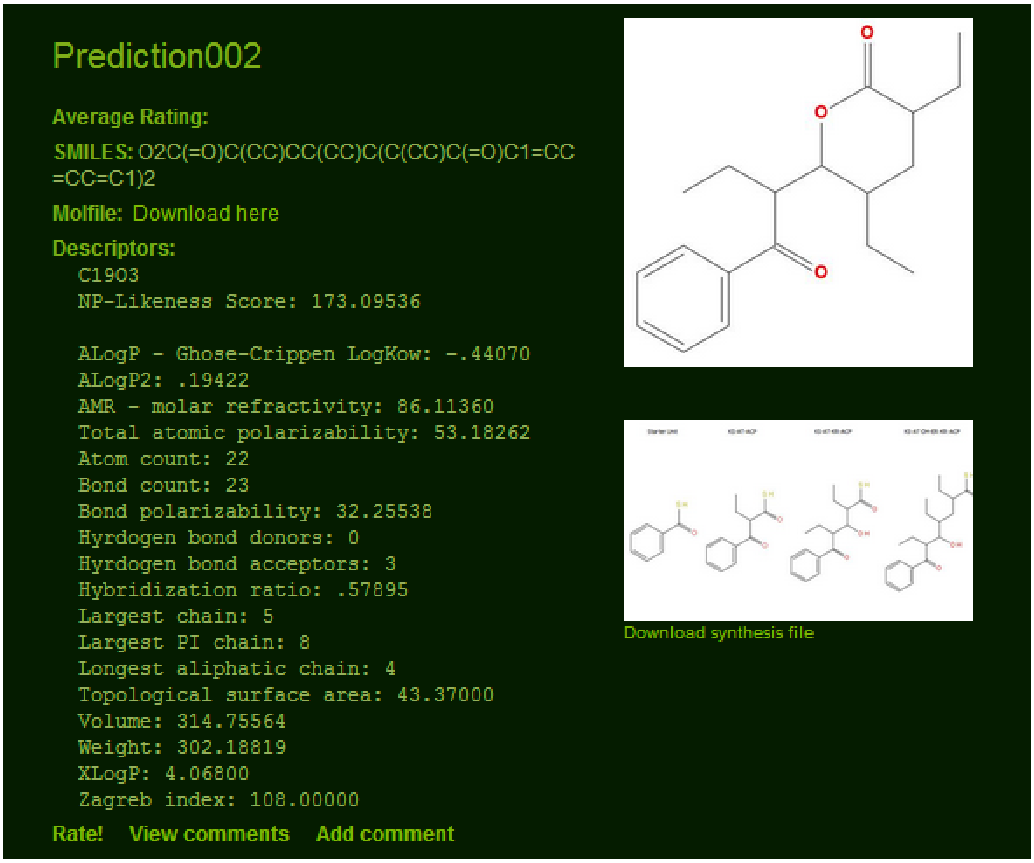
Prediction File*

Choose File No file chosen

Submit

Figure 53 Polyketide prediction form, *Polykase*

The novel polyketide research community component mentioned earlier refers to *Polykase*'s feature to share predictions between different users. Any registered user of the system can upload a maximum of three predictions. This limitation avoids the web server and database to congest. Users are provided with a form (Figure 53), which requires the polyketide zip file (this is the exported zipped file from *Predyketide*) and name of the prediction prior successful submission. Other users can rate and give reviews or comments to any submitted polyketide structure. A web page (Figure 54) containing the novel product name, the SMILES and MolFile representations, the molecule structure, the biosynthesis cartoon and file (pksf), average rating, and comments (if available) is dedicated to a single prediction.



Prediction002

Average Rating:

SMILES: O2C(=O)C(CC)CC(CC)C(C(CC)C(=O)C1=CC=CC=C1)2

Molfile: Download here

Descriptors:

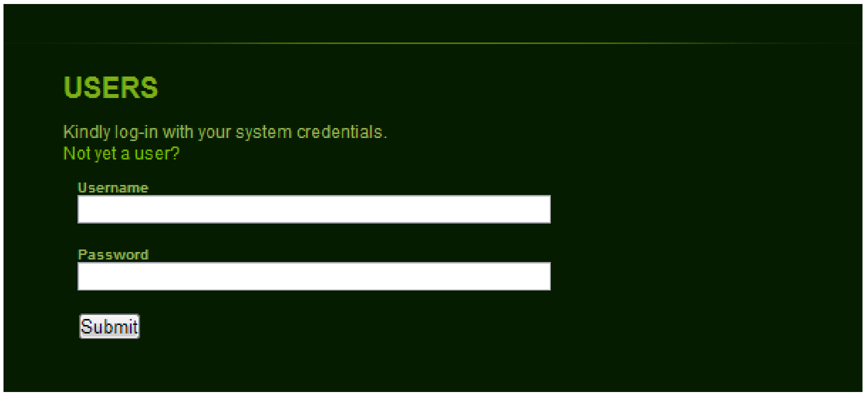
```
C19O3
NP-Likeness Score: 173.09536

ALogP - Ghose-Crippen LogKow: -.44070
ALogP2: .19422
AMR - molar refractivity: 86.11360
Total atomic polarizability: 53.18262
Atom count: 22
Bond count: 23
Bond polarizability: 32.25538
Hyrdogen bond donors: 0
Hyrdogen bond acceptors: 3
Hybridization ratio: .57895
Largest chain: 5
Largest PI chain: 8
Longest aliphatic chain: 4
Topological surface area: 43.37000
Volume: 314.75564
Weight: 302.18819
XLogP: 4.06800
Zagreb index: 108.00000
```

Download synthesis file

Rate! **View comments** **Add comment**

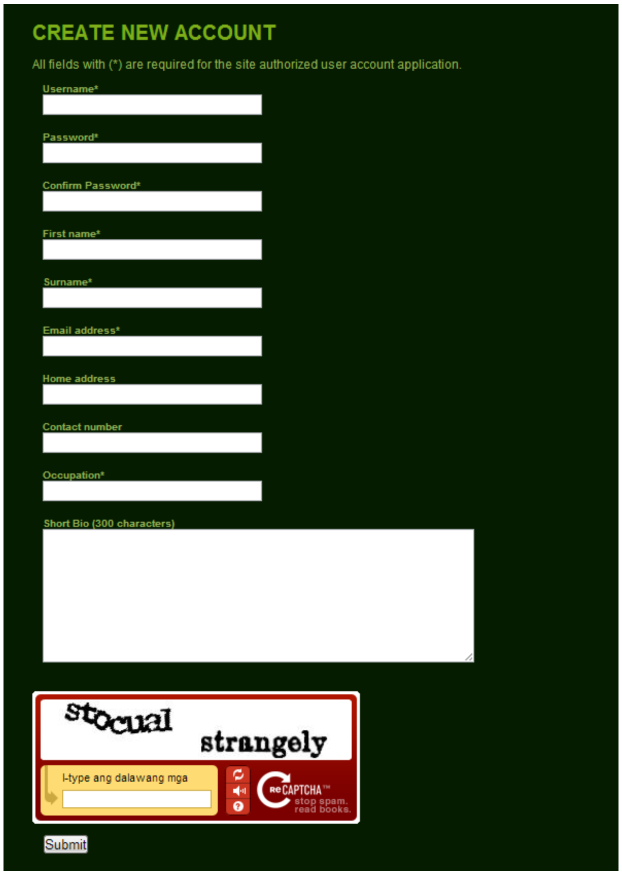
Figure 54 Screenshot of a predicted structure, *Polykase*

Furthermore, this web interface includes a user and data management tool. This allows the system administrator to add, edit, and delete records from the system even if he/she has no technical backgrounds. The said feature is a collection of forms and validation rules. Dashboards are also provided to the administrator and the registered users for moderation and management of data within their scopes. The images below are the forms and dashboards available in *Polykase.*


USERS

Kindly log-in with your system credentials.
Not yet a user?

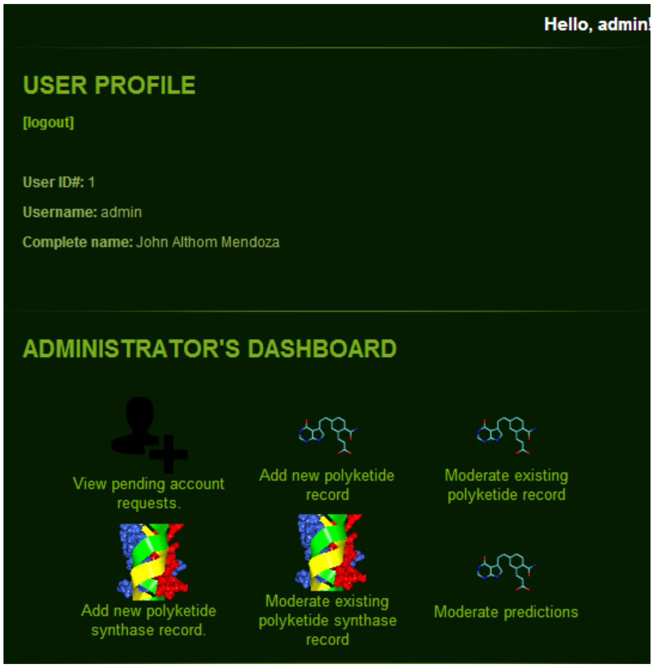
Figure 55 User Login, *Polykase*


CREATE NEW ACCOUNT

All fields with (*) are required for the site authorized user account application.

Figure 56 User creation/application form, *Polykase*

Upon visiting the web site, the system redirects the user to its current homepage – User Login page (Figure 55). Guest users who desire to have authorized access on the system can click the link *“Not yet a member?”*, which loads to the user application form (Figure 56). To avoid robots from spamming the database, a re-CAPTCHA validation is included. Successful login allows the users to view a dashboard containing all the activities permitted for their roles. The administrator dashboard (Figure 57) provides links to (a) viewing pending account requests, (b) addition of PKNP, (c) moderation of PKNP records, (d) addition of PKS record, (e) moderation of PKS records, and (f) moderation of novel product submissions.


Hello, admin!

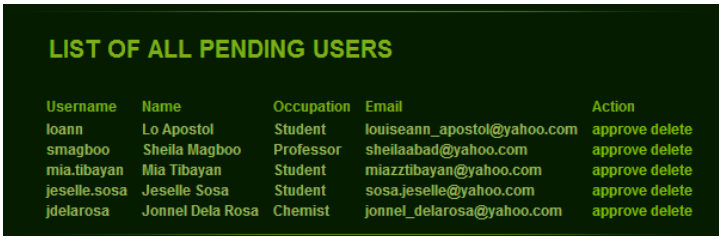
USER PROFILE

[logout]

User ID#: 1
Username: admin
Complete name: John Althom Mendoza

ADMINISTRATOR'S DASHBOARD

Figure 57 Administrator Dashboard, *Polykase*

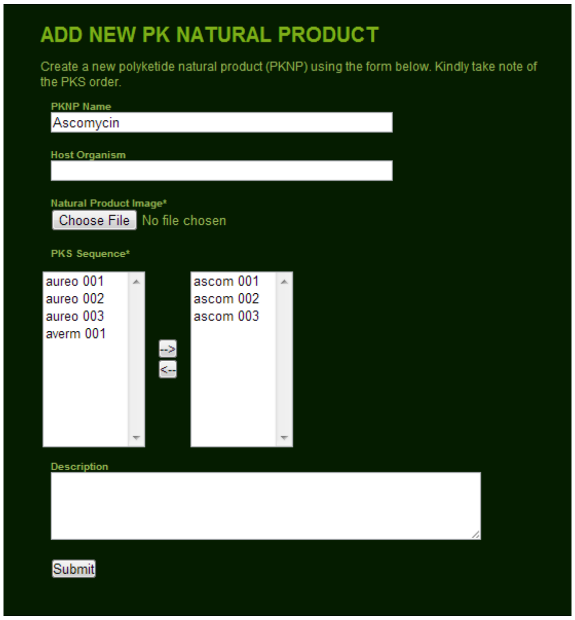


LIST OF ALL PENDING USERS

Username	Name	Occupation	Email	Action
loann	Lo Apostol	Student	louiseann_apostol@yahoo.com	approve delete
smagboo	Sheila Magboo	Professor	sheilaabad@yahoo.com	approve delete
mia.tibayan	Mia Tibayan	Student	miazztibayan@yahoo.com	approve delete
jeselle.sosa	Jeselle Sosa	Student	sosa.jeselle@yahoo.com	approve delete
jdelarosa	Jonnel Dela Rosa	Chemist	jonnel_delarosa@yahoo.com	approve delete

Figure 58 List of pending account applications, *Polykase*

The list of pending users is populated once a successful account application was made. The list provides the administrator options to either delete or approve a specific application (Figure 58). The form for the creation of a new natural product record is shown in Figure 59. It requires the name, host organism, image, PKS sequence (must be in order on the right select box), and description (optional) about the natural product. A polyketide synthase is made using the form shown in Figure 60. The said form requires the PKS name, the gene where it was obtained, and the modules and domain sequences involved.



ADD NEW PK NATURAL PRODUCT

Create a new polyketide natural product (PKNP) using the form below. Kindly take note of the PKS order.

PKNP Name: Ascomycin

Host Organism:

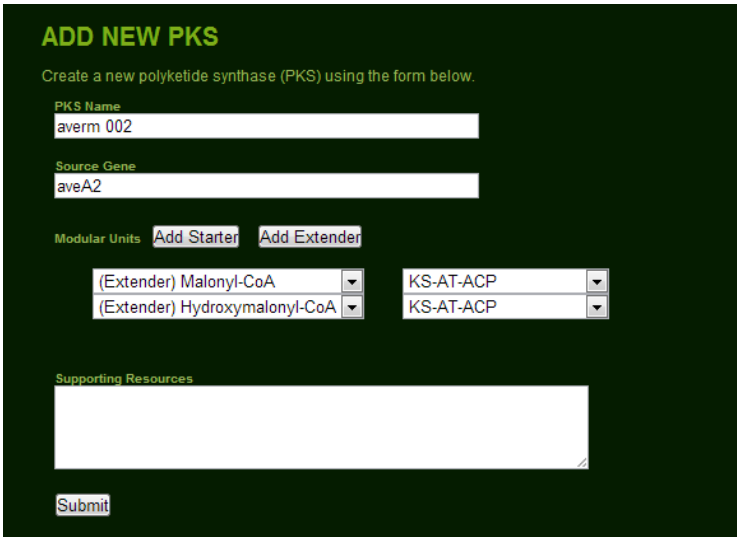
Natural Product Image*: No file chosen

PKS Sequence*:

aureo 001	ascom 001
aureo 002	ascom 002
aureo 003	ascom 003
averm 001	

Description:

Figure 59 Form for creation of new PKNP record, *Polykase*



ADD NEW PKS

Create a new polyketide synthase (PKS) using the form below.

PKS Name: averm 002

Source Gene: aveA2

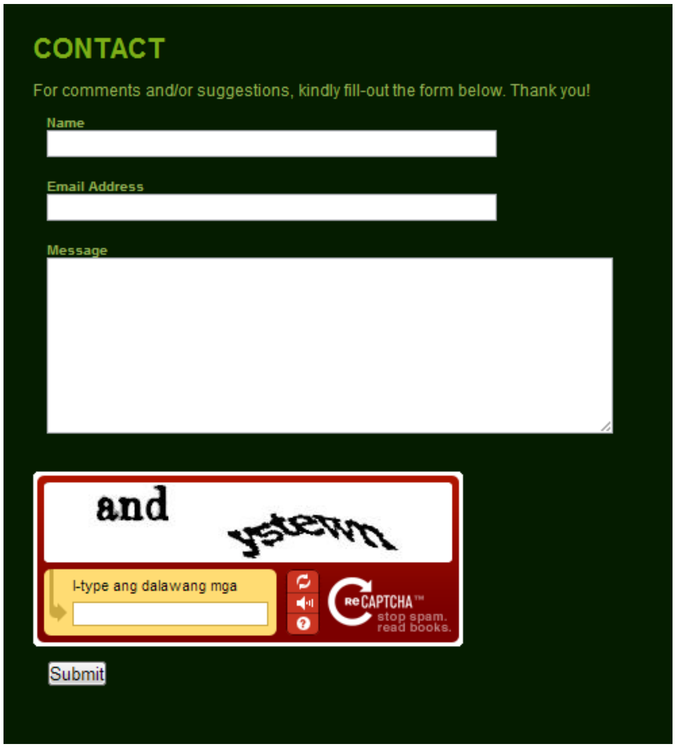
Modular Units:

(Extender) Malonyl-CoA	KS-AT-ACP
(Extender) Hydroxymalonyl-CoA	KS-AT-ACP

Supporting Resources:

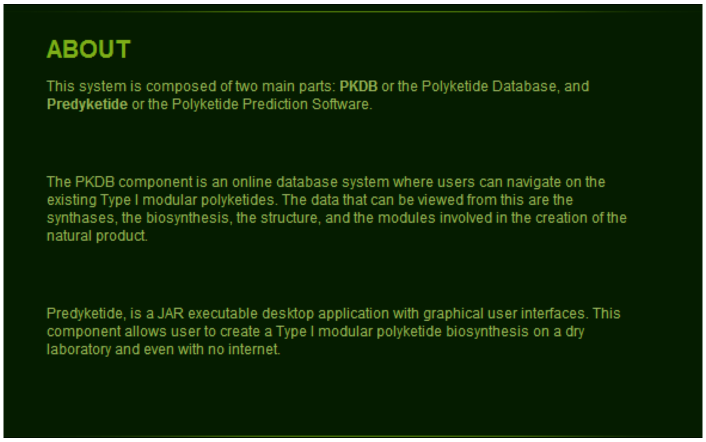
Figure 60 Add new PKS form, *Polykase*

Figure 61 and Figure 62 presents the contact form and the about page of the system. The former page allows all users to send a feedback or comment to the administrator. The messages will be sent to the administrator’s email address. A CAPTCHA confirmation was included to avoid robot or spam attacks. The about page is a static page containing a short description about *Polykase* and *Predyketide*.


CONTACT

For comments and/or suggestions, kindly fill-out the form below. Thank you!

Figure 61 Contact form, *Polykase*


ABOUT

This system is composed of two main parts: **PKDB** or the Polyketide Database, and **Predyketide** or the Polyketide Prediction Software.

The PKDB component is an online database system where users can navigate on the existing Type I modular polyketides. The data that can be viewed from this are the synthases, the biosynthesis, the structure, and the modules involved in the creation of the natural product.

Predyketide, is a JAR executable desktop application with graphical user interfaces. This component allows user to create a Type I modular polyketide biosynthesis on a dry laboratory and even with no internet.

Figure 62 About page, *Polykase*

VI. Discussion

Drug discovery has become a vital part of human existence. Knowing the fact that a number of medicines are derived from polyketides [20, 21], the creation of a computational tool –*Predyketide* – provides the researchers to explore PKNP possibilities even outside a wet laboratory. The stand-alone nature of the application allows the researchers to use the system even without internet connection. Its portability allows the operation of the system onto the user's workstation and personal machine. The continuation of an unfinished project is made easier by the retrievable synthesis file design. With the help of this tool, costs and labor for polyketide experiments may be minimized. [13]

The exported novel product includes two of the most common molecule representation format – SMILES and MolFile. [65] These two representations bridge *Predyketide* with other chemical software. Thus, the generated results can be used in extended research. One good example is Molinspiration's Bioactivity calculator, which takes in a SMILE string as the input and returns six bioactivity descriptors: GPCR ligand, ion channel modulator, kinase inhibitor, nuclear receptor ligand, protease inhibitor, and enzyme inhibitor. [66]

Another useful material is the file containing the descriptor values computed in each predicted compound. These 18 QSAR values can be useful in other researches like the classification of drug or nondrug.

The two issues involved in *Predyketide* are the selection of the reference data, and the inaccurate depiction of structure for large rings. Only 100 of both synthetic and natural product sets are included as reference data due to memory constraints. However, the user can change these references to a larger size during run-time to obtain better results. The usage of the CHEMBL (for the natural product data sets) and ZINC (for the synthetic data set) is based on the previous study of NP-likeness computation. [17, 44] Meanwhile, two libraries are used for the rendering of the chemical structures : CDK-JChemPaint for the chain elongation and ChemAxon MarvinBeans for the natural products. The reason for such is the former produces an inaccurate depiction (Figure 63, left) of the natural product structure, and the latter produces too much folding in the chain elongation. The predicted compound generated by *Predyketide* is limited to the chain elongation. Inclusion of post-processing techniques is not yet included. Also, the chain elongation process is limited to a maximum of 15 extender units.

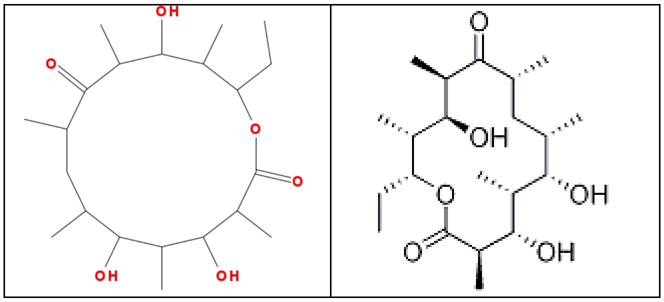


Figure 63 Comparison of Predyketide Sketch and Actual Sketch of *Erythromycin*

On the other hand, *Polykase* provides avenue for the scientists in the field of polyketide research to collaboratively work with their colleagues by using the reviews and ratings section. This section where the researchers can share insights and comments is available in each novel product record. However, the Type I PKNP library interface currently lacks content due to the limited free sources available on the internet. Also, all the contents of the library are mainly dependent on the system administrator and his/her resources.

Polykase's PKNP Database lacks content due to the limited sources available on the net but; this interface can be enhanced by uploading more records. Since this component only serves as a reference material, inclusion of all polyketide types can be done. Also, the web-based design of *Polykase* makes it accessible even in mobile devices. Allowing regular users to submit a known polyketide natural product is a huge improvement of the system.

VII. **Conclusion**

Predyketide is a stand-alone desktop computational tool that allows the prediction and ranking of modular Type I polyketide natural products. The ranking is based on the NP-likeness grade, which is the accumulated Tanimoto fragment scores. The application can be downloaded at http://sourceforge.net/projects/predyketide/. The system is developed to help drug researchers to simulate and to visualize a polyketide synthesis. Though the PKS Assembly exhibits the same chain elongation functionality, *Predyketide* allows the prediction of the resulting compound. The prediction outputs can be shared using the *Polykase* web interface.

The project is currently at its beta stage. Once it gains popularity among different researchers, both *Predyketide* and *Polykase* can aid the drug development industry in the discovery of new medicines. Due to its evident significance, the usage of the systems in basic polyketide drug research is expected to grow. Future developments and maintenance of the softwares are also anticipated in order to cater the increasing need in the field.

VIII. Recommendations

It is highly suggested that the next version of *Predyketide* drops the usage of the CDK library and focus on the usage and utilization of the ChemAxon MarvinBeans alone. A number of functionalities such as database check and complete QSAR computation can be exploited from the said library. Also, the MarvinBeans can also provide ways to include post-processing techniques of the novel products. The amount of extender units for a chain elongation process (currently at max of 15 units) can be increased for the succeeding versions.

The usage of better techniques such as Support Vector Machines and Neural Networks in predicting and ranking the structures can also be done to improve the system. Ranking the novel products other than their natural product-likeness scores (toxicity level, drug-likeness, etc.) can also be done in the future. Moreover, a larger volume and a more standard reference dataset can help in the generation of a more accurate ranking and predictions.

Polykase can be further improved by populating the records included in the system. Inclusion of other polyketide types can provide ease for the researchers since a single database system is accessed for all PK types. A more user-friendly and dynamic interface for both the administrator and the registered users can be also be helpful.

IX. Bibliography

[1] Newman, D. J., & Cragg, G. M. (2007). Natural Products as Sources of New Drugs over the Last 25 Years. *Journal of Natural Products* , 461-477.

[2] Hertweck, C. (2009). The biosynthetic logic of polyketide diversity. *Angew. Chem. Int. Ed. Engl , 48*, 4688-4716.

[3] Yadav, G., Gokhale, R. S., & Mohanty, D. (n.d.). *A Database of Modular Polyketide Synthases*. Retrieved September 7, 2012, from http://www.nii.res.in/pksdb.html

[4] Staunton, J., & Weissman, K. (2001). Polyketide biosynthesis: a millenium review. *Natural Product Reports , 18* (4), pp. 380-416.

[5] Fischbach, M., & Walsh, C. (2006). Assembly-line enzymology for polyketide and nonribosomal. *Chemical Reviews , 106*, 3468-3496.

[6] Smith, S., & Tsai, S. (2007). The type I fatty acid and polyketide synthases: A tale of two megasynthases. *Natural Products Report , 24*, 1041-1072.

[7] *The World of Polyketide.* (n.d.). Retrieved September 9, 2012, from http://www.nii.ac.in/~pksdb/polyketide.html

[8] Shen, B. (2003). Polyketide biosynthesis beyond the type I, II and III polyketide synthase. *Current Opinion in Chemical Biology , 7* (2), 285-295.

[9] Wiesmann, K. C. (1995). Polyketide synthesis in vitro on a modular polyketide synthase. *Chemistry & Biology , 2*, 583-589.

[10] Shen, B. (1996). Deciphering the mechanism for the assembly of aromatic. 6600-6604.

[11] Collie, J., & Myers, W. (1893). The formation of orcinol and other condensation products from dehydroacetic acid. *Journal of Chemical Society , 63.*

[12] Isukapalli, S. S., & Georgopoulos, P. G. (2001). *Computational Methods for the Efficient Sensitivity and Uncertainty Analysis of Models for Environmental and Biological Systems.* Computational Chemodynamics Laboratory.

[13] Kim, M., Kwon, S. J., & Dordick, J. S. (2009). In Vitro Precursor-Directed Synthesis of Polyketide Analogs with Coenzyme A Regeneration for the Development of Antiangiogenic Agents. *Org Lett* , 3806–3809.

[14] Pace, N. (1997). A molecular view of microbial diversity and the biosphere. *Science , 276*, 734–740.

[15] Tae, H., Nam, H., Song, J., & Park, K. (2007). A Web Based Polyketide Analysis System for Modular Polyketide Synthases. South Korea.

[16] Burbidge, R., Trotte, M., Buxton, B., & Holden, S. (2001). Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Computers and Chemistry , 26*, 5-14.

[17] Ertl, P., Roggo, S., & Schuffenhauer, A. (2008). Natural Product-likeness Score and Its Application for Prioritization of Compound Libraries. *J Chem Inf Model , 48*, 68-74.

[18] *SMILES tutorial*. (n.d.). Retrieved October 2, 2012, from Daylight Chemical Information Systems, Inc.: http://www.daylight.com/meetings/summerschool98/course/dave/smiles-intro.html

[19] *ML - Molecule subroutine* . (n.d.). Retrieved October 2, 2012, from http://www.acornnmr.com/NutsHelp/molfiles.htm

[20] *biotica*. (2012). Retrieved September 1, 2012, from Polyketides as Leads: http://www.biotica.com/index.php/biotica/page/36/

[21] *Scientists Solve Mystery Of Polyketide Drug Formation.* (2008, April 9). Retrieved September 6, 2012, from ScienceDaily: http://www.sciencedaily.com/releases/2008/04/080401112350.htm

[22] Li, M. H., Ung, P. M., Zajkowski, J., Garneau-Tsodikova, S., & Sherman, D. H. (2009). Automated genome mining for natural products. *BMC Bioinformatics , 10*.

[23] Thattai, M., Burak, Y., & Shraiman, B. (2007). The origins of specificity in polyketide synthase protein interactions. *PLoS Computational Biology* , 1827-1835.

[24] Gokhalea, R. S., & Khosla, C. (2000). Role of linkers in communication between protein modules. *Current Opinion in Chemical Biology , 4* (1), 22-27.

[25] Yadav, G., Gokhale, R. S., & Mohanty, D. (2003). SEARCHPKS: a program for detection and analysis of polyketide synthase domains. *Nucleic Acids Research , 31* (13), 3654–3658.

[26] Steinbeck, C., Han, Y., Kuhn, S., Horlacher, O., Luttmann, E., & Willighagen, E. (2003). The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci. , 43* (2), 493-500.

[27] ChemAxon. (2013). *Marvin was used for drawing, displaying and characterizing chemical structures, substructures and reactions, Marvin 5.11.5.* Retrieved from http://www.chemaxon.com

[28] Jinglin, L., Yunzi, L., Jung-Kul, L., & Huimin, Z. (2011). Cloning and characterization of a type III polyketide synthase. *Bioorganic & Medicinal Chemistry Letters* , 6085-6089.

[29] Birch, A., & Donovan, F. (1953). Studies in relation to biosynthesis. I. Some possible routes to derivatives of orcinol and phloroglucinol. *Australian Journal of Chemistry , 6*, 360-368.

[30] Fuji, I., Watanabe, A., Sankawa, U., & Ebizuka, Y. (2001). Identification of Claisen cyclase domain in fungal polyketide synthase WA, a naphthopyrone synthase of Aspergillus nidulans. *Chemistry & Biology , 8*, 189-197.

[31] ibioseminars (Producer). (2010). *Chaitan Khosla (Stanford) Part 1- Polyketide Biosynthesis* [Motion Picture].

[32] Watanabe, A., & Ebizuka, Y. (2004). Unprecedented Mechanism for Chain Length Determination in Fungal Aromatic Polyketide Synthases. *Chemistry & Biology , 11*, 1101-1106A.

[33] Frandsen, R. (2010, January 8). *Polyketide Synthases.* Retrieved September 10, 2012, from Rasmus Frandsen Homepage: http://www.rasmusfrandsen.dk/polyketide_synthases.htm

[34] Yi, G.-S., & Kim, J. (2012). PKMiner: a database for exploring type II polyketide synthases. *BMC Microbiology , 12*.

[35] Vijayan, M., Chandrika, S. K., & Vasudevan, S. E. (2011). PKSIIIexplorer: TSVM approach for predicting Type III polyketide synthase proteins. *Bioinformation , 6* (3), 125-127.

[36] Mallika, V., Sivakumar, K., Jaichand, S., & Soniya1, E. (2010). Kernel based machine learning algorithm for the efficient prediction of type III polyketide synthase family of proteins. *Journal of Integrative Bioinformatics , 7* (1).

[37] Jayaram, B. (n.d.). *What is genome?* Retrieved September 15, 2012, from Supercomputing Facility for Bioinformatics & Computational Biology: http://www.scfbio-iitd.res.in/research/genome.htm

[38] Ridley, M. (2006). *Genome: The Autobiography of a Species in 23 Chapters.* HarperCollins Publishers.

[39] Zerikly, M., & Challis, G. L. (2009). Strategies for the Discovery of New Natural Products by Genome Mining. *ChemBioChem , 10* (4), 625–633.

[40] H., G. (2009). Genomic mining--a concept for the discovery of new bioactive natural products. *Current Opinion in Drug Discovery & Development , 12* (2), 207-219.

[41] Crawford, J. M., Vagstad, A. L., Ehrlich, K. C., & Townsend, C. A. (2008). Starter unit specificity directs genomemining of polyketide synthase pathways in fungi. *Bioorganic Chemistry , 36* (1), 16-22.

[42] Song, L., Barona-Gomez, F., Corre, C., Xiang, L., Udwary, D. W., Austin, M. B., et al. (2006). Type III Polyketide Synthase β-Ketoacyl-ACP Starter Unit and Ethylmalonyl-CoA Extender Unit Selectivity Discovered by Streptomyces coelicolor Genome Mining. *Journal of American Chemical Society , 128* (46), 14754–14755.

[43] Yi, G.-S., & Kim, J. (2012). *Help*. Retrieved September 14, 2012, from PKMiner: Genome Mining Tool for Bacterial Aromatic Polyketide: http://pks.kaist.ac.kr/pkminer/index.cgi?menu=4

[44] Jayaseelan, K. V., Moreno, P., Truszkowski, A., Ertl, P., & Steinbeck, C. (2012). Natural product-likeness score revisited: an open-source, open-data implementation. *BMC Bioninformatics , 13* (106).

[45] JL, F., DP, V., & RS, P. (2003). The Signature Molecular Descriptor. 1. Using Extended Valence Sequences in QSAR and QSPR Studies. *J Chem Inf Model 2003 , 43* (3), 707.

[46] Shang, H., Lin, X., Zhang, Y., Yu, J. X., & Wang, W. (2010). Connected Substructure Similarity Search. *SIGMOD '10 Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (pp. 903-904). New York, NY, USA: ACM.

[47] *6. Fingerprints - Screening and Similarity*. (n.d.). Retrieved February 12, 2013, from Daylight: http://www.daylight.com/dayhtml/doc/theory/theory.finger.html

[48] J, G., LJ, B., AP, B., J, C., M, D., A, H., et al. (2011). ChEMBL: a large-scale bioactivity database for drug discovery. *Nucl Acids Res , 1*.

[49] JJ, I., & BK, S. (2005). ZINC - A free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling , 45* (1), 177.

[50] iChemLabs, LLC. (2008). Retrieved October 1, 2012, from ChemDoodle Web Components: http://web.chemdoodle.com/

[51] *What can PHP do?* (2012, September 29). Retrieved October 1, 2012, from php: http://www.php.net/manual/en/intro-whatcando.php

[52] (n.d.). Retrieved September 30, 2012, from ASP vs PHP: http://www.aspvsphp.com/

[53] *Classes and Objects*. (n.d.). Retrieved September 30, 2012, from PHP: http://www.php.net/manual/en/oop5.intro.php

[54] Börger, M. (2007). *Introduction to Object-Oriented Programming with PHP.* Quebec: PHP Conference.

[55] Marston, T. (2004, May 2). *The Model-View-Controller (MVC) Design Pattern for PHP*. Retrieved September 30, 2012, from Radicore: http://www.tonymarston.net/php-mysql/model-view-controller.html#principles

[56] Stump, J. (2005, September 15). *Understanding MVC in PHP*. Retrieved September 30, 2012, from O'Reilly: http://oreilly.com/php/archive/mvc-intro.html

[57] Atwood, J. (2008, May 5). *Understanding Model-View-Controller*. Retrieved September 30, 2012, from Coding Horror: http://www.codinghorror.com/blog/2008/05/understanding-model-view-controller.html

[58] (n.d.). Retrieved from Zend: http://www.zend.com/en/

[59] (n.d.). Retrieved from CakePHP: http://cakephp.org/

[60] (n.d.). Retrieved from YII Framework: http://www.yiiframework.com

[61] (n.d.). Retrieved from CodeIgniter: http://codeigniter.com/

[62] *PHP Frameworks*. (n.d.). Retrieved September 30, 2012, from http://www.phpframeworks.com/top-10-php-frameworks/

[63] Connelly, D. (2011, July 3). *The Best PHP Framework*. Retrieved September 30, 2012, from David Connelly: http://davidjconnelly.wordpress.com/2011/07/03/the-best-php-framework-of-2011/

[64] Wu, F. (2009, March 2). *Choosing a PHP Framework Round 2: Yii vs Kohana vs CodeIgniter*. Retrieved September 30, 2012, from Beyong Coding: http://www.beyondcoding.com/2009/03/02/choosing-a-php-framework-round-2-yii-vs-kohana-vs-codeigniter/

[65] *JChem Cartridge Architecture Overview.* (n.d.). Retrieved February 18, 2013, from Chemaxon: https://www.chemaxon.com/products/jchem-cartridge/

[66] *Calculation of molecular properties and bioactiviy scores*. (n.d.). Retrieved February 19, 2013, from Molinspiration: http://molinspiration.com/cgi-bin/properties

X. Appendix

A. Source Codes for Predyketide

BASE PATH = *\/Predyketide/*

…/resources/help.txt

```
The biosynthesis will start by selecting one of
the 16 starter units. The menu will be displayed
upon clicking the STARTER button.

Then the elongation process will continue by
selecting one of the four domain sequences:
   KS-AT-ACP(Produces ketone)
   KS-AT-KR-ACP(Produces alcohol)
   KS-AT-KR-DH-ACP(Produces alkene)
   KS-AT-KR-EH-DH-ACP(Produces methylene center)

The PREDICT button will enable after
three extender units.

To view the properties of a prediction,
   click the image ONCE.

To view a larger image of a prediction,
   click the image TWICE.
```

…/resources/natprod.txt

```
c1cc(cc(c1)Cl)NNC(=O)c2cc(cnc2)Br
Cc1ccnc(c1)NC(=O)Cc2cccc3c2cccc3
CC(=O)O[C@@H](CCN1CC2CC[NH+]1CC2)c3ccccc3
C[C@@H](C[NH+](C)C)C(C#N)(c1ccccc1)c2ccccc2
CCOc1ccc(cc1)N2C(=O)C[C@@H](C2=O)[NH+]3CCCCC3
CN(Cc1ccccc1)C(=O)CNS(=O)(=O)c2ccc(cc2)F
Cc1cc(cc(c1)NC(=O)C[C@H]2C(=O)N/C(=N/N=C(C)C)/S2)C
c1cc(cc(c1)N2C(=O)/C(=C\NNC(=O)N)/C(=NC2=S)[O-])C(F)(F)F
CCc1c(nc2c(cnn2c1[O-])C(=O)OCC)C
CC(C)(C)c1ccc(cc1)C(=O)Nc2ccccc2[S-]
Cc1nnc(s1)NC(=O)Nc2ccc(cc2)Cl
[H]/N=C/1\NC(=O)/C(=C/c2ccc(c(c2)OCc3ccccc3Cl)OC)/S1
Cn1ccnc1Sc2c(nc3ccccc3n2)Sc4nccn4C
c1ccc2c(c1)nc(c(n2)Cl)N3CCC[C@H](C3)O
c1ccc2c(c1)nc(c(n2)Cl)N3CCC[C@H](C3)CO
COc1ccc2c(c1)NC(=O)[C@H](S2)CC(=O)OC
Cc1cc(ccc1Br)NS(=O)(=O)c2ccc(s2)Br
c1ccc(cc1)C[NH+]2CCc3c(sc4c3c(=O)[nH]c(n4)Cc5ccc(cc5)Cl)C2
CC=C(c1ccccc1)c2ccccc2
Cn1c2c(c(=O)n(c1=O)C)n(c(n2)Oc3ccc(cc3)C(=O)OC)Cc4ccccc4
Cc1ccc(cc1)N=C=Nc2ccc(cc2)C
CCOc1cc(ccc1OCC(=O)N)/C=c/2\c(=O)n3c4ccc(c(c4nc3s2)C)C
Cc1c2cc(ccc2oc1C(=O)N3CCN(CC3)c4ccc(cc4)OC)Br
Cc1cccc(c1)NC(=O)COc2ccc(cc2[C@H]3C4=C(CCCC4=O)N=C5C3=C(CCC5)[O-])Cl
```

c1cc(c(cc1Br)[C@@H]2C3C(=NC4=C2C(=O)CCC4)CCCC3=O)OCC(=O)[O-]

Cn1c(nnc1SCC(=O)Nc2ccccc2Cl)CNC(=O)c3cccs3

Cn1c(nnc1SCC(=O)Nc2cccc(c2)Cl)CNC(=O)c3cccs3

Cc1cc(ccc1NC(=O)CSc2nnc(n2C)CNC(=O)c3cccs3)Cl

Cn1c(nnc1SCC(=O)Nc2ccc(cc2)Cl)CNC(=O)c3cccs3

COc1ccc(cc1)c2nc([nH]n2)SCC(=O)Nc3ccc(cc3F)Br

CC1(CC2=NC3=C([C@H](C2C(=O)C1)c4cc(ccc4OCC(=O)[O-])Cl)C(=O)CC(C3)(C)C)C

C[C@H]1CCCC[C@H]1NC(=O)CSc2nnc(n2C)CC(=O)Nc3ccc(c(c3)Cl)Cl

C[C@@H]1CCCC[C@H]1NC(=O)CSc2nnc(n2C)CC(=O)Nc3ccc(c(c3)Cl)Cl

C[C@H]1CCCC[C@@H]1NC(=O)CSc2nnc(n2C)CC(=O)Nc3ccc(c(c3)Cl)Cl

C[C@@H]1CCCC[C@@H]1NC(=O)CSc2nnc(n2C)CC(=O)Nc3ccc(c(c3)Cl)Cl

Cn1c(nnc1SCC(=O)NC2CCCCC2)CC(=O)Nc3ccc(c(c3)Cl)Cl

Cc1csc(n1)NC(=O)CSc2nnc(n2C)CC(=O)Nc3ccc(c(c3)Cl)Cl

Cn1c(nnc1SCC(=O)NCCc2ccccc2)c3cc4cc(ccc4o3)Br

C[C@H](c1ccccc1)NC(=O)CSc2nnc(n2C)c3cc4cc(ccc4o3)Br

CCOc1cc(cc(c1OC(=O)C)CC=C)/C=c\2/c(=O)n3c4ccccc4nc3s2

c1cc(ccc1C(=O)COC(=O)c2cc(=O)[nH]c3c2cc(cc3)Br)Br

c1ccc2c(c1)[nH]c(n2)c3ccc(cc3)NC(=S)NC(=O)c4cc(cnc4)Br

Cc1cccc(c1C)OCc2nnc(n2C)SCC(=O)Nc3c(cc(cc3Br)F)F

Cn1c2c(c(=O)n(c1=O)C)n(cn2)CCCOC3c4ccccc4C=Cc5c3cccc5

Cn1c2c(c(=O)n(c1=O)C)n(cn2)CCCOC3c4ccccc4CCc5c3cccc5

Cn1c(nnc1SCC(=O)Nc2ccc(cc2F)Br)CNC(=O)c3cccs3

Cc1ccc(cc1)SCc2c3c([nH]n2)OC(=C([C@]34c5cc(ccc5NC4=O)Cl)C#N)N

Cc1ccc(cc1)SCc2c3c([nH]n2)OC(=C([C@]34c5cc(ccc5NC4=O)F)C#N)N

Cc1ccc(cc1)SCc2c3c([nH]n2)OC(=C([C@@]34c5cc(ccc5NC4=O)F)C#N)N

C[C@H](c1nnc(n1C)SCC(=O)Nc2nc3ccccc3s2)NC(=O)c4cccs4

CCN1c2ccccc2[C@]3(C1=O)c4c(n[nH]c4OC(=C3C#N)N)CSc5ccc(cc5)C

C[C@@H](c1nnc(n1C)SCC(=O)Nc2nc3ccccc3s2)NC(=O)c4cccs4

CCN1c2ccccc2[C@@]3(C1=O)c4c(n[nH]c4OC(=C3C#N)N)CSc5ccc(cc5)C

C[C@H](c1nnc(n1C)SCC(=O)Nc2c(c3c(s2)CCCC3)C#N)NC(=O)c4cccs4

COc1ccc(cc1)c2ccc3c4c(c5c(c(=O)[nH]4)[C@@H](C(=C(O5)N)C#N)c6ccc(cc6)F)sc3n2

C[C@H]1CC=C2[C@H](C1)[C@@]3(c4ccccc4N(C3=O)CCc5ccccc5)C(C(=C2C#N)N)(C#N)C#N

C[C@@H](c1nnc(n1C)SCC(=O)Nc2c(c3c(s2)CCCC3)C#N)NC(=O)c4cccs4

C[C@@H]1CC=C2[C@@H](C1)[C@@]3(c4ccccc4N(C3=O)CCc5ccccc5)C(C(=C2C#N)N)(C#N)C#N

Cc1ccc2c(c1)nc([nH]2)CSc3c(c(cc(n3)c4ccco4)C(F)(F)F)C#N

COc1ccc(cc1OC)c2cc(c(c(n2)SCc3[nH]c4ccccc4n3)C#N)C(F)(F)F

CCCOc1ccc(cc1OC)[C@H]2c3c([nH]nc3OC(=C2C#N)N)c4ccc(c(c4)C)C

CCCOc1ccc(cc1OC)[C@@H]2c3c([nH]nc3OC(=C2C#N)N)c4ccc(c(c4)C)C

Cc1ccc(cc1C)c2c3c(n[nH]2)OC(=C([C@H]3c4ccc(cc4)Br)C#N)N

Cc1ccc(cc1C)c2c3c(n[nH]2)OC(=C([C@@H]3c4ccc(cc4)Br)C#N)N

Cc1ccc(cc1C)c2c3c(n[nH]2)OC(=C([C@H]3c4ccc(c(c4)Br)F)C#N)N

Cc1ccc(cc1)c2c3c(n[nH]2)OC(=C([C@@]34c5ccccc5N(C4=O)Cc6ccc(cc6)Cl)C#N)N

c1cc(ccc1c2c3c(n[nH]2)OC(=C([C@H]3c4ccc(c(c4)Br)F)C#N)N)F

c1cc(ccc1c2c3c(n[nH]2)OC(=C([C@@H]3c4ccc(c(c4)Br)F)C#N)N)F

Cn1c(nnc1SC/C(=N/c2ccc(cc2F)Br)/O)CNC(=O)c3ccco3

CCCCOc1cc(ccc1OC)[C@H]2c3c([nH]nc3OC(=C2C#N)N)c4ccc(cc4)F

COc1ccc(cc1Br)[C@@H]2c3c([nH]nc3OC(=C2C#N)N)c4cccs4

```
COc1ccc(cc1[C@H]2c3c([nH]nc3OC(=C2C#N)N)c4ccc5c(c4)OCO5)Br

COc1ccc(cc1[C@@H]2c3c([nH]nc3OC(=C2C#N)N)c4ccc5c(c4)OCO5)Br

[H]/N=C/1\[C@@H]([C@H]([C@H]2C[NH+](CC=C2C1(C#N)C#N)CC)c3cc(ccc3F)Br)C#N

CCc1c2c([nH]n1)OC(=C([C@]23c4ccccc4N(C3=O)Cc5ccc(cc5)Cl)C#N)N

CCc1c2c([nH]n1)OC(=C([C@@]23c4ccccc4N(C3=O)Cc5ccc(cc5)Cl)C#N)N

CC(C)N1CC=C2[C@H](C1)[C@H](C(C(=C2C#N)N)(C#N)C#N)c3cc(cs3)Br

Cc1cccc(c1OCc2nnc(n2C)SCC(=O)N3CCN(CC3)c4ccccc4)C

Cc1ccc(cc1)SCc2c3c(n[nH]2)OC(=C([C@H]3c4ccc(cc4)Br)C#N)N

Cc1ccc(cc1)SCc2c3c(n[nH]2)OC(=C([C@@H]3c4ccc(cc4)Br)C#N)N

Cc1ccc(cc1)SCc2c3c(n[nH]2)OC(=C([C@H]3c4cccc(c4)Br)C#N)N

c1cc(cc(c1)Cl)NNC(=O)c2cc(cnc2)Br

Cc1ccnc(c1)NC(=O)Cc2cccc3c2cccc3

Cc1cc(=O)[nH]c(n1)SCCOc2ccccc2OC

CCc1c(nc2c(cnn2c1[O-])C(=O)OCC)C

CC(C)(C)c1ccc(cc1)C(=O)Nc2ccccc2[S-]

Cc1nnc(s1)NC(=O)Nc2ccc(cc2)Cl

Cn1ccnc1Sc2c(nc3ccccc3n2)Sc4nccn4C

COC(=O)c1ccc(cc1)C(=O)OC

Cc1ccc(cc1)N=C=Nc2ccc(cc2)C

CC=C(c1ccccc1)c2ccccc2

Cc1cc(ccc1Br)NS(=O)(=O)c2ccc(s2)Br

Cn1c(=O)cc(n(c1=O)C)SCc2ccccc2

Cn1ccnc1Sc2c(nc3ccccc3n2)Sc4nccn4C

CCCCOc1cccc(c1)C(=O)Nc2cc(ccc2F)[N+](=O)[O-]

Cc1cc(ccc1OC)CCCC(=O)NC2CCCCC2

CC[n+]1c2ccccc2n(c1N)CC(=O)c3ccco3

c1ccc(cc1)c2nnc(n2N)SCC(=O)[O-]

CCc1ccccc1NC(=O)CSc2nnc(o2)c3ccccc3Br

c1ccc(c(c1)c2nnc(o2)SCC(=O)Nc3ccccc3C(F)(F)F)Br
```

…/resources/synthetic.txt

```
O1C(=O)CC(O)CC(O)CCCC(O)CC(O)CC(=O)CC(O)C(C)C(O)CC(O)C=CC=CC=CC=CC=CC=CC(C)C(O)C(C)C1(C)

O1CCCC(C)CC(C)C(O)C(C)=CC=CCC1(C1CCCC(C(=O)(O))1)

O1C(=O)CC(O)C(C)C(=O)C(C)C(O)C(C)CCCCC(O)CC1(C(C)=CC)

O1C(=O)C(C)C(O)C(C)C(O)C(C)CC(C)C(=O)C(C)C(O)C(C)C1(C(C))

O=C1Oc2ccccc2C=C1

CSc1ccc(CC(C)N)cc1

CN(C)c1cccc2c(cccc12)S(=O)(=O)Nc3ncccn3

COc1ccc2c(c1)[nH]c3cnccc23

COc1ccc2ncc(nc2c1)N3CCNCC3

OC1C(O)C(OC1CBr)N2C=C(F)C(=O)NC2=O

Oc1ncc(F)c(O)n1

CC(C)(C)\N=C(\NC#N)/NC1[C@H](O)C(C)(C)Oc2ccc(cc12)C#N

CN(C)c1cccc2c(cccc12)S(=O)(=O)Nc3nnc(Cl)cc3C

Cc1nc(O)c2cc(CN(CC#C)c3ccc(cc3)C(=O)NCc4cocc4)ccc2n1

COc1cccc2[nH]c3C=NCCc3c12

NC1=NC(=O)N(C=C1)C2O[C@H](CO)[C@H](O)[C@H]2O

NS(=O)(=O)c1cccc2ccccc12

CNCC1NCC(c2ccccc2)c3ccccc13

CN1C(=O)CN=C(c2ccccc2)c3cc(Cl)ccc13

CC(=O)Nc1cccnc1N2CCN(CC2)C(=O)c3cc4ccccc4[nH]3

NC1=Nc2ccccc2CC1

[Cl-].CC(C)Oc1noc2CC[NH2+]Cc12

Cc1cc(O)nc2cc(O)ccc12

CNc1ncnc2c1ncn2Cc3cccc(F)c3

Nc1ccccc1S(=O)(=O)N
```

[Cl-].C[C@@H]([NH3+])Cc1c2occc2c(Br)c3occc13

CCCNC(=O)CC1CCN(CC1)c2nc(N)c3cc(OC)c(OC)cc3n2

CN(C)C1(C)C2CCC(C2)C1(C)C

[Cl-].C#CCOc1noc2CC[NH2+]Cc12

[Cl-].CNCCN(C)C(=O)Oc1ccc(OC)cc1

CN1Cc2c(ncn2c3ccc(Cl)cc3C1=O)C(=O)OCCC(C)(C)C

CNc1ncnc2c1ncn2Cc3cccc(O)c3

CCOC(=O)C1=C(O)C(=O)N(Cc2ccccn2)C1

CC(C)(C)NCC(O)COC(=O)c1ccc(OCC=C)cc1

Cc1ccc2[nH]c(cc2c1)C(=O)c3cc4ccccc4[nH]3

Nc1ccc2nc(N)nc(N)c2c1

CCCC[C@@H]1[C@@H](CC(=O)N)C1=C

CC(C)NCC(O)COc1ccc(COCC2CC2)cc1

COc1cc(Cc2cnc(N)nc2N)cc(OC)c1OC

Nc1nc(N)c2cc(NC(=O)Cc3ccc(Cl)c(Cl)c3)ccc2n1

CC(C)(C)NCC(O)COC(=O)C(c1ccccc1)c2ccccc2

CC(C)(C)NCC(O)COC(=O)c1ccc(cc1)[N+](=O)[O-]

C[C@](O)(CO)c1oc2ccc3[C@@H]4COc5cc(O)ccc5[C@@H]4Oc3c2c1

C(=O)CC(=O)CC(=O)CC(O)CC(O)CC(O)CC(O)CC(O)CC(O)CC(O)CC(O)CC(O)CC(O)C(CC)C(O)C(CC)C(=O)C

Cc1nc(O)c2cc(CN(CC#C)c3ccc(cc3)C(=O)NCc4ccccn4)ccc2n1

COc1cc2nc(nc(N)c2cc1OC)N3CCN(CC3)C(=O)\C=C\c4ccc(F)cc4

CN(C)c1cccc2c(cccc12)S(=O)(=O)Nc3cccnc3

O=C1NCCN(N1)c2cccnc2

Clc1ccc(cc1)c2cc([nH]n2)C3CCN(CCc4ccccc4)CC3

CC(N)Cc1ccccc1

CCOC(=O)c1ncn2c1CN(C)C(=O)c3cc(Cl)ccc23

CN(C)CC1N(C)CC(c2ccccc2)c3ccccc13

Cc1ccc2nc(cnc2c1)N3CCNCC3

CCC1CCN(C)c2cc3OC(=O)C=C(c3cc12)C(F)(F)F

OC1C(O)C(OC1Cl)N2C=C(F)C(=O)NC2=O

Cc1cc(CCCCCOc2c(Cl)cc(cc2Cl)C3=NCCO3)on1

Cc1cc(CCCCCOc2ccc(cc2)c3oc(C)c(C)n3)on1

CCC1Nc2cc3OC(=O)C=C(c3cc2CC1C)C(F)(F)F

NC(C(c1ccccc1)c2ccccc2)C(=O)N3CCCC3C(=O)NCc4cccc(O)c4

Nc1ccc2nc(N)nc(N)c2c1Cl

[Cl-].Cc1ccc2Oc3ccccc3C(SCCN4CCCCC4)c2c1

Cc1nc(O)c2cc(CN(CC#C)c3ccc(cc3)C(=O)NCc4ccccc4)ccc2n1

Cc1ccc(Sc2cncc3sc(cc23)C(=O)N)c(C)c1

COc1cccc(OC)c1CNC(=O)C2CCCN2C(=O)C(N)C(c3ccccc3)c4ccccc4

Fc1ccccc1Cn2cnc3c(ncnc23)N4CCC4

CC(=O)NCC1CN(C(=O)O1)c2ccc(cc2)C3=CC(=O)C(=CC=C3)NCC=C

CC(C)C(Sc1ccc2ccccc2c1)C(=O)Nc3ccc4nc(N)nc(N)c4c3C

CC(C)(C)OC(=O)C1=C(O)C(=O)N(Cc2cccs2)C1

[Cl-].CN(C)CCSC1c2ccccc2Oc3ncccc13

CC1(COc2ccc(CC3SC(=O)NC3=O)cc2)CCCCC1

Fc1cccc2nc(cnc12)N3CCNCC3

CCN(C)c1ncnc2c1ncn2Cc3ccccc3F

C[C@@H]1Cc2cc3c(cc(O)nc3cc2N[C@@H]1C)C(F)(F)F

CC(C)Nc1ncnc2c1ncn2Cc3ccccc3F

CCC(CC)C(=O)OCC1CN(Cc2cc(OC)c(OC)c(OC)c2)CCN1Cc3cc(OC)c(OC)c(OC)c3

CCNCC1CCN(C1)c2c(F)cc3C(=O)C(=CN(NC)c3c2F)C(=O)O

Nc1ncnc2c1ncn2C3OC(Cl)C(O)C3O

Fc1ccc(cc1)c2c(nc3CCCn23)c4ccccn4

COc1cc2nc(nc(N)c2cc1OC)N3CCN(CC3)C(=O)c4occc4

COC(=O)\C=C\C=C\C(=O)N1CCN(CC1)c2nc(N)c3cc(OC)c(OC)cc3n2

Cc1cc(ccc1C(=O)c2ccccc2)N3N=CC(=O)NC3=O

Cc1ccc(Sc2cncc3sc(cc23)C(=O)Nc4ccc(Cl)cc4)cc1

Nc1nc(O)c2CCCCc2n1

```
CCOC(=O)c1ncn2c1CN(C)C(=O)c3cc(F)ccc23
CNC(=O)c1cc(nnc1Cl)c2ccncc2
COc1cc2nc(nc(N)c2cc1OC)N3CCC(CC3)C(=O)NCc4ccccc4
Clc1nnc(cc1C#N)c2ccncc2
COc1ccc(cc1)c2cc(C(=O)N)c(Cl)nn2
CCOC(=O)c1ncn2c1CN(C)C(=O)c3cc(ccc23)N=[N+]=[N-]
COc1cc2nc(nc(N)c2cc1OC)N3CCC(CC3)C(=O)NC4CCCC4
C\C(=C/CN1OC(=O)NC1=O)\c2ccc(OCc3nc(oc3C)c4ccc(cc4)C(F)(F)F)cc2
CN1Cc2c(ncn2c3ccc(Br)cc3C1=O)C(=O)OC(C)(C)C
Nc1nc(N)c2c(Cl)c(Cl)ccc2n1
CN(C)c1ncnc2c1ncn2Cc3ccccc3
CCc1nnc2c(NC3CC3)nc4ccccc4n12
Nc1ccc(cc1)S(=O)(=O)Nc2ccccc2S(=O)(=O)N
Cc1ccc(cc1)N2C(=NC(=NC2(C)C)N)N
C1CN[C@@H](C1)c2cccnc2
Cn1c(CC(=O)O)cc(CO)c1C(=O)c2ccc(Cl)cc2
NC(=O)c1cc2c(Cl)cncc2s1
Cc1ccc(Sc2cncc3sc(\C=N\O)cc23)cc1
COc1ccc2c3CCNCc3[nH]c2c1
Oc1ccc2c(OC(=O)c3ccccc23)c1O
Oc1cc2C(=O)Oc3ccccc3c2cc1O
```

…/src/starterUnits.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<starter-units>


    <aStarterUnit>
        <name>Acetoacetyl-CoA</name>
        <SMILES-forward>C(=O)C</SMILES-forward>
        <SMILES-backward>CC(=O)C</SMILES-backward>
        <image>Acetoacetyl-CoA.jpg</image>
        <description></description>
    </aStarterUnit>


    <aStarterUnit>
        <name>2-Methylbutyryl-CoA</name>
        <SMILES-forward>(C)CC</SMILES-forward>
        <SMILES-backward>CCC(C)</SMILES-backward>
        <image>2-Methylbutyryl-CoA.jpg</image>
        <description></description>
    </aStarterUnit>


    <aStarterUnit>
        <name>3,4-DHCHC-CoA</name>
        <SMILES-forward>1CC(O)C(O)CC1</SMILES-forward>
        <SMILES-backward>C1CC(O)C(O)CC1</SMILES-backward>
        <image>3_4_DHCHC_CoA.jpg</image>
        <description></description>
    </aStarterUnit>


    <aStarterUnit>
        <name>3,5-AHBA</name>
        <SMILES-forward>1C=C(O)C=C(N)C=1</SMILES-forward>
        <SMILES-backward>C1=C(O)C=C(N)C=C1</SMILES-backward>
        <image>3_5_AHBA_CoA.jpg</image>
        <description></description>
```

```
	</aStarterUnit>

	<aStarterUnit>
		<name>3-Methylbutyryl-CoA</name>
		<SMILES-forward>CC(C)C</SMILES-forward>
		<SMILES-backward>CC(C)C</SMILES-backward>
		<image>3-Methylbutyryl-CoA.jpg</image>
		<description></description>
	</aStarterUnit>

	<aStarterUnit>
		<name>Acetyl-CoA</name>
		<SMILES-forward>XXX</SMILES-forward>
		<SMILES-backward>C</SMILES-backward>
		<image>Acetyl-CoA.jpg</image>
		<description></description>
	</aStarterUnit>

	<aStarterUnit>
		<name>Benzonyl-CoA</name>
		<SMILES-forward>1=CC=CC=C1</SMILES-forward>
		<SMILES-backward>C1C=CC=CC1</SMILES-backward>
		<image>Benzoyl-CoA.jpg</image>
		<description></description>
	</aStarterUnit>

	<aStarterUnit>
		<name>Butyryl-CoA</name>
		<SMILES-forward>CC</SMILES-forward>
		<SMILES-backward>CCC</SMILES-backward>
		<image>Butyryl-CoA.jpg</image>
		<description></description>
	</aStarterUnit>

	<aStarterUnit>
		<name>Cyclohexanecarboxylic acid</name>
		<SMILES-forward>1CCCCC1</SMILES-forward>
		<SMILES-backward>C1CCCCC1</SMILES-backward>
		<image>CHC.jpg</image>
		<description></description>
	</aStarterUnit>

	<aStarterUnit>
		<name>Glycine</name>
		<SMILES-forward>N</SMILES-forward>
		<SMILES-backward>NC</SMILES-backward>
		<image>Glycine.jpg</image>
		<description></description>
	</aStarterUnit>

	<aStarterUnit>
		<name>Glycolate</name>
		<SMILES-forward>O</SMILES-forward>
		<SMILES-backward>OC</SMILES-backward>
		<image>Glycolate.jpg</image>
		<description></description>
```

```
    </aStarterUnit>

    <aStarterUnit>
        <name>p-Aminobenzoate</name>
        <SMILES-forward>1C=CC(N)=CC=1</SMILES-forward>
        <SMILES-backward>C1=CC(N)=CC=C1</SMILES-backward>
        <image>p-Aminobenzoate.jpg</image>
        <description></description>
    </aStarterUnit>

    <aStarterUnit>
        <name>Phenylacetyl-CoA</name>
        <SMILES-forward>1C=CC=CC=1</SMILES-forward>
        <SMILES-backward>C1=CC=CC=C1</SMILES-backward>
        <image>Phenylacetyl-CoA.jpg</image>
        <description></description>
    </aStarterUnit>

    <aStarterUnit>
        <name>p-Nitrobenzoate</name>
        <SMILES-forward>1C=CC(NO)=CC=1</SMILES-forward>
        <SMILES-backward>C1=CC(NO)=CC=C1</SMILES-backward>
        <image>p-Nitrobenzoate.jpg</image>
        <description></description>
    </aStarterUnit>

    <aStarterUnit>
        <name>Propionyl-CoA</name>
        <SMILES-forward>C</SMILES-forward>
        <SMILES-backward>CC</SMILES-backward>
        <image>Propionyl-CoA.jpg</image>
        <description></description>
    </aStarterUnit>

    <aStarterUnit>
        <name>trans-1,2-CPDA</name>
        <SMILES-forward>1C(C(=O)(O))CCC1</SMILES-forward>
        <SMILES-backward>C1CCC(C(=O)(O))C1</SMILES-backward>
        <image>trans-1_2-CPDA.jpg</image>
        <description></description>
    </aStarterUnit>
</starter-units>
```

<u>…/src/extenderUnits.xml</u>

```
<?xml version="1.0" encoding="UTF-8"?>
<extender-units>

    <anExtenderUnit>
        <name>Malonyl-CoA</name>
        <SMILES-forward>XXX</SMILES-forward>
        <SMILES-backward>C</SMILES-backward>
        <image>Malonyl-CoA.jpg</image>
        <description></description>
    </anExtenderUnit>

    <anExtenderUnit>
        <name>Ethylmalonyl-CoA</name>
```

```
                <SMILES-forward>(CC)</SMILES-forward>

                <SMILES-backward>C(CC)</SMILES-backward>

                <image>Ethylmalonyl-CoA.jpg</image>

                <description></description>

        </anExtenderUnit>



        <anExtenderUnit>

                <name>Hydroxymalonyl-CoA</name>

                <SMILES-forward>(O)</SMILES-forward>

                <SMILES-backward>C(O)</SMILES-backward>

                <image>Hydroxymalonyl-CoA.jpg</image>

                <description></description>

        </anExtenderUnit>



        <anExtenderUnit>

                <name>Methylmalonyl-CoA</name>

                <SMILES-forward>(C)</SMILES-forward>

                <SMILES-backward>C(C)</SMILES-backward>

                <image>Methylmalonyl-CoA.jpg</image>

                <description></description>

        </anExtenderUnit>

</extender-units>
```

…/src/sp/AboutDialog.java

```
package sp;


import javax.swing.*;

/**

 *

 * @author user

 */

@SuppressWarnings("serial")

public class AboutDialog extends JDialog

{



  /** Creates new form AboutDialog */

  public AboutDialog()

  {

    initComponents();

  }



  /** This method is called from within the constructor to

   * initialize the form.

   * WARNING: Do NOT modify this code. The content of this method is

   * always regenerated by the Form Editor.

   */

  // <editor-fold defaultstate="collapsed" desc="Generated Code">

  private void initComponents()

  {



    aboutDialogMainPanel = new JPanel();

    jLabel1 = new JLabel();

    jLabel2 = new JLabel();

    jLabel3 = new JLabel();

    jLabel4 = new JLabel();

    jLabel5 = new JLabel();

    jLabel6 = new JLabel();
```

```
    jLabel1.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N

jLabel1.setHorizontalAlignment(SwingConstants.CENTER);

    jLabel1.setText("PREDYKETIDE");



jLabel2.setHorizontalAlignment(SwingConstants.CENTER);

    jLabel2.setText("A Special Problem for BS Computer Science Degree Program");



jLabel3.setHorizontalAlignment(SwingConstants.CENTER);

    jLabel3.setText("John Althom A. Mendoza");



jLabel4.setHorizontalAlignment(SwingConstants.CENTER);

    jLabel4.setText("Copyright Information 2013");



jLabel5.setHorizontalAlignment(SwingConstants.CENTER);

    jLabel5.setText("All Rights Reserved.");



jLabel6.setHorizontalAlignment(SwingConstants.CENTER);

    jLabel6.setText("University of the Philippines - Manila");


    GroupLayout aboutDialogMainPanelLayout = new GroupLayout(aboutDialogMainPanel);

aboutDialogMainPanel.setLayout(aboutDialogMainPanelLayout);

    aboutDialogMainPanelLayout.setHorizontalGroup(

aboutDialogMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)

.addGroup(aboutDialogMainPanelLayout.createSequentialGroup()

            .addContainerGap()

.addGroup(aboutDialogMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)

                .addComponent(jLabel2, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(jLabel1, GroupLayout.DEFAULT_SIZE, 286, Short.MAX_VALUE)

                .addComponent(jLabel3, GroupLayout.DEFAULT_SIZE, 286, Short.MAX_VALUE))

            .addContainerGap())

.addGroup(aboutDialogMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)

.addGroup(aboutDialogMainPanelLayout.createSequentialGroup()

                .addContainerGap()

                .addComponent(jLabel4, GroupLayout.DEFAULT_SIZE, 286, Short.MAX_VALUE)

                .addContainerGap()))

.addGroup(aboutDialogMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)

.addGroup(aboutDialogMainPanelLayout.createSequentialGroup()

                .addContainerGap()

                .addComponent(jLabel5, GroupLayout.DEFAULT_SIZE, 286, Short.MAX_VALUE)

                .addContainerGap()))

.addGroup(aboutDialogMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)

.addGroup(aboutDialogMainPanelLayout.createSequentialGroup()
```

```
                        .addContainerGap()
                        .addComponent(jLabel6, GroupLayout.DEFAULT_SIZE, 286, Short.MAX_VALUE)
                        .addContainerGap()))
        );
        aboutDialogMainPanelLayout.setVerticalGroup(
            aboutDialogMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
                .addGroup(aboutDialogMainPanelLayout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel1, GroupLayout.PREFERRED_SIZE, 36, GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel2, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGap(18, 18, 18)
                    .addComponent(jLabel3)
                    .addGap(35, 35, 35))
                .addGroup(aboutDialogMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
                    .addGroup(GroupLayout.Alignment.TRAILING, aboutDialogMainPanelLayout.createSequentialGroup()
                        .addContainerGap(96, Short.MAX_VALUE)
                        .addComponent(jLabel4)
                        .addGap(24, 24, 24)))
                .addGroup(aboutDialogMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
                    .addGroup(GroupLayout.Alignment.TRAILING, aboutDialogMainPanelLayout.createSequentialGroup()
                        .addContainerGap(106, Short.MAX_VALUE)
                        .addComponent(jLabel5)
                        .addGap(14, 14, 14)))
                .addGroup(aboutDialogMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
                    .addGroup(GroupLayout.Alignment.TRAILING, aboutDialogMainPanelLayout.createSequentialGroup()
                        .addGap(63, 63, 63)
                        .addComponent(jLabel6, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addGap(57, 57, 57)))
        );

        GroupLayout layout = new GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                .addComponent(aboutDialogMainPanel, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                .addComponent(aboutDialogMainPanel, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        setModal(true);
        setResizable(false);
        pack();
        setLocationRelativeTo(null);
```

```
        setVisible(true);
    }// </editor-fold>



    // Variables declaration - do not modify
    private JPanel aboutDialogMainPanel;
    private JLabel jLabel1;
    private JLabel jLabel2;
    private JLabel jLabel3;
    private JLabel jLabel4;
    private JLabel jLabel5;
    private JLabel jLabel6;
    // End of variables declaration
}
```

…/src/sp/ExtenderUnitCollection.java

```
/**
 * @author John Althom A. Mendoza
 *
 * Class to retrieve the the known extender units from the xml file included in the
 *                         application.
 *
 * */

package sp;

import java.io.InputStream;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import utils.XML_Utils;



public class ExtenderUnitCollection
{



    public ExtenderUnitCollection()
    {

        try
        {
            InputStream starterXML = ExtenderUnitCollection.class.getResourceAsStream("/extenderUnits.xml");
            Document doc = XML_Utils.readXml(starterXML);
            doc.getDocumentElement().normalize();

            NodeList nList = doc.getElementsByTagName("anExtenderUnit");

            for (int temp = 0; temp < nList.getLength(); temp++)
            {

                Node nNode = nList.item(temp);
                if (nNode.getNodeType() == Node.ELEMENT_NODE)
                {

                    Element eElement = (Element) nNode;
```

```
                Predyketide.extenderUnits.add(new
PolyketideUnit(getTagValue("name", eElement),

                                        getTagValue("SMILES-forward",
eElement),

                                        getTagValue("image",
eElement)));



             }

          }

       }

       catch (Exception e)

       {

          e.printStackTrace();

       }

    }


    private static String getTagValue(String sTag, Element
eElement)

    {

       NodeList nlList =
eElement.getElementsByTagName(sTag).item(0).getChildNo
des();

       Node nValue = (Node) nlList.item(0);

       return nValue.getNodeValue();

    }

}
```

… /src/sp/HelpDialog.java

```
package sp;


import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;


import javax.swing.*;


/**

 *

 * @author user

 */

@SuppressWarnings("serial")

public class HelpDialog extends JDialog

{


   private String helpText;

   /** Creates new form HelpDialog */

   public HelpDialog(String helpText)

   {

      this.helpText = helpText;

      initComponents();

   }


   public HelpDialog()

   {


      try

      {

         helpText = "";

         InputStream in =
getClass().getResourceAsStream("/resources/help.txt");
```

```
      if(in == null)
      {
        in = getClass().getResourceAsStream("/help.txt");
      }

      BufferedReader br = new BufferedReader(new
InputStreamReader(in));
      String text;
      while ((text = br.readLine()) != null)
      {
        helpText += text + "\n";
      }
    }
    catch (IOException e)
    {
      e.printStackTrace();
    }
    initComponents();
  }

  /** This method is called from within the constructor to
   * initialize the form.
   * WARNING: Do NOT modify this code. The content of this
method is
   * always regenerated by the Form Editor.
   */
  // <editor-fold defaultstate="collapsed" desc="Generated
Code">
  private void initComponents()
  {

    helpTitlePanel = new JPanel();
    helpLabel = new JLabel();
    helpScrollPane = new JScrollPane();
    helpTextArea = new JTextArea();
    helpSeparator = new JSeparator();

    helpLabel.setFont(new java.awt.Font("Tahoma", 0, 14));
// NOI18N
    helpLabel.setText("Help");

    GroupLayout helpTitlePanelLayout = new
GroupLayout(helpTitlePanel);
    helpTitlePanel.setLayout(helpTitlePanelLayout);
    helpTitlePanelLayout.setHorizontalGroup(

helpTitlePanelLayout.createParallelGroup(GroupLayout.Align
ment.LEADING)

.addGroup(helpTitlePanelLayout.createSequentialGroup()
          .addContainerGap()
          .addComponent(helpLabel)
          .addContainerGap(364, Short.MAX_VALUE))
    );
    helpTitlePanelLayout.setVerticalGroup(

helpTitlePanelLayout.createParallelGroup(GroupLayout.Align
ment.LEADING)

.addGroup(helpTitlePanelLayout.createSequentialGroup()
          .addContainerGap()
          .addComponent(helpLabel)
          .addContainerGap(GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
```

```
    helpTextArea.setColumns(20);

    helpTextArea.setRows(5);

    helpScrollPane.setViewportView(helpTextArea);



    GroupLayout layout = new
GroupLayout(getContentPane());

    getContentPane().setLayout(layout);

    layout.setHorizontalGroup(


layout.createParallelGroup(GroupLayout.Alignment.LEADING
)

        .addComponent(helpTitlePanel,
GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addGroup(GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

            .addContainerGap()


.addGroup(layout.createParallelGroup(GroupLayout.Alignme
nt.TRAILING)

                .addComponent(helpSeparator,
GroupLayout.Alignment.LEADING,
GroupLayout.DEFAULT_SIZE, 380, Short.MAX_VALUE)

                .addComponent(helpScrollPane,
GroupLayout.Alignment.LEADING,
GroupLayout.DEFAULT_SIZE, 380, Short.MAX_VALUE))

            .addContainerGap())

    );

    layout.setVerticalGroup(


layout.createParallelGroup(GroupLayout.Alignment.LEADING
)

        .addGroup(layout.createSequentialGroup()

            .addComponent(helpTitlePanel,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)

            .addGap(1, 1, 1)

            .addComponent(helpSeparator,
GroupLayout.PREFERRED_SIZE, 10,
GroupLayout.PREFERRED_SIZE)


.addPreferredGap(LayoutStyle.ComponentPlacement.RELATE
D, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addComponent(helpScrollPane,
GroupLayout.PREFERRED_SIZE, 254,
GroupLayout.PREFERRED_SIZE))

    );



    helpTextArea.setText(helpText);

    helpTextArea.setEditable(false);

    setModal(true);

    setResizable(false);

    pack();

    setLocationRelativeTo(null);

    setVisible(true);

  }// </editor-fold>



  // Variables declaration - do not modify

  private JLabel helpLabel;

  private JScrollPane helpScrollPane;

  private JSeparator helpSeparator;

  private JTextArea helpTextArea;

  private JPanel helpTitlePanel;

  // End of variables declaration

}
```

…/src/sp/Main.java

```
package sp;



public class Main

{
```

```
    /**
     * @param args the command line arguments
     */
    public static void main(String args[])
    {

        try {
                                        new Predyketide().setVisible(true);
                        } catch (InterruptedException e) {
                                        e.printStackTrace();
                        }
    }
}
```

…/src/sp/PolyketideChainColumnPanel.java

```
package sp;

import java.awt.Color;
import java.awt.GridLayout;

import javax.swing.JLabel;
import javax.swing.JPanel;

@SuppressWarnings("serial")
public class PolyketideChainColumnPanel extends JPanel
{



    /**
     * TODO isStarter kasi dapat DELETE lang ang option dun
     * TODO action sa bawat selection
     * TODO Redo yung printing (looped version na lang. baguhin ang mga variables.
     *
     * */
    public PolyketideChainColumnPanel(PolyketideChainPanel newPKPanel, String module, String unit, int captionHeight)
    {
        setBackground(Color.WHITE);
        JPanel captionPanel = new JPanel(new GridLayout(2, 1));

        captionPanel.setSize(WIDTH, captionHeight);
        captionPanel.setBackground(Color.WHITE);
        JLabel moduleLabel = new JLabel(module);
        captionPanel.add(moduleLabel);
        JLabel unitLabel = new JLabel(unit);
        captionPanel.add(unitLabel);

        add(captionPanel);
        add(newPKPanel);

        setVisible(true);
    }

}
```

…/src/sp/PolyketideChainPanel.java

```
package sp;

import java.awt.Color;
import java.awt.Dimension;
```

```
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.util.ArrayList;

import javax.swing.JPanel;
import javax.vecmath.Point2d;

import org.openscience.cdk.exception.InvalidSmilesException;
import org.openscience.cdk.interfaces.IAtom;
import org.openscience.cdk.interfaces.IAtomContainer;
import org.openscience.cdk.layout.StructureDiagramGenerator;
import org.openscience.cdk.renderer.AtomContainerRenderer;
import org.openscience.cdk.renderer.font.AWTFontManager;
import org.openscience.cdk.renderer.generators.BasicAtomGenerator;
import org.openscience.cdk.renderer.generators.BasicBondGenerator;
import org.openscience.cdk.renderer.generators.BasicSceneGenerator;
import org.openscience.cdk.renderer.generators.ExtendedAtomGenerator;
import org.openscience.cdk.renderer.generators.IGenerator;
import org.openscience.cdk.renderer.visitor.AWTDrawVisitor;


@SuppressWarnings("serial")
public class PolyketideChainPanel extends JPanel
{

  private int height;
  private int width;
  private IAtomContainer molecule;
  private AtomContainerRenderer renderer;
  private boolean willFlip = false;

  public PolyketideChainPanel(String moleculeSMILES, int width, int height) throws InvalidSmilesException
  {
    this.setSize(width, height);


    IAtomContainer dummyMolecule = (IAtomContainer) Predyketide.smpars.parseSmiles(moleculeSMILES);
    StructureDiagramGenerator sdg = new StructureDiagramGenerator();
    sdg.setMolecule(dummyMolecule);
    try {
      sdg.generateCoordinates();
    }
    catch (Exception ex)
    {
//Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
    }

    this.molecule = sdg.getMolecule();

    //determine whether to rotate the image
    //since the first atom will always be S, check if it has 2d Points (0,0)
```

```
        IAtom atom = molecule.getAtom(0);
        //System.out.println(atom.getSymbol());
        if(atom.getPoint2d().equals(new Point2d(0, 0)) )
        {
            willFlip = true;
        }
        else{
            willFlip = false;
        }


        this.width = width;
        this.height = height;

        // generators make the image elements
        ArrayList<IGenerator<IAtomContainer>> generators = new ArrayList<IGenerator<IAtomContainer>>();
        generators.add(new BasicSceneGenerator());
        generators.add(new BasicBondGenerator());
        generators.add(new BasicAtomGenerator());
        generators.add(new ExtendedAtomGenerator());

        // the renderer needs to have a toolkit-specific font manager
        this.renderer = new AtomContainerRenderer(generators, new AWTFontManager());


    }


    public Dimension getPreferredSize()
    {
        return new Dimension(width, height);
    }


    public void paint(Graphics graphics)
    {
        // the call to 'setup' only needs to be done on the first paint
        //renderer.setup(this.molecule, new Rectangle(width, height));
        //renderer.setZoom(0.6);
        //TODO words sa ibaba.
        // paint the background
        graphics.setColor(Color.WHITE);
        graphics.fillRect(0, 0, width, height);

        // the paint method also needs a toolkit-specific renderer
        Graphics2D g2d = (Graphics2D) graphics; // Create a Java2D version of g.
        if(willFlip)
        {
            g2d.translate(width/2, height/2);
            g2d.scale(-1.0, -1.0);
            g2d.translate(-width/2, -height/2);
        }
        //g2d.translate(-45, 0); // Translate the center of our coordinates.
        //g2d.rotate(-0.26);  // Rotate the image by 1 radian.
        AWTDrawVisitor visitor = new AWTDrawVisitor(g2d);
        renderer.paint(this.molecule, visitor, new Rectangle(width, height), true);


    }
}
```

…/src/sp/PolyketideUnit.java

```
package sp;


import java.awt.Image;
import java.io.InputStream;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;

/**
 * PolyketideUnit.java is a class to represent a polyketide unit.
 * This includes both starter and extender units.
 *
 * */

public class PolyketideUnit
{

   /**
    * Constant integers that describe the different synthesis processes.
    *
    * */
   static final int KETOSYNTHASE = 0;
   static final int KETOREDUCTASE = 1;
   static final int DEHYDRATASE = 2;
   static final int ENOYL_REDUCTASE = 3;

   /**
    * PKSModule determines what synthesis process was performed.
    * */
   private int PKSModule = -1;


   private String unitName;                  //The name of the polyketide unit.
   private String forwardSMILES;             //A SMILES representation of the polyketide unit.
   private ImageIcon image;                  //The image of the unit to be used in the menu.
   private String imagePath;                 //The path of the image to be used in the menu.



   /**
    * Constructors of the polyketide unit.
    * */
   public PolyketideUnit(String unitName, String forwardSMILES, String imagePath)
   {
      this.unitName = unitName;
      this.forwardSMILES = (!forwardSMILES.equals("XXX")) ? forwardSMILES : "";
      this.setImagePath(imagePath);
   }

   public PolyketideUnit(String unitName, String forwardSMILES)
   {
      this.unitName = unitName;
      this.forwardSMILES = (!forwardSMILES.equals("XXX")) ? forwardSMILES : "";
   }


   public PolyketideUnit(PolyketideUnit polyketideUnit)
```

```
    {

        this.unitName = polyketideUnit.getUnitName();

        this.forwardSMILES =
polyketideUnit.getForwardSMILES();

    }





    /**

     * @return the unitName

     */

    public String getUnitName()

    {

        return unitName;

    }





    /**

     * @param unitName the unitName to set

     */

    public void setUnitName(String unitName)

    {

        this.unitName = unitName;

    }





    /**

     * @return the forwardSMILES

     */

    public String getForwardSMILES()

    {

        return forwardSMILES;

    }





    /**

     * @param forwardSMILES the forwardSMILES to set

     */

    public void setForwardSMILES(String forwardSMILES)

    {

        this.forwardSMILES = forwardSMILES;

    }





    /**

     * @return the image

     */

    public ImageIcon getImage()

    {

        return image;

    }





    /**

     * @param image the image to set

     */

    public void setImage(String imageName)

    {
```

```
    try
    {
      String unitFolder = "/resources/";
      InputStream in =
getClass().getResourceAsStream(unitFolder+imageName);
      if(in == null)
      {
        in =
getClass().getResourceAsStream("/"+imageName);
      }

      //Read from an input stream

      Image tempImage = ImageIO.read(in);
      this.image= new ImageIcon(tempImage);

      int NEW_WIDTH = (int)
(this.image.getIconWidth()*0.75);
      int NEW_HEIGHT = (int)
(this.image.getIconHeight()*0.75);
      tempImage = tempImage.getScaledInstance(
NEW_WIDTH, NEW_HEIGHT,  Image.SCALE_SMOOTH );

      this.image = new ImageIcon(tempImage);

    }
    catch (Exception e1)
    {
      e1.printStackTrace();
    }
  }
```

```
/**
 * @return the imagePath
 */
public String getImagePath()
{
  return imagePath;
}


/**
 * @param imagePath the imagePath to set
 */
public void setImagePath(String imagePath)
{
  this.imagePath = imagePath;
}

/**
 * @return the pKSModule
 */
public int getPKSModule()
{
  return PKSModule;
}


/**
 * @param pKSModule the pKSModule to set
 */
```

```
public void setPKSModule(int pKSModule)

{

   PKSModule = pKSModule;

}



/**

 *

 * Methods that manipulates the SMILES string.

 * These methods deal with the actual chemical synthesis.

 *

 * */


public void doNothing()

{

   this.setPKSModule(KETOSYNTHASE);

   this.forwardSMILES = "C(=O)C" + forwardSMILES;

}


public void addDehydretase()

{

   this.setPKSModule(DEHYDRATASE);

   this.forwardSMILES = "=CC" + forwardSMILES;

}


public void addKetoReductase()

{

   this.setPKSModule(KETOREDUCTASE);

   this.forwardSMILES = "C(O)C" + forwardSMILES;

}

   public void addEnoylReductase()

   {

      this.setPKSModule(ENOYL_REDUCTASE);

      this.forwardSMILES = "CC" + forwardSMILES;

   }


}
```

…/src/sp/PolyketideUnitFrame.java

```
package sp;


import java.awt.GridLayout;

import java.util.ArrayList;


import javax.swing.JButton;

import javax.swing.JDialog;

import javax.swing.JPanel;


@SuppressWarnings("serial")

public class PolyketideUnitFrame extends JDialog

{


   private ArrayList<JButton> buttonSet;

   private JPanel layoutPanel;


   public PolyketideUnitFrame(boolean starter)

   {


      layoutPanel = new JPanel();


      if(starter)
```

```
    {
      layoutPanel.setLayout(new GridLayout(4, 0));
      this.setTitle("Select starter unit");
      buttonSet = Predyketide.starterUnitSelectionButton;
    }
    else
    {
      layoutPanel.setLayout(new GridLayout(2, 2));
      this.setTitle("Select extender unit");
      buttonSet =
Predyketide.extenderUnitSelectionButton;
    }

for(JButton button : buttonSet)
    {
      layoutPanel.add(button);
    }

    this.add(layoutPanel);
    setResizable(false);
    this.pack();
    this.setLocationRelativeTo(null);
    this.setModal(true);
  }


  public void showPolyketideMenu()
  {
    setVisible(true);
  }
  public void closePolyketideMenu()
  {
    dispose();
    setVisible(false);
  }


  public void addNewCustomizedButton(PolyketideUnit
pkUnit, boolean starter)
  {
    //TODO on creation of a new extender and starter unit.
    if(starter)
      Predyketide.starterUnits.add(pkUnit);
    pkUnit.setImage(pkUnit.getImagePath());
    JButton newButton = new JButton(pkUnit.getImage());
    //
    //edit this to use the universal action listener
    //newButton.addActionListener(new
Predyketide_Events());
    buttonSet.add(newButton);
    layoutPanel.add(newButton);
    layoutPanel.revalidate();
  }


}
```

…/src/sp/PredictionResultsFrame.java

```
package sp;


import java.awt.BasicStroke;
import java.awt.BorderLayout;
import java.awt.Color;
```

```
import java.awt.FlowLayout;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.image.BufferedImage;

import java.util.ArrayList;

import java.util.Collections;



import javax.swing.ButtonGroup;

import javax.swing.GroupLayout;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

import javax.swing.JToggleButton;

import javax.swing.LayoutStyle;



import org.openscience.cdk.exception.CDKException;

import org.openscience.cdk.exception.InvalidSmilesException;


import chemaxon.marvin.beans.MViewPane;


import utils.FileManipulation;

import utils.ResultRankingByNPScore;
```

```
/**

 *

 * @author user

 */

@SuppressWarnings("serial")

public class PredictionResultsFrame extends JFrame implements Runnable

{



   // Variables declaration - do not modify


   @SuppressWarnings("unused")

   private int RESULT_COUNT = 5;


   @SuppressWarnings("unused")

   private ArrayList<PolyketideUnit> workingPKS;

   private ArrayList<String> SMILES_Strings;


   private ArrayList<PredyketideCompoundResult> predictedResults;



   private PredyketideWorkspaceVars workspace;

   private JPanel synthesisMainPanel;


   private PredyketideCompoundResult selectedResult;


   private JPanel resultsListPanel;

   private JScrollPane resultsListScrollPane;

   private JScrollPane resultInfoScrollPane;
```

```
    private JTextArea resultInfoTextArea;

    private JButton resultsCancelButton;

    private JButton resultsExportButton;

    // End of variables declaration



    protected int SYN_WIDTH;



    protected int SYN_HEIGHT;





    /** Creates new form PredictionResultsFrame

     * @param i

     * @param totalWidth */

    public PredictionResultsFrame(ArrayList<String> SMILES_Strings, ArrayList<PolyketideUnit> workingPKS,

                            JPanel synthesisPanel, PredyketideWorkspaceVars workspace, int width, int height)

    {



        this.SMILES_Strings = SMILES_Strings;

        this.workingPKS = workingPKS;

        this.predictedResults = new ArrayList<PredyketideCompoundResult>();

        this.workspace = workspace;

        this.synthesisMainPanel = synthesisPanel;





        this.SYN_HEIGHT = height;

        this.SYN_WIDTH = width;

        initComponents();



    }



    /** This method is called from within the constructor to

     * initialize the form.

     * WARNING: Do NOT modify this code. The content of this method is

     * always regenerated by the Form Editor.

     */

    // <editor-fold defaultstate="collapsed" desc="Generated Code">

    private void initComponents()

    {



        resultsListScrollPane = new JScrollPane();

        resultsListPanel = new JPanel();

        resultInfoScrollPane = new JScrollPane();

        resultInfoTextArea = new JTextArea();

        resultsCancelButton = new JButton();

        resultsExportButton = new JButton();

        resultsListPanel.setBackground(new Color(255, 255, 255));



        GroupLayout resultsListPanelLayout = new GroupLayout(resultsListPanel);

        resultsListPanel.setLayout(new FlowLayout(FlowLayout.CENTER));

        resultsListPanelLayout.setHorizontalGroup(

resultsListPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)

            .addGap(0, 624, Short.MAX_VALUE)

        );

        resultsListPanelLayout.setVerticalGroup(
```

```
resultsListPanelLayout.createParallelGroup(GroupLayout.Alig
nment.LEADING)
      .addGap(0, 167, Short.MAX_VALUE)
    );

    resultsListScrollPane.setViewportView(resultsListPanel);

    resultInfoTextArea.setColumns(20);
    resultInfoTextArea.setRows(5);
    resultInfoTextArea.setEditable(false);

resultInfoScrollPane.setViewportView(resultInfoTextArea);


    resultsCancelButton.setText("Cancel");
    resultsCancelButton.addActionListener(new
ActionListener()
    {

      @Override
      public void actionPerformed(ActionEvent arg0)
      {
        dispose();
      }
    });

    resultsExportButton.setText("Export");
    resultsExportButton.setEnabled(false);
    resultsExportButton.addActionListener(new
ActionListener()
    {

      @Override
      public void actionPerformed(ActionEvent e)
      {

FileManipulation.exportPredictionAsZip(selectedResult,
workspace, synthesisMainPanel,
                            SYN_WIDTH, SYN_HEIGHT);
      }
    });


    GroupLayout layout = new
GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(GroupLayout.Alignment.LEADING
)
      .addGroup(layout.createSequentialGroup()
           .addContainerGap()

.addGroup(layout.createParallelGroup(GroupLayout.Alignme
nt.LEADING)
                .addComponent(resultsListScrollPane,
GroupLayout.DEFAULT_SIZE, 626, Short.MAX_VALUE)
                .addComponent(resultInfoScrollPane,
GroupLayout.DEFAULT_SIZE, 626, Short.MAX_VALUE)


.addGroup(GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                     .addComponent(resultsExportButton)

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATE
D)

.addComponent(resultsCancelButton)))
```

```
                .addContainerGap())
        );
        layout.setVerticalGroup(
layout.createParallelGroup(GroupLayout.Alignment.LEADING
)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(resultsListScrollPane,
GroupLayout.PREFERRED_SIZE, 169,
GroupLayout.PREFERRED_SIZE)
.addPreferredGap(LayoutStyle.ComponentPlacement.RELATE
D)
                .addComponent(resultInfoScrollPane,
GroupLayout.DEFAULT_SIZE, 234, Short.MAX_VALUE)
                .addGap(12, 12, 12)
.addGroup(layout.createParallelGroup(GroupLayout.Alignme
nt.BASELINE)
                    .addComponent(resultsCancelButton)
                    .addComponent(resultsExportButton))
                .addContainerGap())
        );

resultsListScrollPane.setSize(PredyketideCompoundResult.IM
AGE_THUMBLENGTH,
PredyketideCompoundResult.IMAGE_THUMBLENGTH);
```

```
        this.setTitle("Ranked Polyketide Natural Products");
        this.pack();
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        this.setVisible(true);
    }// </editor-fold>


    private void createRankedResult()
    {


for(String SMILES: this.SMILES_Strings)
        {

            PredyketideCompoundResult pksPredictionCompound
= new PredyketideCompoundResult(SMILES);
            this.predictedResults.add(pksPredictionCompound);

            Collections.sort(predictedResults, new
ResultRankingByNPScore());
            //draw on Panel

            try
            {
                pksPredictionCompound.generateResultImage();
            }
            catch (InvalidSmilesException e)
            {
                e.printStackTrace();
            }
```

```
      resultsListPanel.removeAll();

      resultsListPanel.revalidate();

      ButtonGroup bg = new ButtonGroup();

for(PredyketideCompoundResult predictionInstance :
predictedResults)

      {

        //TODO



        Image image =
predictionInstance.getThumbnail().getImage();


predictionInstance.getThumbnail().setImage(image);



        int borderWidth = 3;

        int spaceAroundIcon = 0;

        Color borderColor = Color.GREEN;



        BufferedImage bi = new
BufferedImage(predictionInstance.getThumbnail().getIconWi
dth() + (2 * borderWidth + 2 *
spaceAroundIcon),predictionInstance.getThumbnail().getIcon
Height() + (2 * borderWidth + 2 * spaceAroundIcon),
BufferedImage.TYPE_INT_ARGB);



        Graphics2D g = bi.createGraphics();

        g.setColor(borderColor);


g.drawImage(predictionInstance.getThumbnail().getImage(),
borderWidth + spaceAroundIcon, borderWidth +
spaceAroundIcon, null);



        BasicStroke stroke = new BasicStroke(3); //3 pixels
wide (thickness of the border)

        g.setStroke(stroke);



        g.drawRect(0, 0, bi.getWidth() - 1, bi.getHeight() -
1);
```

```
        g.dispose();



        JToggleButton aPredictionButton = new
JToggleButton(predictionInstance.getThumbnail());

        aPredictionButton.setSelectedIcon(new
ImageIcon(bi));

        bg.add(aPredictionButton);

        aPredictionButton.addMouseListener(new
PredictionMouseAction(predictionInstance));

        resultsListPanel.add(aPredictionButton);

      }



      resultsListPanel.revalidate();



    }







  }





  @Override

  public void run()

  {

    createRankedResult();

  }



  public class PredictionMouseAction implements
MouseListener

  {



    PredyketideCompoundResult aPrediction;
```

```
    public
PredictionMouseAction(PredyketideCompoundResult
pksPredictionCompound)
    {
      this.aPrediction = pksPredictionCompound;
    }

    @Override
    public void mouseClicked(MouseEvent arg0)
    {
      if(arg0.getClickCount() == 2)
      {
        JFrame sketchPad = new JFrame();
sketchPad.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        MViewPane msketchPane = new MViewPane();
        msketchPane.setM(0,
this.aPrediction.getChemaxonMolecule());
        sketchPad.getContentPane().setLayout(new
BorderLayout());
        sketchPad.getContentPane().add(msketchPane,
BorderLayout.CENTER);

        sketchPad.pack();
        sketchPad.setLocationRelativeTo(null);
        sketchPad.setVisible(true);
        sketchPad.requestFocus();
      }
    }

    @Override
    public void mouseEntered(MouseEvent arg0)
    {
```

```
    }

    @Override
    public void mouseExited(MouseEvent arg0)
    {

    }

    @Override
    public void mousePressed(MouseEvent arg0)
    {

    }

    @Override
    public void mouseReleased(MouseEvent arg0)
    {
      try
      {
resultInfoTextArea.setText(aPrediction.getDescriptors());
        resultInfoTextArea.setCaretPosition(0);
        resultsExportButton.setEnabled(true);
        selectedResult = this.aPrediction;
      }
      catch (CDKException e)
      {
        e.printStackTrace();
      }
    }
```

```
   }



}
```

…/src/sp/Predyketide.java

```
package sp;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.event.InputEvent;

import java.awt.event.KeyEvent;

import java.util.ArrayList;


import javax.swing.*;


import org.openscience.cdk.DefaultChemObjectBuilder;

import org.openscience.cdk.interfaces.IAtomContainer;

import org.openscience.cdk.smiles.SmilesParser;



/**

 *

 * @author user

 */

@SuppressWarnings("serial")

public class Predyketide extends JFrame

{



   // Variables declaration - do not modify

   static ArrayList<IAtomContainer> NP_ReferenceData;

   static ArrayList<IAtomContainer> Synthetic_ReferenceData;

   static SmilesParser smpars = new SmilesParser(DefaultChemObjectBuilder.getInstance());


   static JButton aboutButton;

   static JButton exitButton;


   static  JMenuItem aboutMenuItem;

   private JMenu fileMenu;

   private JPanel guiMainPanel;


   static JMenuItem helpMenuItem;


   static JMenuItem newSynthesisMenuItem;

   static JMenuItem exitMenuItem;

   static JMenuItem importSynthesisMenuItem;

   static JMenuItem exportSynthesisMenuItem;

   static JMenuItem exportImageMenuItem;


   private JMenu optionsMenu;


   private JMenu referenceDataMenu;

   static JMenuItem changeNatProdMenuItem;

   static JMenuItem changeSyntheticMenuItem;


   private JMenuBar predyketideMenu;


   static JPanel synthesisMainPanel;

   static JScrollPane synthesisScrollPanel;
```

```
private Predyketide_Events eventsHandler;


static ArrayList<JButton> starterUnitSelectionButton;

static ArrayList<JButton> extenderUnitSelectionButton;


static JButton starterSelectButton;

static JButton synthesis_DH_Button;

static JButton synthesis_ER_Button;

static JButton synthesis_KR_Button;

static JButton synthesis_Plain_Button;


static JButton generatePredictionsButton;


static StarterUnitCollection allStarterUnits;

static ExtenderUnitCollection allExtenderUnits;

static ArrayList<PolyketideUnit> starterUnits;

static ArrayList<PolyketideUnit> extenderUnits;


static PolyketideUnitFrame selectStarterMenu;

static PolyketideUnitFrame selectExtenderMenu;

public static String helpString;


// End of variables declaration


/** Creates new form Predyketide_GUI

 * @throws InterruptedException */

public Predyketide() throws InterruptedException

{

    PredyketideWelcome welcome = new PredyketideWelcome();

    welcome.setStatusLabel("Initializing variables...");

    welcome.repaint();



    eventsHandler = new Predyketide_Events();

    starterUnits = new ArrayList<PolyketideUnit>();

    starterUnitSelectionButton = new ArrayList<JButton>();

    extenderUnits = new ArrayList<PolyketideUnit>();

    extenderUnitSelectionButton = new ArrayList<JButton>();


    //tasks on initializing starter units

    welcome.setStatusLabel("Gathering info on starter units...");

    welcome.repaint();

    allStarterUnits = new StarterUnitCollection();


    welcome.setStatusLabel("Rendering images of starter units...");

    welcome.repaint();

    for(int i = 0; i < starterUnits.size(); i++)

    {

      String unitPath = starterUnits.get(i).getImagePath();

      welcome.setStatusLabel("Rendering " + unitPath);

      welcome.repaint();

      starterUnits.get(i).setImage(unitPath);

    }


    // System.out.println("Creating menu for starter units...");

    welcome.setStatusLabel("Creating menu for starter units...");
```

```
    welcome.repaint();
    for(int i = 0; i < starterUnits.size(); i++)
    {
      PolyketideUnit unit = starterUnits.get(i);

      JButton button = new JButton(unit.getImage());
      starterUnitSelectionButton.add(button);
starterUnitSelectionButton.get(i).addActionListener(eventsHandler);
    }
    selectStarterMenu = new PolyketideUnitFrame(true);


    //tasks on initializing extender units
    //System.out.println("Gathering info on extender units...");
    welcome.setStatusLabel("Gathering info on extender units...");
    welcome.repaint();
    allExtenderUnits = new ExtenderUnitCollection();

    //System.out.println("Rendering images of extender units...");
    welcome.setStatusLabel("Rendering images of extender units...");
    welcome.repaint();

    for(int i = 0; i < extenderUnits.size(); i++)
    {
      String unitPath = extenderUnits.get(i).getImagePath();
      //System.out.println("Rendering " + unitPath);
      welcome.setStatusLabel("Rendering " + unitPath);
      welcome.repaint();
      extenderUnits.get(i).setImage(unitPath);
    }

    //System.out.println("Creating menu for extender units...");
    welcome.setStatusLabel("Creating menu for extender units...");
    welcome.repaint();
    for(int i = 0; i < extenderUnits.size(); i++)
    {
      PolyketideUnit unit = extenderUnits.get(i);

      JButton button = new JButton(unit.getImage());
      extenderUnitSelectionButton.add(button);
extenderUnitSelectionButton.get(i).addActionListener(eventsHandler);
    }
    selectExtenderMenu = new PolyketideUnitFrame(false);

    welcome.setStatusLabel("Fetching NATPROD reference data...");
    welcome.repaint();
    NP_ReferenceData = new ReferenceData(true);
    //System.out.print(NP_ReferenceData.size());

    welcome.setStatusLabel("Fetching SYNTHETIC reference data...");
    welcome.repaint();
    Synthetic_ReferenceData = new ReferenceData(false);
    //System.out.println(Synthetic_ReferenceData.size());

    welcome.setStatusLabel("Initializing user interface...");
    welcome.repaint();
```

```
        Thread.sleep(5000);


        welcome.dispose();
        initComponents();
    }


    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents()
    {


        guiMainPanel = new JPanel();
        starterSelectButton = new JButton();
        synthesis_Plain_Button = new JButton();
        synthesis_KR_Button = new JButton();
        synthesis_DH_Button = new JButton();
        synthesis_ER_Button = new JButton();
        exitButton = new JButton();
        aboutButton = new JButton();
        generatePredictionsButton = new JButton();
        synthesisScrollPanel = new JScrollPane();


        synthesisMainPanel = new JPanel();
        predyketideMenu = new JMenuBar();
        referenceDataMenu = new JMenu();
        fileMenu = new JMenu();
        changeNatProdMenuItem = new JMenuItem();
        changeSyntheticMenuItem = new JMenuItem();
        newSynthesisMenuItem = new JMenuItem();
        exportImageMenuItem = new JMenuItem();
        exitMenuItem = new JMenuItem();
        importSynthesisMenuItem = new JMenuItem();
        exportSynthesisMenuItem = new JMenuItem();
        optionsMenu = new JMenu();
        helpMenuItem = new JMenuItem();
        aboutMenuItem = new JMenuItem();


setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);



        starterSelectButton.setText("Starter");
        starterSelectButton.addActionListener(eventsHandler);


        synthesis_Plain_Button.setText("KS-AT-ACP");
        synthesis_Plain_Button.setEnabled(false);

synthesis_Plain_Button.addActionListener(eventsHandler);


        synthesis_KR_Button.setText("KS-AT-KR-ACP");
        synthesis_KR_Button.setEnabled(false);
        synthesis_KR_Button.addActionListener(eventsHandler);


        synthesis_DH_Button.setText("KS-AT-DH-KR-ACP");
        synthesis_DH_Button.setEnabled(false);

synthesis_DH_Button.addActionListener(eventsHandler);
```

```
        synthesis_ER_Button.setText("KS-AT-DH-ER-KR-ACP");
        synthesis_ER_Button.setEnabled(false);
        synthesis_ER_Button.addActionListener(eventsHandler);

        exitButton.setText("Exit");
        exitButton.addActionListener(eventsHandler);

        aboutButton.setText("About");
        aboutButton.addActionListener(eventsHandler);

        generatePredictionsButton.setText("Predictions");

generatePredictionsButton.addActionListener(eventsHandler
);
        generatePredictionsButton.setEnabled(false);

        synthesisScrollPanel.setBackground(new Color(255, 255,
255));
        synthesisScrollPanel.setAutoscrolls(true);

        synthesisMainPanel.setBackground(new Color(255, 255,
255));

        GroupLayout synthesisMainPanelLayout = new
GroupLayout(synthesisMainPanel);
        synthesisMainPanel.setLayout(new
FlowLayout(FlowLayout.LEFT));
        synthesisMainPanelLayout.setHorizontalGroup(

synthesisMainPanelLayout.createParallelGroup(GroupLayout
.Alignment.LEADING)
            .addGap(0, 818, Short.MAX_VALUE)
        );
        synthesisMainPanelLayout.setVerticalGroup(

synthesisMainPanelLayout.createParallelGroup(GroupLayout
.Alignment.LEADING)
            .addGap(0, 402, Short.MAX_VALUE)
        );

synthesisScrollPanel.setViewportView(synthesisMainPanel);

        GroupLayout guiMainPanelLayout = new
GroupLayout(guiMainPanel);
        guiMainPanel.setLayout(guiMainPanelLayout);
        guiMainPanelLayout.setHorizontalGroup(

guiMainPanelLayout.createParallelGroup(GroupLayout.Align
ment.LEADING)
            .addGroup(GroupLayout.Alignment.TRAILING,
guiMainPanelLayout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(synthesisScrollPanel,
GroupLayout.DEFAULT_SIZE, 820, Short.MAX_VALUE)

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(guiMainPanelLayout.createParallelGroup(GroupL
ayout.Alignment.LEADING, false)

.addComponent(generatePredictionsButton,
GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                            .addComponent(starterSelectButton,
GroupLayout.DEFAULT_SIZE, 150, Short.MAX_VALUE)
                            .addComponent(synthesis_Plain_Button,
GroupLayout.DEFAULT_SIZE, 150, Short.MAX_VALUE)
                            .addComponent(synthesis_KR_Button,
GroupLayout.DEFAULT_SIZE, 150, Short.MAX_VALUE)
                            .addComponent(synthesis_DH_Button,
GroupLayout.Alignment.TRAILING,
GroupLayout.DEFAULT_SIZE, 150, Short.MAX_VALUE)
```

```
                        .addComponent(synthesis_ER_Button, GroupLayout.Alignment.TRAILING, GroupLayout.DEFAULT_SIZE, 150, Short.MAX_VALUE)
                        .addComponent(exitButton, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(aboutButton, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap())
    );
    guiMainPanelLayout.setVerticalGroup(
guiMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
.addGroup(guiMainPanelLayout.createSequentialGroup()
                .addGap(63, 63, 63)
.addGroup(guiMainPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
.addGroup(guiMainPanelLayout.createSequentialGroup()
                            .addComponent(starterSelectButton, GroupLayout.PREFERRED_SIZE, 23, GroupLayout.PREFERRED_SIZE)
.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
.addComponent(synthesis_Plain_Button, GroupLayout.PREFERRED_SIZE, 23, GroupLayout.PREFERRED_SIZE)
.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
.addComponent(synthesis_KR_Button, GroupLayout.PREFERRED_SIZE, 23, GroupLayout.PREFERRED_SIZE)
.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
.addComponent(synthesis_DH_Button, GroupLayout.PREFERRED_SIZE, 23, GroupLayout.PREFERRED_SIZE)
.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
.addComponent(synthesis_ER_Button, GroupLayout.PREFERRED_SIZE, 23, GroupLayout.PREFERRED_SIZE)
.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 184, Short.MAX_VALUE)
.addComponent(generatePredictionsButton)
.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                              .addComponent(aboutButton)
.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                              .addComponent(exitButton))
                        .addComponent(synthesisScrollPanel, GroupLayout.DEFAULT_SIZE, 404, Short.MAX_VALUE))
                .addContainerGap())
    );
```

```
        fileMenu.setText("File");



newSynthesisMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N, InputEvent.CTRL_MASK));

        newSynthesisMenuItem.setText("Create new synthesis");

newSynthesisMenuItem.addActionListener(eventsHandler);

        fileMenu.add(newSynthesisMenuItem);


        fileMenu.addSeparator();



importSynthesisMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O, InputEvent.CTRL_MASK));

        importSynthesisMenuItem.setText("Import synthesis");

importSynthesisMenuItem.addActionListener(eventsHandler);

        fileMenu.add(importSynthesisMenuItem);



exportSynthesisMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_S, InputEvent.CTRL_MASK));

        exportSynthesisMenuItem.setText("Export synthesis");

exportSynthesisMenuItem.addActionListener(eventsHandler);

        fileMenu.add(exportSynthesisMenuItem);



exportImageMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_P, InputEvent.CTRL_MASK));

        exportImageMenuItem.setText("Save synthesis as JPG");

exportImageMenuItem.addActionListener(eventsHandler);

        fileMenu.add(exportImageMenuItem);


        referenceDataMenu.setText("Change reference data");


        changeNatProdMenuItem.setText("Natural product data");

changeNatProdMenuItem.addActionListener(eventsHandler);

        referenceDataMenu.add(changeNatProdMenuItem);


        changeSyntheticMenuItem.setText("Synthetic data");

changeSyntheticMenuItem.addActionListener(eventsHandler);

        referenceDataMenu.add(changeSyntheticMenuItem);


        fileMenu.addSeparator();


        fileMenu.add(referenceDataMenu);


        fileMenu.addSeparator();



exitMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_Q, InputEvent.CTRL_MASK));

        exitMenuItem.addActionListener(eventsHandler);

        exitMenuItem.setText("Exit");

        fileMenu.add(exitMenuItem);


        predyketideMenu.add(fileMenu);


        optionsMenu.setText("Options");
```

```
helpMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_H, InputEvent.SHIFT_MASK | InputEvent.CTRL_MASK));
        helpMenuItem.setText("Help");
        helpMenuItem.addActionListener(eventsHandler);
        optionsMenu.add(helpMenuItem);


aboutMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_A, InputEvent.SHIFT_MASK | InputEvent.CTRL_MASK));
        aboutMenuItem.setText("About");
        aboutMenuItem.addActionListener(eventsHandler);
        optionsMenu.add(aboutMenuItem);

        predyketideMenu.add(optionsMenu);

        setJMenuBar(predyketideMenu);

        GroupLayout layout = new GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
layout.createParallelGroup(GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(guiMainPanel, GroupLayout.DEFAULT_SIZE, 996, Short.MAX_VALUE)
                .addContainerGap())
        );
        layout.setVerticalGroup(
layout.createParallelGroup(GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(guiMainPanel, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
        );
        setTitle("Predyketide v1.0");
        setResizable(false);
        pack();
        setLocationRelativeTo(null);
    }// </editor-fold>

}
```

…/src/sp/Predyketide_Events.java

```
/**
 * @author John Althom A. Mendoza
 *
 * The ActionListener class of the whole application.
 *
 * */

package sp;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
```

```
import javax.swing.JOptionPane;

import javax.swing.JPanel;


import org.openscience.cdk.interfaces.IAtomContainer;




import utils.FileManipulation;




public class Predyketide_Events implements ActionListener

{



   private PredyketideWorkspaceVars workspaceVariables =
new PredyketideWorkspaceVars();



   /**

    * height and width per segment of the biosynthesis

    */

   private int HEIGHT = 350;

   private int baseWidth =130;
                          //baseWidth, to be referenced when a
new synthesis is made

   private int WIDTH = 130;

   private int totalWidth = 0;





   /**

    * height of the supporting text

    */

   private int captionHeight = 30;

   //caption width will always be equal to panel width
```

```
   private double widthFactor = 15;
            //widthfactor for larger molecules

   private double smallWidthFactor = 15;
            //widthFactor for small molecules



   private String currModule = "";

   private String workingSMILES = "";


   private PolyketideUnit currUnit;



   private boolean isNewlyImported = false;



   @Override

   public void actionPerformed(ActionEvent e)

   {





      if(e.getSource() == Predyketide.starterSelectButton)

      {

         /**

          * Selects the starter button. Shows all the available
starter units.

          */


Predyketide.selectStarterMenu.showPolyketideMenu();



      }

      else
if(Predyketide.starterUnitSelectionButton.contains(e.getSour
ce()))

      {



         /**
```

```
       * Selects a starter.

       */

      int index =
Predyketide.starterUnitSelectionButton.indexOf(e.getSource(
));

      workingSMILES = getWorkingSMILES_Starter(index);

workspaceVariables.addSequenceOfSMILES(workingSMILES);


      if(workingSMILES.length() < 12)

      {

        widthFactor = this.smallWidthFactor;

      }


      paintNewSynthesisPanel(workingSMILES, "Starter
Unit", currUnit.getUnitName());



      updateGUI_Buttons(false);

Predyketide.selectStarterMenu.closePolyketideMenu();


    }

    else if(e.getSource() ==
Predyketide.synthesis_Plain_Button)

    {

      /**

       * Selects the KS-AT-ACP button.

       */

      if(workspaceVariables.isLimited())

      {

        JOptionPane.showMessageDialog(null, "Maximum
of " + PredyketideWorkspaceVars.LIMIT + " units are
allowed!");

        return;

      }

      currModule = "KS-AT-ACP";

Predyketide.selectExtenderMenu.showPolyketideMenu();


    }

    else if(e.getSource() ==
Predyketide.synthesis_KR_Button)

    {

      /**

       * Selects the KS-AT-KR-ACP button.

       */

      if(workspaceVariables.isLimited())

      {

        JOptionPane.showMessageDialog(null, "Maximum
of " + PredyketideWorkspaceVars.LIMIT + " units are
allowed!");

        return;

      }

      currModule = "KS-AT-KR-ACP";

Predyketide.selectExtenderMenu.showPolyketideMenu();


    }

    else if(e.getSource() ==
Predyketide.synthesis_DH_Button)

    {

      /**

       * Selects the KS-AT-DH-KR-ACP button.

       */

      if(workspaceVariables.isLimited())

      {

        JOptionPane.showMessageDialog(null, "Maximum
of " + PredyketideWorkspaceVars.LIMIT + " units are
allowed!");
```

```
            return;

          }

          currModule = "KS-AT-DH-KR-ACP";


Predyketide.selectExtenderMenu.showPolyketideMenu();



        }

        else if(e.getSource() ==
Predyketide.synthesis_ER_Button)

        {

          /**

           * Selects the KS-AT-DH-ER-KR-ACP button.

           */

          if(workspaceVariables.isLimited())

          {

            JOptionPane.showMessageDialog(null, "Maximum
of " + PredyketideWorkspaceVars.LIMIT + " units are
allowed!");

            return;

          }

          currModule = "KS-AT-DH-ER-KR-ACP";


Predyketide.selectExtenderMenu.showPolyketideMenu();



        }

        else
if(Predyketide.extenderUnitSelectionButton.contains(e.getSo
urce()))

        {



          isNewlyImported = false;

          /**

           * Selects an extender unit.

           */

          int index =
Predyketide.extenderUnitSelectionButton.indexOf(e.getSour
ce());



          workingSMILES = getWorkingSMILES_Extender(index);


workspaceVariables.addSequenceOfSMILES(workingSMILES);



          paintNewSynthesisPanel(workingSMILES, currModule,
currUnit.getUnitName());



          updateGUI_Buttons(false);


Predyketide.selectExtenderMenu.closePolyketideMenu();

        }

        else if(e.getSource() ==
Predyketide.newSynthesisMenuItem)

        {

          /**

           * Chooses the "Create new synthesis" menu item.

           * Clears all variables.

           *

           *

           */

          int toSave = -1;

          if(workspaceVariables.getSequenceOfSMILES().size() >
0)

          {

            toSave = askVerification("Do you want to save this
workspace\nbefore creating a new one?");

          }



          if(toSave == JOptionPane.CANCEL_OPTION)

          {
```

```
        return;
      }
      else if(toSave == JOptionPane.YES_OPTION)
      {
        workspaceVariables.setLastUnit(currUnit);
        FileManipulation.exportSynthesisGUI(workspaceVariables);
      }
      revalidateWorkspacePanel(true);
      updateGUI_Buttons(true);
    }
    else if(e.getSource() == Predyketide.generatePredictionsButton)
    {
      /**
       * Clicks the "Predictions" button.
       *
       * (O) signifies an OH in the actual chemical molecule. This is the place
       *              where thioesterification will occur.
       */
      String findStr = "(O)";
      ArrayList<String> resultsSMILES = new ArrayList<String>();

      String resultCompundSMILES = "";

      if(isNewlyImported)
      {
        resultCompundSMILES = "SC(=O)C" + workingSMILES;
      }
      else
      {
        resultCompundSMILES = workingSMILES;
      }

      int lastIndex = 0; //to eliminate a ring having less than 5 atoms
      int ringNumber = 1;
      while(resultCompundSMILES.indexOf(""+ringNumber) >= 0)
      {
        ringNumber++;
      }



      while(lastIndex != -1)
      {
        //assures that the OH in the starter are not candidates for the thioesterification
        lastIndex = resultCompundSMILES.indexOf(findStr,lastIndex);

        if( lastIndex != -1)
        {
          //there is an OH
          String newRing = resultCompundSMILES.substring(1, lastIndex);

          String aPrediction = "O" + ringNumber+ newRing;
          String remainder = resultCompundSMILES.substring(lastIndex+3, resultCompundSMILES.length());
          if(remainder.length() > 0)
            aPrediction += "(" +remainder + ")" + ringNumber;
          else
```

```
            aPrediction += ringNumber;


          // System.out.println(aPrediction);

          if(newRing.replaceAll("\\([^\\(]*\\)", "").length()
>= 5)

            resultsSMILES.add(aPrediction);

          lastIndex+=findStr.length();

        }

      }


      /**

       * Condition if the candidates are less than or equal to
2.

       * This will add the polyketide chain to the candidates
list.

       * */

      if(resultsSMILES.size() <= 2)

      {

        resultsSMILES.add(0, "O" +
resultCompundSMILES.substring(1));

        //adds the polyketide chain at the beggining of the
list

      }


      /**

       * This condition is for the preliminary candidate
selection.

       *

       * Due to chemical strain, the more possible chemical
molecules from the compound

       *          will form bonds on the farther OH.

       *

       * The last five will be considered for ranking.

       * */
```

```
      ArrayList<String> top5Molecules = new
ArrayList<String>();

      if(resultsSMILES.size() > 5)

      {

        for(int init = resultsSMILES.size()-1; ; init--)

        {

          top5Molecules.add(resultsSMILES.get(init));

          if(top5Molecules.size() >=5 )

            break;

        }

        resultsSMILES = top5Molecules;

      }


      workspaceVariables.setLastUnit(currUnit);

      PredictionResultsFrame generatedResults = new
PredictionResultsFrame(resultsSMILES,


workspaceVariables.getWorkingPolyketideUnits(),Predyketid
e.synthesisMainPanel, workspaceVariables,

          totalWidth, HEIGHT+captionHeight);

      new Thread(generatedResults).start();

    }

    else if(e.getSource() ==
Predyketide.importSynthesisMenuItem)

    {


      /**

       * Imports a text file to the current workspace.

      */

      if(workspaceVariables.getSequenceOfSMILES().size() >
0)

      {

        int toSave = askVerification("Do you want to save
this workspace\nbefore opening a new one?");
```

```
          if(toSave == JOptionPane.CANCEL_OPTION)

          {

            return;

          }

          else if(toSave == JOptionPane.YES_OPTION)

          {

            workspaceVariables.setLastUnit(currUnit);

FileManipulation.exportSynthesisGUI(workspaceVariables);

          }

        }



        //backing up of records

        PredyketideWorkspaceVars oldvars =
workspaceVariables;

        int oldWIDTH = this.WIDTH;

        JPanel oldPanel = Predyketide.synthesisMainPanel;

        PolyketideUnit oldUNit = currUnit;

        int oldTotal = totalWidth;

        //end of backing up



        revalidateWorkspaceVariables(true);

        workspaceVariables =
FileManipulation.importSynthesis();

        if(workspaceVariables == null ||
workspaceVariables.equals(null))

        {

          workspaceVariables = oldvars;



          this.WIDTH = oldWIDTH;

          Predyketide.synthesisMainPanel = oldPanel;



          currUnit = oldUNit;
```

```
          totalWidth = oldTotal;



          return;

        }



        Predyketide.synthesisMainPanel.removeAll();

        Predyketide.synthesisMainPanel.repaint();

        Predyketide.synthesisMainPanel.revalidate();

        currUnit = workspaceVariables.getLastUnit();



        boolean firstUnit = true;

        for(int x = 0; x <
workspaceVariables.getSequenceOfSMILES().size(); x++)

        {

          String SMILES =
workspaceVariables.getSequenceOfSMILES().get(x);

          String labelString = "";

          String unitString = "";

          if(x <
workspaceVariables.getSequenceOfSMILES().size()-1)

            unitString =
workspaceVariables.getWorkingPolyketideUnits().get(x).getU
nitName();

          if(firstUnit)

          {

            labelString = "Starter Unit";

            firstUnit = false;

          }

          else

          {

switch(workspaceVariables.getWorkingPolyketideUnits().get(
x-1).getPKSModule())

            {
```

```
          case PolyketideUnit.KETOSYNTHASE:
            labelString = "KS-AT-ACP";
            break;
          case PolyketideUnit.KETOREDUCTASE:
            labelString = "KS-AT-KR-ACP";
            break;
          case PolyketideUnit.DEHYDRATASE:
            labelString = "KS-AT-DH-KR-ACP";
            break;
          case PolyketideUnit.ENOYL_REDUCTASE:
            labelString = "KS-AT-DH-ER-KR-ACP";
            break;
          }
        }

        paintNewSynthesisPanel(SMILES, labelString,
unitString);

      }

      updateGUI_Buttons(false);
      workingSMILES =
workspaceVariables.getWorkingSMILES();
      isNewlyImported = true;
    }
    else if(e.getSource() ==
Predyketide.exportSynthesisMenuItem)
    {
      /**
       * Exports the workspace into a text file.
       */;
      if(workspaceVariables.getSequenceOfSMILES().size() >
0)
      {
        workspaceVariables.setLastUnit(currUnit);

FileManipulation.exportSynthesisGUI(workspaceVariables);
      }
      else
      {
        JOptionPane.showMessageDialog(null, "No
workspace to save.");
      }

    }
    else if(e.getSource() ==
Predyketide.exportImageMenuItem)
    {
      /**
       * Exports the workspace into a text file.
       */
      if(workspaceVariables.getSequenceOfSMILES().size() >
0)
      {

FileManipulation.exportImageGUI(Predyketide.synthesisMai
nPanel, totalWidth, HEIGHT+captionHeight);
      }
      else
      {
        JOptionPane.showMessageDialog(null, "No
workspace to render.");
      }
    }
    else if(e.getSource() ==
Predyketide.changeNatProdMenuItem)
    {
```

```
      ArrayList<IAtomContainer> newData =
FileManipulation.importReferenceData();

      if(newData.equals(null) || newData == null)

      {



      }

      else

      {

        Predyketide.NP_ReferenceData = newData;

      }

    }

    else if(e.getSource() ==
Predyketide.changeSyntheticMenuItem)

    {

      ArrayList<IAtomContainer> newData =
FileManipulation.importReferenceData();

      if(newData.equals(null) || newData == null)

      {



      }

      else

      {

        Predyketide.Synthetic_ReferenceData = newData;

      }

    }

    else if(e.getSource() == Predyketide.aboutButton ||
e.getSource() == Predyketide.aboutMenuItem)

    {

      new AboutDialog();

    }

    else if(e.getSource() == Predyketide.helpMenuItem)

    {

      new HelpDialog();
```

```
    }

    else if(e.getSource() == Predyketide.exitButton ||
e.getSource() == Predyketide.exitMenuItem)

    {

      /**

       * Closes the application.

       */

      int toSave = -1;

      if(workspaceVariables.getSequenceOfSMILES().size() >
0)

      {

        toSave = askVerification("Do you want to save this
workspace\nbefore closing the application?");

      }



      if(toSave == JOptionPane.CANCEL_OPTION)

      {

        return;

      }

      else if(toSave == JOptionPane.YES_OPTION)

      {

        workspaceVariables.setLastUnit(currUnit);

FileManipulation.exportSynthesisGUI(workspaceVariables);

      }

      System.exit(0);

    }

  }






  private void revalidateWorkspaceVariables(boolean b)
```

```
    {
        if(b)
        {
            this.WIDTH = this.baseWidth ;

            workspaceVariables = new PredyketideWorkspaceVars();
            currUnit = null;
            totalWidth = 0;
        }

        Predyketide.synthesisMainPanel.revalidate();
    }




    /**
     * Function: returns the SMILES string of the requested starter unit.
     * @param starterID
     * */
    private String getWorkingSMILES_Starter(int starterID)
    {

        currUnit = new PolyketideUnit(Predyketide.starterUnits.get(starterID));
        String forwardSMILES = currUnit.getForwardSMILES();
        forwardSMILES = "SC(=O)C" + forwardSMILES;

        return forwardSMILES;
```

```
    }


    private String getWorkingSMILES_Extender(int extenderID)
    {

        if(currModule.equals("KS-AT-ACP"))
        {
            currUnit.doNothing();
        }
        else if(currModule.equals("KS-AT-KR-ACP"))
        {
            currUnit.addKetoReductase();
        }
        else if(currModule.equals("KS-AT-DH-KR-ACP"))
        {
            currUnit.addDehydretase();
        }
        else if(currModule.equals("KS-AT-DH-ER-KR-ACP"))
        {
            currUnit.addEnoylReductase();
        }

workspaceVariables.addWorkingPolyketideUnit(currUnit);
        currUnit = new PolyketideUnit(Predyketide.extenderUnits.get(extenderID));

        String forwardSMILES = "";

for(PolyketideUnit unit : workspaceVariables.getWorkingPolyketideUnits())
```

```
        {
            forwardSMILES = unit.getForwardSMILES() + forwardSMILES;
        }
        forwardSMILES = "SC(=O)C" + currUnit.getForwardSMILES() + forwardSMILES;



        return forwardSMILES;
    }



    private void revalidateWorkspacePanel(boolean isNewPanel)
    {
        if(isNewPanel)
        {
            this.WIDTH = this.baseWidth ;
            Predyketide.synthesisMainPanel.removeAll();
            Predyketide.synthesisMainPanel.repaint();

            workspaceVariables = new PredyketideWorkspaceVars();
            currUnit = null;
            totalWidth = 0;
        }

        Predyketide.synthesisMainPanel.revalidate();
    }



    private void updateGUI_Buttons(boolean isStarterActive)
    {
        Predyketide.starterSelectButton.setEnabled(isStarterActive);

        Predyketide.synthesis_Plain_Button.setEnabled(!isStarterActive);

        Predyketide.synthesis_KR_Button.setEnabled(!isStarterActive);

        Predyketide.synthesis_DH_Button.setEnabled(!isStarterActive);

        Predyketide.synthesis_ER_Button.setEnabled(!isStarterActive);



        if(workspaceVariables.getWorkingPolyketideUnits().size() >= 3)
        {
            Predyketide.generatePredictionsButton.setEnabled(true);
        }
        else
        {
            Predyketide.generatePredictionsButton.setEnabled(false);
        }
    }



    private void paintNewSynthesisPanel(String SMILES, String module, String unit)
    {
        try
        {
```

```
      PolyketideChainPanel newPKPanel = new
PolyketideChainPanel("[S-]"+SMILES.substring(1),
WIDTH+=widthFactor, HEIGHT);



      PolyketideChainColumnPanel newColPanel = new
PolyketideChainColumnPanel(newPKPanel, module, unit,
captionHeight);

      newColPanel.setLayout(new
FlowLayout(FlowLayout.CENTER));

      newColPanel.setPreferredSize(new Dimension(WIDTH,
HEIGHT+captionHeight));

      newColPanel.setBackground(Color.WHITE);



      Predyketide.synthesisMainPanel.add(newColPanel);

      revalidateWorkspacePanel(false);

      totalWidth += WIDTH;

    }

    catch(Exception e)

    {

      e.printStackTrace();

    }

  }


  private int askVerification(String question)

  {

    try

    {

      return JOptionPane.showConfirmDialog(null, question,
"", JOptionPane.YES_NO_CANCEL_OPTION);


    }

    catch(Exception e)

    {

      return JOptionPane.CANCEL_OPTION;

    }


  }


}
```

…/src/sp/PredyketideCompoundResult.java

```
package sp;


import java.awt.Image;

import java.text.DecimalFormat;

import java.util.BitSet;


import javax.swing.ImageIcon;


import org.openscience.cdk.exception.CDKException;

import
org.openscience.cdk.exception.InvalidSmilesException;

import
org.openscience.cdk.fingerprint.HybridizationFingerprinter;

import org.openscience.cdk.interfaces.IAtomContainer;

import
org.openscience.cdk.layout.StructureDiagramGenerator;

import
org.openscience.cdk.qsar.descriptors.molecular.ALOGPDescri
ptor;

import
org.openscience.cdk.qsar.descriptors.molecular.APolDescript
or;

import
org.openscience.cdk.qsar.descriptors.molecular.BPolDescript
or;

import
org.openscience.cdk.qsar.descriptors.molecular.HBondAccep
torCountDescriptor;
```

```
import org.openscience.cdk.qsar.descriptors.molecular.HBondDonorCountDescriptor;

import org.openscience.cdk.qsar.descriptors.molecular.HybridizationRatioDescriptor;

import org.openscience.cdk.qsar.descriptors.molecular.LargestChainDescriptor;

import org.openscience.cdk.qsar.descriptors.molecular.LargestPiSystemDescriptor;

import org.openscience.cdk.qsar.descriptors.molecular.LongestAliphaticChainDescriptor;

import org.openscience.cdk.qsar.descriptors.molecular.TPSADescriptor;

import org.openscience.cdk.qsar.descriptors.molecular.VABCDescriptor;

import org.openscience.cdk.qsar.descriptors.molecular.WeightDescriptor;

import org.openscience.cdk.qsar.descriptors.molecular.XLogPDescriptor;

import org.openscience.cdk.qsar.descriptors.molecular.ZagrebIndexDescriptor;

import org.openscience.cdk.similarity.Tanimoto;


import chemaxon.formats.MolFormatException;

import chemaxon.formats.MolImporter;

import chemaxon.struc.Molecule;


import com.ibm.icu.util.StringTokenizer;


public class PredyketideCompoundResult {


	private final String imageFormat = "jpeg:w800,h800";

	private final String imageFormatThumb = "jpeg:w120,h120,maxscale28";



	private ImageIcon thumbnail;

	private Image image;

	private	String resultSMILES;

	private double NP_LikenessScore = 0;

	private IAtomContainer compoundMolecule;

	private Molecule novel;

	private String formula;

	static final int IMAGE_THUMBLENGTH = 120;

	static final int IMAGE_LENGTH = 400;




	public PredyketideCompoundResult(String resultsSMILES){

		this.resultSMILES = resultsSMILES;

		//System.out.println(resultsSMILES);

		try {

	generatePredictedCompoundMolecule();

			computeScores();

		} catch (InvalidSmilesException e) {

			// TODO Auto-generated catch block

			e.printStackTrace();

		} catch (CDKException e) {

			// TODO Auto-generated catch block

			e.printStackTrace();

		}
```

```
	}

	/**
	 * @return the thumbnail
	 * @throws InvalidSmilesException
	 */
	@SuppressWarnings("deprecation")
	public void generateResultImage() throws InvalidSmilesException {

		try {
			novel = MolImporter.importMol(this.resultSMILES);
			novel.aromatize();
			novel.clean(2, null);
			formula = novel.getFormula();
			byte[] imageData  = novel.toBinFormat(imageFormat);
			this.image = new ImageIcon(imageData).getImage();
			byte[] imageDataThumb  = novel.toBinFormat(imageFormatThumb);
			this.thumbnail = new ImageIcon(imageDataThumb);
		} catch (MolFormatException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
	}


	//create the molecule object
	private void generatePredictedCompoundMolecule() throws InvalidSmilesException {

		IAtomContainer dummyMolecule = (IAtomContainer) Predyketide.smpars.parseSmiles(this.resultSMILES);
    StructureDiagramGenerator sdg = new StructureDiagramGenerator();
    sdg.setMolecule(dummyMolecule);
    try {
      sdg.generateCoordinates();
    }
    catch (Exception ex)
    {
//Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
    }

    this.compoundMolecule = sdg.getMolecule();

	}

	private void computeScores() throws CDKException{

		HybridizationFingerprinter fingerprinter = new HybridizationFingerprinter();
		BitSet bitset1 = fingerprinter.getFingerprint(this.compoundMolecule);

		//computing for the NP score
		float natProdScore = 0;
		for(IAtomContainer anNP : Predyketide.NP_ReferenceData){
```

```
            BitSet bitset2 =
fingerprinter.getFingerprint(anNP);
            float val =
Tanimoto.calculate(bitset1, bitset2);
            if(val != 0)
                natProdScore +=
Math.log(val);
        }

        //computing for the NP score
        float syntheticScore = 0;
        for(IAtomContainer anNP :
Predyketide.Synthetic_ReferenceData){
                BitSet bitset2 =
fingerprinter.getFingerprint(anNP);
                float val =
Tanimoto.calculate(bitset1, bitset2);
                if(val != 0)

    syntheticScore += Math.log(val);
        }

        double ratioSyntheticToNP = ((double)
Predyketide.Synthetic_ReferenceData.size()/(double)
Predyketide.NP_ReferenceData.size());
        this.NP_LikenessScore = natProdScore -
syntheticScore + Math.log(ratioSyntheticToNP);


    }


    public double getNP_LikenessScore(){
        return NP_LikenessScore;
    }



    public IAtomContainer getPredictedMolecule(){
```

```
    return this.compoundMolecule;
}


public ImageIcon getThumbnail(){
    return thumbnail;
}


/**
 * @return the resultSMILES
 */
public String getResultSMILES() {
    return resultSMILES;
}


/**
 * @param resultSMILES the resultSMILES to set
 */
public void setResultSMILES(String resultSMILES) {
    this.resultSMILES = resultSMILES;
}


/**
 * @return the image
 */
public Image getImage() {
    return image;
}


/**
 * @param image the image to set
 */
```

```
	public void setImage(Image image) {

		this.image = image;

	}


	/**

	 * @return the descriptors

	 * @throws CDKException

	 */

	public String getDescriptors() throws CDKException {


		DecimalFormat df = new DecimalFormat("#.00000");


		String returnString = formula;


		returnString += "\nNP-Likeness Score: " + df.format(getNP_LikenessScore());


    ALOGPDescriptor alogp = new ALOGPDescriptor();

    StringTokenizer tokenizeLogP = new StringTokenizer(alogp.calculate(this.compoundMolecule).getValue().toString(), ",");

    returnString += "\n\nALogP - Ghose-Crippen LogKow: " + df.format(Double.parseDouble(tokenizeLogP.nextToken()));

    returnString += "\nALogP2: " + df.format(Double.parseDouble(tokenizeLogP.nextToken()));

    returnString += "\nAMR - molar refractivity: " + df.format(Double.parseDouble(tokenizeLogP.nextToken()));


    APolDescriptor apol = new APolDescriptor();

    Double apolValue = Double.parseDouble(apol.calculate(this.compoundMolecule).getValue().toString());

    returnString += "\nTotal atomic polarizability: " + df.format(apolValue);

    returnString += "\nAtom count: " + this.compoundMolecule.getAtomCount();


    returnString += "\nBond count: " + this.compoundMolecule.getBondCount();


    BPolDescriptor bondPol= new BPolDescriptor();

    Double bondPolValue = Double.parseDouble(bondPol.calculate(this.compoundMolecule).getValue().toString());

    returnString += "\nBond polarizability: " + df.format(bondPolValue);


    HBondDonorCountDescriptor hDonor = new HBondDonorCountDescriptor();

    returnString += "\nHyrdogen bond donors: " + hDonor.calculate(this.compoundMolecule).getValue().toString();


    HBondAcceptorCountDescriptor hAcceptor = new HBondAcceptorCountDescriptor();

    returnString += "\nHyrdogen bond acceptors: " + hAcceptor.calculate(this.compoundMolecule).getValue().toString();


    HybridizationRatioDescriptor hRatio = new HybridizationRatioDescriptor();

    Double hRatioVal = Double.parseDouble(hRatio.calculate(this.compoundMolecule).getValue().toString());

    returnString += "\nHybridization ratio: " + df.format(hRatioVal);


    LargestChainDescriptor chain = new LargestChainDescriptor();

    returnString += "\nLargest chain: " + chain.calculate(this.compoundMolecule).getValue().toString();
```

```
    LargestPiSystemDescriptor piChain = new
LargestPiSystemDescriptor();
    returnString += "\nLargest PI chain: " +
piChain.calculate(this.compoundMolecule).getValue().toStrin
g();


    LongestAliphaticChainDescriptor aliphaticChain = new
LongestAliphaticChainDescriptor();
    returnString += "\nLongest aliphatic chain: " +
aliphaticChain.calculate(this.compoundMolecule).getValue().
toString();


    TPSADescriptor tpsa = new TPSADescriptor();
    Double tpsaVal =
Double.parseDouble(tpsa.calculate(this.compoundMolecule).
getValue().toString());
    returnString += "\nTopological surface area: " +
df.format(tpsaVal);


    VABCDescriptor vabc = new VABCDescriptor();
    Double vabcVal =
Double.parseDouble(vabc.calculate(this.compoundMolecule)
.getValue().toString());
    returnString += "\nVolume: " + df.format(vabcVal);


    WeightDescriptor weight = new WeightDescriptor();
    Double weightVal =
Double.parseDouble(weight.calculate(this.compoundMolecul
e).getValue().toString());
    returnString += "\nWeight: " + df.format(weightVal);


    XLogPDescriptor xlogp = new XLogPDescriptor();
    Double xlogVal =
Double.parseDouble(xlogp.calculate(this.compoundMolecule
).getValue().toString());
    returnString += "\nXLogP: " + df.format(xlogVal);


    ZagrebIndexDescriptor zagreb = new
ZagrebIndexDescriptor();
    Double zagrebVal =
Double.parseDouble(zagreb.calculate(this.compoundMolecul
e).getValue().toString());
    returnString += "\nZagreb index: " +
df.format(zagrebVal);


                return returnString;
        }


        /**
         * @return the chemaxonMolecule
         */
        public Molecule getChemaxonMolecule() {
                return novel;
        }


}
```

…/src/sp/PredyketideWelcome.java

```
package sp;


import java.awt.Color;
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.io.InputStream;


import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
```

```
import javax.swing.JPanel;



@SuppressWarnings("serial")

public class PredyketideWelcome extends JFrame

{


   private BufferedImage image;

   private JPanel mainWelcomePanel;

   private String statusLabel = "Initializing...";


   private int StringX = 70;

   private int StringY = 300;



   public PredyketideWelcome()

   {


      mainWelcomePanel = new JPanel();


      try

      {


         String logo = "/resources/logo.jpg";

         InputStream in =
getClass().getResourceAsStream(logo);

         if(in == null)

         {

            in = getClass().getResourceAsStream("/logo.jpg");

         }

         this.image= ImageIO.read(in);


         mainWelcomePanel.add(new JLabel(new
ImageIcon(image)));

         mainWelcomePanel.setBackground(new
Color(0,0,0,0));

         this.add(mainWelcomePanel);

         this.setUndecorated(true);

         this.pack();

         this.setLocationRelativeTo(null);

         this.setVisible(true);



      }

      catch (IOException ex)

      {


      }

   }


   @Override

   public void paint(Graphics g)

   {

      super.paintComponents(g);

      g.drawString(statusLabel, StringX, StringY); // see
javadoc for more info on the parameters

   }



   public void setStatusLabel(String statusLabel)

   {

      this.statusLabel = statusLabel;
```

```
  }

}
```

…/src/sp/PredyketideWorkspaceVars.java

```
/**
 * @author John Althom A. Mendoza
 *
 * Class collectively representins the variables used in every biosynthesis.
 *
 * */

package sp;

import java.util.ArrayList;

public class PredyketideWorkspaceVars
{

  private ArrayList<PolyketideUnit> workingPolyketideUnits;
  private ArrayList<String> sequenceOfSMILES;
  private PolyketideUnit lastUnit;                    //important variable necessary when retrieving a saved work
  static final int LIMIT = 20;

  public PredyketideWorkspaceVars(ArrayList<PolyketideUnit> workingPolyketideUnits, ArrayList<String> sequenceOfSMILES)
  {
    this.workingPolyketideUnits = workingPolyketideUnits;
    this.sequenceOfSMILES = sequenceOfSMILES;
  }

  public PredyketideWorkspaceVars()
  {
    this.workingPolyketideUnits = new ArrayList<PolyketideUnit>();
    this.sequenceOfSMILES = new ArrayList<String>();
  }

  /**
   * @return the sequenceOfSMILES
   */
  public ArrayList<String> getSequenceOfSMILES()
  {
    return sequenceOfSMILES;
  }

  /**
   * @param sequenceOfSMILES the sequenceOfSMILES to set
   */
  public void addSequenceOfSMILES(String SMILES)
  {
    this.sequenceOfSMILES.add(SMILES);
  }

  /**
   * @return the workingPolyketideUnits
   */
  public ArrayList<PolyketideUnit> getWorkingPolyketideUnits()
  {
```

```
        return workingPolyketideUnits;

    }



    /**

     * @param workingPolyketideUnits the
workingPolyketideUnits to set

     */

    public void addWorkingPolyketideUnit(PolyketideUnit
workingPolyketideUnit)

    {


this.workingPolyketideUnits.add(workingPolyketideUnit);

    }



    /**

     * @param the last workingPolyketideUnits to be removed

     */

    public PolyketideUnit removeLast()

    {

        return
this.workingPolyketideUnits.remove(this.workingPolyketideU
nits.size()-1);

    }



    /**

     * @return Substring of the last working SMILES. This is
used

     *                         after importing a saved synthesis.

     * */

    public String getWorkingSMILES()

    {

        if(this.getSequenceOfSMILES().size() > 1)

        {

            String temp =
this.getSequenceOfSMILES().get(this.getSequenceOfSMILES()
.size()-1);


            return temp.substring(7);

        }

        else

        {

            return "";

        }

    }



    /**

     * @return the lastUnit

     */

    public PolyketideUnit getLastUnit()

    {

        return lastUnit;

    }



    /**

     * @param lastUnit the lastUnit to set

     */

    public void setLastUnit(PolyketideUnit lastUnit)

    {

        this.lastUnit = lastUnit;

    }



    public boolean isLimited()

    {



        return (this.workingPolyketideUnits.size() > LIMIT -1 );
```

```
    }

}
```

…/src/sp/ReferenceData.java

```
/**
 * @author John Althom A. Mendoza
 *
 * Class to retrieve the reference data. By default, the attached list -- synthetic.txt and
 *                         natprod.txt -- are processed as reference data.
 *
 * TODO Catch if may error ang SMILES
 *
 * */

package sp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;

import org.openscience.cdk.exception.InvalidSmilesException;
import org.openscience.cdk.interfaces.IAtomContainer;

@SuppressWarnings("serial")
public class ReferenceData extends ArrayList<IAtomContainer>
{

  private boolean isNaturalProduct;

  public ReferenceData(boolean isNaturalProduct)
  {
    this.isNaturalProduct = isNaturalProduct;
    if(this.isNaturalProduct)
      createReferenceDataObject("natprod.txt");
    else
      createReferenceDataObject("synthetic.txt");
  }

  private void createReferenceDataObject(String path)
  {

    try
    {

      InputStream in = getClass().getResourceAsStream("/resources/" + path);

      if(in == null)
      {
        in = getClass().getResourceAsStream("/"+path);
      }

      BufferedReader br = new BufferedReader(new InputStreamReader(in));
      String referenceSMILE;
      while ((referenceSMILE = br.readLine()) != null)
      {
```

```
        //TODO fix kung bakit may lumalabas na print 3

        IAtomContainer referenceMolecule = Predyketide.smpars.parseSmiles(referenceSMILE);


        this.add(referenceMolecule);

      }

    }

    catch (IOException e)

    {

      e.printStackTrace();

    }

    catch (InvalidSmilesException e)

    {

      e.printStackTrace();

    }

  }


}
```

…/src/sp/StarterUnitCollection.java

```
/**

 * @author John Althom A. Mendoza

 *

 * Class to retrieve the known starter units from the xml file included in the

 *                         application.

 *

 * */


package sp;


import java.io.InputStream;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;


import utils.XML_Utils;



public class StarterUnitCollection

{



  public StarterUnitCollection()

  {


    try

    {

      InputStream starterXML = StarterUnitCollection.class.getResourceAsStream("/starterUnits.xml");

      Document doc = XML_Utils.readXml(starterXML);

      doc.getDocumentElement().normalize();


      NodeList nList = doc.getElementsByTagName("aStarterUnit");


      for (int temp = 0; temp < nList.getLength(); temp++)

      {


        Node nNode = nList.item(temp);

        if (nNode.getNodeType() == Node.ELEMENT_NODE)
```

```
            {

               Element eElement = (Element) nNode;



                 Predyketide.starterUnits.add(new
PolyketideUnit(getTagValue("name", eElement),

                                      getTagValue("SMILES-forward",
eElement),

                                      getTagValue("image",
eElement)));



            }

          }

       }

       catch (Exception e)

       {

          e.printStackTrace();

       }

    }


    private static String getTagValue(String sTag, Element
eElement)

    {

       NodeList nlList =
eElement.getElementsByTagName(sTag).item(0).getChildNo
des();


       Node nValue = (Node) nlList.item(0);


       return nValue.getNodeValue();

    }

}
```

…/src/utils/FileManipulation.java

```
package utils;



import java.awt.Dimension;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.PrintWriter;

import java.io.Writer;

import java.util.ArrayList;

import java.util.NoSuchElementException;

import java.util.Scanner;

import java.util.zip.ZipEntry;

import java.util.zip.ZipOutputStream;


import javax.imageio.ImageIO;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.filechooser.FileFilter;

import javax.swing.filechooser.FileNameExtensionFilter;


import org.openscience.cdk.DefaultChemObjectBuilder;

import org.openscience.cdk.exception.CDKException;
```

```
import org.openscience.cdk.exception.InvalidSmilesException;

import org.openscience.cdk.interfaces.IAtomContainer;

import org.openscience.cdk.io.Mol2Writer;

import org.openscience.cdk.smiles.SmilesParser;


import com.ibm.icu.util.StringTokenizer;


import sp.PolyketideUnit;

import sp.PredyketideCompoundResult;

import sp.PredyketideWorkspaceVars;



public class FileManipulation

{


  static final String DELIMITER = "|";


  public static void exportSynthesisGUI(PredyketideWorkspaceVars workspaceVariables)

  {


    JFileChooser saveSynthesis = new JFileChooser();


    saveSynthesis.setDialogType(JFileChooser.SAVE_DIALOG);

    FileFilter filter = new FileNameExtensionFilter("pks files", "pksf");

    saveSynthesis.addChoosableFileFilter(filter);



    int ret = saveSynthesis.showDialog(null, "Export synthesis file");

    if (ret == JFileChooser.APPROVE_OPTION)

    {

      String filename = saveSynthesis.getSelectedFile().toString();

      exportSynthesis(workspaceVariables, filename);


      JOptionPane.showMessageDialog(null, "Successfully exported workspace as text file!");

    }

  }



  public static PredyketideWorkspaceVars importSynthesis()

  {

    JFileChooser openSynthesis = new JFileChooser();

    FileFilter filter = new FileNameExtensionFilter("pks files", "pksf");

    openSynthesis.addChoosableFileFilter(filter);


    int ret = openSynthesis.showDialog(null, "Import synthesis file");


    if (ret == JFileChooser.APPROVE_OPTION)

    {

      File file = openSynthesis.getSelectedFile();

      try

      {

        Scanner scanner = new Scanner(file);

        ArrayList<String> SMILES = new ArrayList<String>();

        ArrayList<PolyketideUnit> PKS = new ArrayList<PolyketideUnit>();


        String next = "";
```

```
            int counter = 0;



            String currUnitStringRep = scanner.nextLine();

            StringTokenizer tokenizer = new
StringTokenizer(currUnitStringRep, DELIMITER);


            int module =
Integer.parseInt(tokenizer.nextToken(DELIMITER));

            String unitName =
tokenizer.nextToken(DELIMITER).trim();

            String unitSMILES =
tokenizer.nextToken(DELIMITER).trim();


            PolyketideUnit lastUnit = new
PolyketideUnit(unitName, unitSMILES);

            lastUnit.setPKSModule(module);


            PolyketideUnit tempUnit = null;

            //start of loop reading

            while(scanner.hasNextLine() && !(next =
scanner.nextLine().trim()).isEmpty())

            {



               if(counter % 2 == 0)

               {

                  //System.out.println(next.trim());

                  SMILES.add(next.trim());

               }

               else

               {

                  tokenizer = new StringTokenizer(next,
DELIMITER);

                  module =
Integer.parseInt(tokenizer.nextToken(DELIMITER));

                  unitName =
tokenizer.nextToken(DELIMITER).trim();

                  unitSMILES =
tokenizer.nextToken(DELIMITER).trim();


                  tempUnit = new PolyketideUnit(unitName,
unitSMILES);

                  tempUnit.setPKSModule(module);

                  PKS.add(tempUnit);


                  //System.out.println("mod: " + module + "
uname: " + unitName + " SMILES: " + unitSMILES);

               }



               counter++;

            }

            scanner.close();


            PredyketideWorkspaceVars workspaceToReturn =
new PredyketideWorkspaceVars(PKS, SMILES);

            workspaceToReturn.setLastUnit(lastUnit);


            return workspaceToReturn;

         }

         catch (NoSuchElementException e)

         {

            JOptionPane.showMessageDialog(null, "The file can
not be read properly.\nIt can be corrupted or empty.");

            return null;

         }
```

```
            catch (FileNotFoundException e)
            {
                return null;
            }
            catch(@SuppressWarnings("hiding") IOException e)
            {
                return null;
            }

        }
        return null;
    }


    public static void exportImageGUI(JPanel
synthesisMainPanel, int WIDTH, int HEIGHT)
    {

        JFileChooser saveImage = new JFileChooser();
        saveImage.setDialogType(JFileChooser.SAVE_DIALOG);
        FileFilter filter = new FileNameExtensionFilter("JPEG
file", "jpg");
        saveImage.addChoosableFileFilter(filter);

        int ret = saveImage.showDialog(null, "Save workspace as
image");

        if (ret == JFileChooser.APPROVE_OPTION)
        {
            String filename =
saveImage.getSelectedFile().toString();
            if(filename.indexOf(".jpg") < 0)
            {
                filename += ".jpg";
            }

            exportImage(synthesisMainPanel, WIDTH, HEIGHT,
filename);

            JOptionPane.showMessageDialog(null, "Successfully
exported workspace as image!");
        }
    }

    public static File exportImage(JPanel synthesisMainPanel,
int WIDTH, int HEIGHT, String filename)
    {


        WIDTH += 150; //TODO fix bakit kapos

        File file = new File(filename);

        Dimension size = new Dimension(WIDTH, HEIGHT);
        BufferedImage image =
(BufferedImage)synthesisMainPanel.createImage(size.width,
size.height);
        Graphics g = image.getGraphics();
        synthesisMainPanel.paint(g);
        g.dispose();
        try
        {
            ImageIO.write(image, "jpg", file);
        }
        catch (IOException e)
        {
```

```
      e.printStackTrace();
    }


    return file;


  }



  public static ArrayList<IAtomContainer> importReferenceData()
  {
    JFileChooser openSynthesis = new JFileChooser();
    FileFilter filter = new FileNameExtensionFilter("Reference text collection (.txt)", "txt");
    openSynthesis.addChoosableFileFilter(filter);

    int ret = openSynthesis.showDialog(null, "Import synthesis file");

    if (ret == JFileChooser.APPROVE_OPTION)
    {
      File file = openSynthesis.getSelectedFile();

      ArrayList<IAtomContainer> newRefData = new ArrayList<IAtomContainer>();

      try
      {
        Scanner refDataReader = new Scanner(file);
        SmilesParser smparser = new SmilesParser(DefaultChemObjectBuilder.getInstance());
        while(refDataReader.hasNext())
        {
          String SMILES = refDataReader.nextLine().trim();
          newRefData.add(smparser.parseSmiles(SMILES));
        }
        JOptionPane.showMessageDialog(null, "Succesfully imported new data.");
        return newRefData;
      }
      catch (FileNotFoundException e)
      {
        e.printStackTrace();

        JOptionPane.showMessageDialog(null, "Import new data failed.");
        return null;
      }
      catch(@SuppressWarnings("hiding") IOException e)
      {
        JOptionPane.showMessageDialog(null, "Import new data failed.");
        return null;
      }
      catch (InvalidSmilesException e)
      {
        JOptionPane.showMessageDialog(null, "Import new data failed.");
        return null;
      }
    }
    return null;
  }
```

```
    public static File
exportSynthesis(PredyketideWorkspaceVars
workspaceVariables, String filename)

    {


        PolyketideUnit lastUnit =
workspaceVariables.getLastUnit();


        ArrayList<String> SMILES =
workspaceVariables.getSequenceOfSMILES();

        ArrayList<PolyketideUnit> PKSUnits =
workspaceVariables.getWorkingPolyketideUnits();


        if(filename.indexOf(".pksf") < 0)

        {

            filename += ".pksf";

        }


        File file = new File(filename);

        try

        {

            PrintWriter pw = new PrintWriter(file);


            /**

             * Writes as the first line, the last unit of the synthesis.

             * */

            pw.println(lastUnit.getPKSModule() + DELIMITER +
lastUnit.getUnitName() + DELIMITER +
lastUnit.getForwardSMILES());


            for(int j = 0; j < SMILES.size(); j++)

            {


                pw.println(SMILES.get(j));

                if(j < PKSUnits.size())

                {

                    PolyketideUnit tempUnit = PKSUnits.get(j);

                    pw.println(tempUnit.getPKSModule() +
DELIMITER + tempUnit.getUnitName() + DELIMITER
+tempUnit.getForwardSMILES());

                }

            }


            pw.close();

            pw.flush();

            return file;

        }

        catch (FileNotFoundException e)

        {

            e.printStackTrace();

        }

        catch(@SuppressWarnings("hiding") IOException e)

        {

            e.printStackTrace();

        }

        return file;


    }


    public static void
exportPredictionAsZip(PredyketideCompoundResult
prediction,

            PredyketideWorkspaceVars workspace, JPanel
synthesisMainPanel, int WIDTH, int HEIGHT)

    {
```

```
    JFileChooser savePrediction = new JFileChooser();

    savePrediction.setDialogType(JFileChooser.SAVE_DIALOG);

    FileFilter filter = new FileNameExtensionFilter("Polyketide zip file", "pkzip");

    savePrediction.addChoosableFileFilter(filter);


    int ret = savePrediction.showDialog(null, "Save this prediction");


    if (ret == JFileChooser.APPROVE_OPTION)

    {

      String filename = savePrediction.getSelectedFile().toString();

      String fileNameTemplate = "prediction";

      if(filename.indexOf(".pk.zip") < 0)

      {

        filename += ".pk.zip";

      }



      byte[] buffer = new byte[1024];


      /**

       * 1. Synthesis panel pksf file.

       * 2. Synthesis panel as image.

       * 3. SMILES representation of prediction.

       * 4. MolFile representation of prediction.

       * 5. Descriptors of the prediction

       * 6. Image of the prediction.

       * */

      File[] sourceFiles = new File[6];

      sourceFiles[0] = exportSynthesis(workspace, fileNameTemplate+"_synthesis.pksf");

      sourceFiles[1] = exportImage(synthesisMainPanel, WIDTH, HEIGHT, fileNameTemplate+"_synthesis.jpg");

      sourceFiles[2] = exportAsSMILES(prediction.getResultSMILES(), fileNameTemplate+".smi");

      sourceFiles[3] = exportAsMolFile(prediction.getPredictedMolecule(), fileNameTemplate+".mol");

      try

      {

        sourceFiles[4] = exportPredictionDescriptor(prediction.getDescriptors(), fileNameTemplate+"_descr.txt");

      }

      catch (CDKException e1)

      {

        e1.printStackTrace();

      }

      sourceFiles[5] = exportPredictionAsImage(prediction.getImage(), fileNameTemplate+".jpg");


      try

      {

        FileOutputStream fout = new FileOutputStream(filename);

        ZipOutputStream zout = new ZipOutputStream(fout);


        for(int l = 0; l < sourceFiles.length; l++)

        {

          FileInputStream fin = new FileInputStream(sourceFiles[l]);

          zout.putNextEntry(new ZipEntry(sourceFiles[l].getName()));

          int length;
```

```
                while((length = fin.read(buffer)) > 0)
                {
                   zout.write(buffer, 0, length);
                }

                /*
                 * After writing the file to ZipOutputStream, use
                 *
                 * void closeEntry() method of ZipOutputStream
class to
                 * close the current entry and position the stream
to
                 * write the next entry.
                 */

                zout.closeEntry();

                //close the InputStream
                fin.close();
                sourceFiles[l].delete();
             }

             zout.close();
             zout.flush();
          }
          catch (FileNotFoundException e)
          {
             e.printStackTrace();
          }
          catch (IOException e)
          {
             e.printStackTrace();
          }

          JOptionPane.showMessageDialog(null, "Successfully
exported prediction!");
       }

    }



    private static File exportPredictionAsImage(Image image,
String filename)
    {

       File file = new File(filename);
       try
       {
          BufferedImage bi = new BufferedImage

(image.getWidth(null),image.getHeight(null),BufferedImage.
TYPE_INT_RGB);
          Graphics bg = bi.getGraphics();
          bg.drawImage(image, 0, 0, null);
          bg.dispose();
          ImageIO.write(bi, "jpg", file);
       }
       catch (IOException e)
       {
          e.printStackTrace();
       }

       return file;
    }
```

```
  private static File exportPredictionDescriptor(String
descriptors, String file)

  {

    File temp = new File(file);

    try

    {

      PrintWriter pw = new PrintWriter(temp);

      pw.println(descriptors);

      pw.close();

      pw.flush();

    }

    catch(Exception e)

    {


    }

    return temp;

  }




  private static File exportAsMolFile(IAtomContainer
predictedMolecule, String file)

  {

    File temp = new File(file);

    try

    {

      Writer pw = new PrintWriter(temp);

      Mol2Writer molwriter = new Mol2Writer(pw);

      molwriter.writeMolecule(predictedMolecule);

      molwriter.close();

      pw.close();

      pw.flush();

    }

    catch(Exception e)

    {


    }

    return temp;

  }




  private static File exportAsSMILES(String resultSMILES,
String file)

  {

    File temp = new File(file);

    try

    {

      PrintWriter pw = new PrintWriter(temp);

      pw.println(resultSMILES);

      pw.close();

      pw.flush();

    }

    catch(Exception e)

    {


    }

    return temp;

  }


}
```

…/src/utils/ResultRankingByNPScore.java

```
package utils;


import java.util.Comparator;


import sp.PredyketideCompoundResult;


public class ResultRankingByNPScore implements Comparator<PredyketideCompoundResult>{


	@Override
	public int compare(PredyketideCompoundResult arg0, PredyketideCompoundResult arg1) {
		return (arg0.getNP_LikenessScore() >arg1.getNP_LikenessScore() ? -1 : (arg0.getNP_LikenessScore() == arg1.getNP_LikenessScore() ? 0 : 1));
	}
}
```

…/src/utils/XML_Utils.java

```
package utils;
/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */


import java.io.IOException;
import java.io.InputStream;
import java.io.StringReader;


import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;


import org.w3c.dom.Document;
import org.xml.sax.EntityResolver;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;


/**
 * Few simple utils to read DOM. This is originally from the Jakarta Commons
 * Modeler.
 *
 * @author Costin Manolache
 */
public class XML_Utils {
  /**
   * Read XML as DOM.
```

```
  */

 public static Document readXml(InputStream is) throws
SAXException, IOException,

    ParserConfigurationException {

    DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();


    dbf.setValidating(false);

    dbf.setIgnoringComments(false);

    dbf.setIgnoringElementContentWhitespace(true);

    dbf.setNamespaceAware(true);

    // dbf.setCoalescing(true);

    // dbf.setExpandEntityReferences(true);


    DocumentBuilder db = null;

    db = dbf.newDocumentBuilder();

    db.setEntityResolver(new NullResolver());


    // db.setErrorHandler( new MyErrorHandler());


    return db.parse(is);

  }

}


class NullResolver implements EntityResolver {

 public InputSource resolveEntity(String publicId, String
systemId) throws SAXException,

    IOException {

   return new InputSource(new StringReader(""));

  }

}
```

B. Source Codes for Polykase

BASE PATH = */polykase/application/*

NOTE: *This component uses CodeIgniter v2.2.1. All files other than the ones mentioned below are NOT modified and not included in this section.*

…/controllers/administrator_tasks.php

```
<?php/**

* @author ajmendoza

* @copyright 2013

*/class Administrator_tasks extends CI_Controller{public
function __construct()

{

parent::__construct();

if(!isset($_SESSION))

session_start();

}public function index(){

}public function pending_list(){$data['title'] = 'Pending Users
Lists';$this->load->view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');
```

```
$this->load->view('templates/menu');if
(isset($_SESSION['username']) && $_SESSION['username'] ==
"admin") {$this->load->model('user_model');

$pendingUsersCount = $this

->user_model

->get_unauthUserListCount();$pendingUsers = $this

->user_model

->get_unauthUserList();$data['resultsTotal'] =
$pendingUsersCount;

$data['pendingUsers'] = $pendingUsers;

$this->load->view('admin/pendingUsersPage', $data);}$this-
>load->view('templates/footer');}public function
approve_user($id){

if(!isset($_SESSION['username']) || $_SESSION['username']
!= "admin"){print header("refresh:2;url=".site_url()."/users");

print "You don't do that to me.";

return;

}

$updates = array('approved' => 1);

$this->db->where('user_id', $id);

$this->db->update('users', $updates);$this->load-
>model("user_model");$thisUser = (array)$this-
>user_model->get_userID($id);

$email = $thisUser['email'];$this->load->library('email');$this-
>email->from('no-reply@polykase.com', 'Polykase
Administrator');

$this->email->to($email); $this->email->subject('Account
Approval');

$this->email->message("Your account has already been
approved. Kindly login <a
href=".site_url("users").">here</a>");
                                                          $t
his->email->send();redirect("admin/pending", "refresh");

}public function delete_user($id){

if(isset($_SESSION['username']) && $_SESSION['username']
== "admin"){print
header("refresh:2;url=".site_url()."/admin/pending");

print "Deleting user...";

$this->db->where('user_id', $id);

$this->db->delete('users');

}else{print header("refresh:2;url=".site_url()."/users");

print "You don't do that to me.";

}}

public function moderatepredictions(){$data['title'] =
'Moderate Predictions';$this->load-
>view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');

$this->load->view('templates/menu');if
(isset($_SESSION['username']) && $_SESSION['username'] ==
"admin") {$this->load->model('prediction_model');

$predictionCount = $this

->prediction_model

->get_AllPredictionsCount();$predictions = $this

->prediction_model

->get_AllPredictions();$data['resultsTotal'] =
$predictionCount;

$data['predictions'] = $predictions;

$this->load->view('admin/moderatepreds', $data);}$this-
>load->view('templates/footer');}

public function moderatecomments(){$data['title'] =
'Moderate Comments';$this->load-
>view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');

$this->load->view('templates/menu');if
(isset($_SESSION['username']) && $_SESSION['username'] ==
"admin") {$this->load->model('prediction_model');

$predictionCount = $this

->prediction_model

->get_AllCommentsCount();$comments = $this

->prediction_model

->get_AllComments();$data['resultsTotal'] =
$predictionCount;
```

```
$data['comments'] = $comments;

$this->load->view('admin/moderatecomments',
$data);}$this->load->view('templates/footer');}

public function deletecomment($id){

$this->db->where('comment_id', $id);

$this->db->delete('comment');
redirect("moderatecomments", "refresh");

}

public function deletepred($author, $id, $name){$this->load-
>model('user_model');

$user = $this->user_model->get_userID($author);

$user = (array)$user;

$user = $user['uname'];

if(isset($_SESSION['username']) && ($user ==
$_SESSION['username'] || $_SESSION['username'] ==
"admin")){

if($_SESSION['username'] == "admin"){

print
header("refresh:2;url=".site_url()."/moderatepredictions");

}else{

print
header("refresh:2;url=".site_url()."/reguser/viewpredictionsb
y/$author/$user");

}

print "Deleting record...";

$this->load->helper('file');

delete_files('./predictions/'.$author.'/'.$id.'/', TRUE);

unlink('./predictions/'.$author.'/'.$name.'.zip');

rmdir('./predictions/'.$author.'/'.$id.'/');

$this->db->where('pred_id', $id);

$this->db->delete('prediction');$this->db-
>where('related_prediction', $id);

$this->db->delete('comment');

}else{

if(isset($_SESSION['username'])){

print header("refresh:2;url=".site_url()."/user_profile");

}else{

print header("refresh:2;url=/polykase/");

}

print "You are trying to delete a record you do not own.
Bad!";

}}public function moderatepolyketide(){

$data['title'] = 'Moderate polyketide records';$this->load-
>view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');

$this->load->view('templates/menu');if
(isset($_SESSION['username']) && $_SESSION['username'] ==
"admin") {$this->load->database();

$this->db->from('pk_natprod');

$results = $this->db->get();$pknpRecords = array();

if($results->num_rows() > 0)

foreach($results->result() as $row)

$pknpRecords[] = $row;

$data['pknpRecords'] = $pknpRecords;

$this->load->view('admin/allPKNP', $data);}$this->load-
>view('templates/footer');

}

public function moderatepks(){

$data['title'] = 'Moderate polyketide synthase records';$this-
>load->view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');

$this->load->view('templates/menu');if
(isset($_SESSION['username']) && $_SESSION['username'] ==
"admin") {$this->load->database();

$this->db->from('pk_synthase');

$this->db->where('natprod_name');

$results = $this->db->get();$pknpRecords = array();
```

```
if($results->num_rows() > 0)

foreach($results->result() as $row)

$pknpRecords[] = $row;

$data['pknpRecords'] = $pknpRecords;

$this->load->view('admin/remPKS', $data);}$this->load->view('templates/footer');

}public function deletepks($id){

if(!isset($_SESSION['username']) || $_SESSION['username'] != "admin"){print header("refresh:2;url=".site_url()."/users");

print "You don't do that to me.";

return;

}$this->db->where('PKS_id', $id);

$x = $this->db->get('pk_synthase');

$organism = (array)$x->row(0);

$organismID = $organism['source_organism_id'];$this->db->where('PKS_id', $id);

$this->db->delete('pk_synthase');$this->db->where('involved_synthase', $id);

$this->db->delete('pks_module'); $this->db->where('organism_id', $organismID);

$this->db->delete('organism');redirect("moderatepks", "refresh");

}

public function deletepolyketide($id){

if(!isset($_SESSION['username']) || $_SESSION['username'] != "admin"){print header("refresh:2;url=".site_url()."/users");

print "You don't do that to me.";

return;

}

$this->db->where('natprod_name', $id);

$x = $this->db->get('pk_natprod');

$organism = (array)$x->row(0);

$organismID = $organism['host_organism_id'];$this->db->where('natprod_name', $id);

$this->db->delete('pk_natprod');

$updates = array('natprod_name' => NULL);

$this->db->where('natprod_name', $id);

$this->db->update('pk_synthase', $updates); $this->db->where('organism_id', $organismID);

$this->db->delete('organism');redirect("moderatepolyketide", "refresh");

}  public function editpolyketide($id){

$data['id'] = $id;if (isset($_SESSION['username']) && $_SESSION['username'] == "admin")

{$uploadPath = './polyketide_images/';

if(!is_dir($uploadPath))

{

mkdir($uploadPath,0777,TRUE);

}

$config['upload_path'] = $uploadPath;

$config['allowed_types'] = 'jpg|gif|png';$this->load->library('upload', $config);

$this->load->library('form_validation');$this->form_validation->set_rules('pknpName', 'PKNP Name', 'required');

$this->form_validation->set_rules('host_organism', 'Host Organism', 'required');

$this->form_validation->set_rules('pkssubmitted', 'PKS Order', 'required');//GETTING ALL MODULE UNITS

$this->load->database();

$this->db->select('*');

$this->db->from('pk_synthase');

$data['pksRecords'] = $this->db->where('natprod_name')->get();

$this->db->select('*');

$this->db->from('pk_synthase');

$data['pksRecordsInNatProd'] = $this->db->where('natprod_name', $id)->get();$this->db->select('*');

$this->db->from('pk_natprod');

$natprodDB = $this->db->where('natprod_name', $id)->get();
```

```
$natprodArray = (array)$natprodDB->row();

$data['pknpName'] = $natprodArray['natprod_name'];

$data['descr'] = $natprodArray['NP_desc'];

$organismID = $natprodArray['host_organism_id'];

$this->db->from('organism');

$organismDB = $this->db->where('organism_id', $organismID)->get();

$organismArray = (array)$organismDB->row();

$data['host_organism'] = $organismArray['organism_specie'];if ($this->form_validation->run() == FALSE || !$this->upload->do_upload())

{$this->load->database();$data['title'] = 'Edit '.$id;

$this->load->view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon', 'image/ico');

$this->load->view('templates/menu');$data = array_merge($data, array('error' => $this->upload->display_errors()));

$this->load->view('pages/PKNPEditForm', $data);$this->load->view('templates/footer');

}

else

{$this->load->database();//INSERT TO ORGANISM TABLE

$organism = array();

$organism['organism_specie'] = $_POST['host_organism'];

$organism['organism_type'] = 'host';

$this->db->where('organism_id', $organismID)->update('organism', $organism);

//UPDATE TO PKNP TABLE

$natprod = array();

$natprod['natprod_name'] = $_POST['pknpName'];

$natprod['host_organism_id'] = $organismID;

$natprod['PKS_Type'] = 1;

$temp = $this->upload->data();

$natprod['pknpImage'] = $temp['file_name'];

$natprod['NP_desc'] = $_POST['descr'];

$this->db->where('natprod_name', $id)->update('pk_natprod', $natprod);

//CLEAR CONTENTS OF PREVIOUS PKS INVOLVED

$update = array('natprod_name' => NULL);

$this->db->where('natprod_name', $id)->update('pk_synthase', $update);//UPDATE PKS_MODULE TABLE

$update = array();

$update['natprod_name'] = $_POST['pknpName'];

$submittedPKSString = $_POST['pkssubmitted'];

$tok = strtok($submittedPKSString, "_");

for($i = 0; $tok !== false; $i++)

{

$update['pks_order'] = ($i+1);

$this->db->where('PKS_id', intval($tok))->update('pk_synthase', $update);

$tok = strtok("_");

}$this->db->close();redirect("moderatepolyketide", "refresh");

}}

}

}

?>
```

<u>…/controllers/contactform.php</u>

```
<?php/**

* @author ajmendoza

* @copyright 2012

*/class Contactform extends CI_Controller {public function __construct()

{
```

```
parent::__construct();

if(!isset($_SESSION))

session_start();

}

function index()

{$page = 'Contact';

$data['title'] = ucfirst($page); // Capitalize the first
letter$this->load->view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');

$this->load->view('templates/menu', $data);$this->load-
>library('form_validation');

$this->load->helper('recaptchalib');

// Validation rules.$this->form_validation->set_rules('name',
'Name', 'required');

$this->form_validation->set_rules('email', 'Email',
'required|valid_email');

$this->form_validation->set_rules('message', 'Message',
'required');

$this->form_validation-
>set_rules('recaptcha_challenge_field', 'Security Question',
'required|callback_val_recaptcha');if(isset($_SESSION['usern
ame'])){

$this->load->model("user_model");

$data['loggedUser'] = $this->user_model-
>get_user($_SESSION['username']);

$data['loggedUser'] = (array)$data['loggedUser'];

}

if ($this->form_validation->run() == FALSE)

{

$this->load->view('pages/contact', $data);

}

else

{

$this->load->library('email');$this->email-
>from(set_value('email'), set_value('name'));

$this->email->to('200942070@post.upm.edu.ph'); $this-
>email->subject('Polykase Suggestion');

$this->email->message(set_value('message'));
                                                        $t
his->email->send();//echo $this->email->print_debugger();

$this->load->view('pages/contactsuccess');

}$this->load->view('templates/footer', $data);

}function val_recaptcha($string)

{

$resp = recaptcha_check_answer($this->config-
>item('recap_private'),

$_SERVER["REMOTE_ADDR"],

$this->input->post("recaptcha_challenge_field"),

$this->input->post("recaptcha_response_field"));if(!$resp-
>is_valid) {

$this->form_validation->set_message('val_recaptcha','Your
answer for the security question was incorrect, please try
again.');

return FALSE;

}

else {

return TRUE;

}

}

}

?>
```

…/controllers/create_user.php

```
<?php/**

* @author ajmendoza

* @copyright 2012

*/class Create_user extends CI_Controller{public function
__construct()
```

```
{
parent::__construct();
if(!isset($_SESSION))
session_start();
}public function index(){$data['title'] = 'Create Account';$this->load->view('templates/header', $data);
echo link_tag('css/style.css');
echo link_tag('images/favicon.jpg', 'shortcut icon', 'image/ico');
$this->load->view('templates/menu');$this->load->library('form_validation');
$this->load->helper('recaptchalib');
// Validation rules.$this->form_validation->set_rules('uname', 'Username', 'required|min_length[5]|max_length[12]|is_unique[users.uname]');
$this->form_validation->set_rules('password', 'Password', 'required|matches[password2]');
$this->form_validation->set_rules('password2', 'Confirm Password', 'required');
$this->form_validation->set_rules('email', 'Email', 'required|valid_email|is_unique[users.email]');$this->form_validation->set_rules('fname', 'First Name', 'required');
$this->form_validation->set_rules('surname', 'Surname', 'required');
$this->form_validation->set_rules('occupation', 'Occupation', 'required');$this->form_validation->set_rules('short_bio', 'Short Bio');
$this->form_validation->set_rules('home_address', 'Home Address');
$this->form_validation->set_rules('contact_num', 'Contact Number');
$this->form_validation->set_rules('recaptcha_challenge_field', 'Security Question', 'required|callback_val_recaptcha');if ($this->form_validation->run() == FALSE)
{
$this->load->view('pages/create_user_form', $data);
}
else
{
$this->load->database();
$data = $_POST;
$approved = array('approved' => 0);
$data["password"] = sha1($data["password"]);
unset($data["password2"]);
unset($data["recaptcha_challenge_field"]);
unset($data["recaptcha_response_field"]);
$data = array_merge($data, $approved);$this->db->insert('users', $data);$this->load->library('email');$this->email->from('no-reply@polykase.com', 'Polykase Administrator');
$this->email->to($data['email']); $this->email->subject('Account Creation');
$this->email->message("Your account has already been created. Kindly wait for administrator's review.");
$this->email->send();$data['notifType'] = "create-user-pending";
$data['title'] = "Waiting for approval";
$this->load->view('pages/notifications', $data);$this->db->close();
}$this->load->view('templates/footer');}function val_recaptcha($string)
{
$resp = recaptcha_check_answer($this->config->item('recap_private'),
$_SERVER["REMOTE_ADDR"],
$this->input->post("recaptcha_challenge_field"),
$this->input->post("recaptcha_response_field"));if(!$resp->is_valid) {
$this->form_validation->set_message('val_recaptcha','Your answer for the security question was incorrect, please try again.');
return FALSE;
}
else {
```

```
return TRUE;

}

}}?>
```

…/controllers/pages.php

```
<?phpclass Pages extends CI_Controller {public function
view($page = 'home')
{if ( ! file_exists('application/views/pages/'.$page.'.php'))
{
// Whoops, we don't have a page for that!
show_404();
}$data['title'] = ucfirst($page); // Capitalize the first
letter$this->load->view('templates/header', $data);
$this->load->view('pages/'.$page, $data);
$this->load->view('templates/footer', $data);}
}
```

…/controllers/polykase.php

```
<?php/**
* @author ajmendoza
* @copyright 2012
*/class Polykase extends CI_Controller{public function
__construct()
{
parent::__construct();
if(!isset($_SESSION))
session_start();
}public function view($page = 'users')
{
if ( isset($_SESSION['username']) && $page=='users' ) {
redirect('user_profile');
}   if ( ! file_exists('application/views/pages/'.$page.'.php'))
{
// Whoops, we don't have a page for that!
show_404();
}$this->load->helper('html');
$this->load->helper('url');$data['title'] = ucfirst($page); //
Capitalize the first letter//modification of page title
switch($data['title']){
case "Dbase":
$data['title'] = "Database";
break;
case "User_profile":
$data['title'] = "User Profile";
break;
}
//end of page title modification$this->load-
>view('templates/header', $data);
echo link_tag('css/style.css');
echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');
$this->load->view('templates/menu', $data);
$this->load->view('pages/'.$page, $data);
$this->load->view('templates/footer', $data);}}
?>
```

…/controllers/polyketides.php

```
<?php/**
* @author ajmendoza
* @copyright 2012
*/class Polyketides extends CI_Controller
{

public function __construct()
{
parent::__construct();

}

public function index()
{

}

public function addpksrecord()
{



$data['title'] = 'Add New PKS';



$this->load->view('templates/header', $data);
echo link_tag('css/style.css');
echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');
$this->load->view('templates/menu');



if (isset($_SESSION['username']) && $_SESSION['username']
== "admin")
{





$this->load->library('form_validation');





$this->form_validation->set_rules('PKS_name', 'PKS Name',
'required|is_unique[pk_synthase.PKS_name]');
$this->form_validation->set_rules('source_gene', 'Source
gene', 'required|is_unique[organism.gene]');
$this->form_validation->set_rules('resources', 'Supporting
resources', '');
```

```
if ($this->form_validation->run() == FALSE)
{
//GETTING ALL MODULE UNITS
$this->load->database();
$data['modUnits'] = $this->db->order_by("module_type",
"asc")->get('module_unit');
$this->load->view('pages/createpks', $data);
}
else
{

$this->load->database();

//INSERT TO ORGANISM TABLE
$organism = array();
$organism['gene'] = $_POST['source_gene'];
$organism['organism_type'] = 'source';
$this->db->insert('organism', $organism);

//RETRIEVE ID OF ORGANISM
$results =  $this->db->where('gene', $_POST['source_gene'])-
>get('organism');
$x = (array)$results->row();
$organismID = $x['organism_id'];
//INSERT INTO PKS TABLE
$pksRecord = array();
$pksRecord['PKS_name'] = $_POST['PKS_name'];
$pksRecord['source_organism_id'] = $organismID;
if(isset($_POST['resources']))
$pksRecord['resources'] = $_POST['resources'];
$this->db->insert('pk_synthase', $pksRecord);
```

```
//RETRIEVE PKS ID
$pksIDsearch = $this->db->where('PKS_name',
$_POST['PKS_name'])->get('pk_synthase');
$x = (array)$pksIDsearch->row();
$pksID = $x['PKS_id'];

//INSERT INTO PKS_MODULE TABLE
$pksModRecord = array();
$pksModRecord['involved_synthase'] = $pksID;
$submittedPKSMods = $_POST['pksunits'];
$submittedDomainSeq = $_POST['domainseq'];
for($i = 0; $i < sizeof($submittedPKSMods); $i++)
{
$pksModRecord['module_unit'] = $submittedPKSMods[$i];
$pksModRecord['domain_seq'] = $submittedDomainSeq[$i];
if(empty($submittedDomainSeq[$i]))
{
$pksModRecord['domain_seq'] = "Starter";
}
$pksModRecord['module_order'] = ($i+1);
$this->db->insert('pks_module', $pksModRecord);
}

$data['notifType'] = "pks-add-success";
$data['title'] = "New PKS record added";
$this->load->view('pages/notifications', $data);

$this->db->close();
}
}

$this->load->view('templates/footer');

}
public function addpknprecord()
{

$data['title'] = 'Add New PK Natural Product';
```

```
$this->load->view('templates/header', $data);
echo link_tag('css/style.css');
echo link_tag('images/favicon.jpg', 'shortcut icon', 'image/ico');
$this->load->view('templates/menu');

if (isset($_SESSION['username']) && $_SESSION['username'] == "admin")
{

$this->load->library('form_validation');

$uploadPath = './polyketide_images/';
if(!is_dir($uploadPath))
{
mkdir($uploadPath,0777,TRUE);
}
$config['upload_path'] = $uploadPath;
$config['allowed_types'] = 'jpg|gif|png';

$this->load->library('upload', $config);

$this->form_validation->set_rules('pknpName', 'PKNP Name', 'required|is_unique[pk_natprod.natprod_name]');
$this->form_validation->set_rules('host_organism', 'Host Organism', 'required|is_unique[organism.organism_specie]');
$this->form_validation->set_rules('pkssubmitted', 'PKS Order', 'required');

if ($this->form_validation->run() == FALSE || !$this->upload->do_upload())
{
//GETTING ALL MODULE UNITS
$this->load->database();
$this->db->select('*');
$this->db->from('pk_synthase');
$data['pksRecords'] = $this->db->where('natprod_name')->get();
$data = array_merge($data, array('error' => $this->upload->display_errors()));
```

```
$this->load->view('pages/PKNPForm', $data);
}
else
{

$this->load->database();

//INSERT TO ORGANISM TABLE
$organism = array();
$organism['organism_specie'] = $_POST['host_organism'];
$organism['organism_type'] = 'host';
$this->db->insert('organism', $organism);
//RETRIEVE ID OF ORGANISM
$results =  $this->db->where('organism_specie', $_POST['host_organism'])->get('organism');
$x = (array)$results->row();
$organismID = $x['organism_id'];

//INSERT TO PKNP TABLE
$natprod = array();
$natprod['natprod_name'] = $_POST['pknpName'];
$natprod['host_organism_id'] = $organismID;
$natprod['PKS_Type'] = 1;
$temp = $this->upload->data();
$natprod['pknpImage'] = $temp['file_name'];
$natprod['NP_desc'] = $_POST['descr'];
$this->db->insert('pk_natprod', $natprod);
//UPDATE PKS_MODULE TABLE
$update = array();
$update['natprod_name'] = $_POST['pknpName'];
$submittedPKSString = $_POST['pkssubmitted'];
$tok = strtok($submittedPKSString, "_");
for($i = 0; $tok !== false; $i++)
{
$update['pks_order'] = ($i+1);
$this->db->where('PKS_id', intval($tok))->update('pk_synthase', $update);
$tok = strtok("_");
}
```

```
$data['notifType'] = "pknp-add-success";
$data['title'] = "New PKNP record added";
$this->load->view('pages/notifications', $data);




$this->db->close();
}
}


$this->load->view('templates/footer');

}
public function viewallrecords()
{


$data['title'] = 'Record of Type I PK Natural Products';



$this->load->view('templates/header', $data);
echo link_tag('css/style.css');
echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');
$this->load->view('templates/menu');
$this->load->database();
$data['pkdbRecords'] = $this->db-
>order_by("natprod_name", "asc")->get('pk_natprod');



$this->db->close();
$this->load->view('pages/dbase', $data);
$this->load->view('templates/footer');

}

public function viewPKNP($name)
{


$data['title'] = $name;



$this->load->view('templates/header', $data);
echo link_tag('css/style.css');
echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');
$this->load->view('templates/menu');
$this->load->database();
$this->db->from('pks_module');
$this->db->join('pk_synthase', 'pk_synthase.PKS_id =
pks_module.involved_synthase');
$this->db->join('module_unit', 'module_unit.module_id =
pks_module.module_unit');
$this->db->join('organism as gene',
'pk_synthase.source_organism_id = gene.organism_id');
$this->db->order_by("pks_order", "asc");
$this->db->where('pk_synthase.natprod_name', $name);
$results = $this->db->get();

$pknpData = array();
if($results->num_rows() > 0)
foreach($results->result() as $row)
$pknpData[] = $row;

$data['pknpRecord'] = $pknpData;

$this->db->from('pk_natprod');
$this->db->join('organism', 'pk_natprod.host_organism_id =
organism.organism_id');
$this->db->where('natprod_name', $name);
$xx = $this->db->get();


$data['basicInfo'] = (array)$xx->row();

$this->db->close();
$this->load->view('pages/PKNPTemplate', $data);
$this->load->view('templates/footer');

}}
?>
```

…/controllers/reguser_tasks.php

```
<?php/** * @author ajmendoza * @copyright 2013 */class
Reguser_tasks extends CI_Controller{



public function __construct()

{



parent::__construct();



if(!isset($_SESSION))

session_start();
```

```
}


public function index(){

}


public function viewpred_form($user_id){


$data['title'] = 'Upload New Prediction';


$data['user_id'] = $user_id;


$this->load->view('templates/header', $data);


echo link_tag('css/style.css');


echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');
                                                     $t
his->load->view('templates/menu');


if (isset($_SESSION['username'])) {

$this->load->library('form_validation');

$this->form_validation->set_rules('predname', 'Prediction\'s
Name',
'required|min_length[5]|alpha_numeric|is_unique[predictio
n.pred_name]');

$uploadPath = './predictions/'.$user_id."/";

if(!is_dir($uploadPath))

{

mkdir($uploadPath,0777,TRUE);

}

$config['upload_path'] = $uploadPath;

$config['allowed_types'] = 'zip';

if(isset($_POST['predname']))

$config['file_name'] = $_POST['predname'].".zip";

$this->load->library('upload', $config);

if ($this->form_validation->run() == FALSE || !$this->upload-
>do_upload())

{


$data = array_merge($data, array('error' => $this->upload-
>display_errors()));


$this->load->view('regular/pred_form', $data);

}else{


//store to database


$this->load->database();


$predictionEntry = array();


$predictionEntry['pred_name'] = $_POST['predname'];


$predictionEntry['user_id'] = $user_id;


$this->db->insert('prediction', $predictionEntry);


//get prediction_id


$this->load->model('prediction_model');


$predID = $this->prediction_model->getID($predictionEntry);


$this->load->library('unzip');
```

```
if(!is_dir($uploadPath.$predID.'/')){

mkdir($uploadPath.$predID.'/', 0777, TRUE);


}


$this->unzip->extract($uploadPath.$config['file_name'],
$uploadPath.$predID.'/');


$filePath = $uploadPath.$predID.'/';


$viewData = $this->readPredictionFiles($filePath,
$_POST['predname']);


$this->load->view('pages/predictionTemplateNew',
$viewData);

}


}else{

$data['notifType'] = 'invalid-access';


$this->load->view('pages/notifications', $data);


}


$this->load->view('templates/footer');

}

public function viewprediction($user_id, $pred_id,
$predName){


$data['title'] = $predName;


$this->load->view('templates/header', $data);
echo link_tag('css/style.css');


echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');
                                                  $t
his->load->view('templates/menu');


if (isset($_SESSION['username'])) {

$filePath = "./predictions/".$user_id."/".$pred_id."/";

$viewData = $this->readPredictionFiles($filePath,
$predName);

$this->load->model('prediction_model');

$viewData['author'] = $this->prediction_model-
>getAuthor($user_id);

$viewData['comments'] = $this->prediction_model-
>getComments($pred_id);

$viewData['commentsCount'] = $this->prediction_model-
>getCommentsCount($pred_id);


$viewData['predName'] = $predName;

$viewData['user_id'] = $user_id;

$viewData['pred_id'] = $pred_id;

$this->load->database();

$ratings = (array)$this->db->query("SELECT
AVG(rating_score) FROM rating WHERE related_prediction =
$pred_id;")->row();


$viewData['AveRating'] = $ratings['AVG(rating_score)'];

$this->load->view('pages/predictionTemplate', $viewData);

}else{

$data['notifType'] = 'invalid-access';


$this->load->view('pages/notifications', $data);
```

```
}

$this->load->view('templates/footer');

}

public function viewpredictionsby($user_id, $userName){

$data['title'] = "Predictions by ".$userName;

$data['user_id'] = $user_id;

$this->load->view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon', 'image/ico');

$this->load->view('templates/menu');

if (isset($_SESSION['username'])) {

$this->load->model('prediction_model');

$allPredictions = $this

->prediction_model

->get_AllPredictionsBy($user_id);

$predCount = $this

->prediction_model

->get_AllPredictionsByCount($user_id);

$data['predictionCount'] = $predCount;

$data['allPredictions'] = $allPredictions;

$data['isOwnAccount'] = ($_SESSION['username'] == $userName);

$this->load->view('pages/predict', $data);

}else{

$data['notifType'] = 'invalid-access';

$this->load->view('pages/notifications', $data);

}

$this->load->view('templates/footer');

}

public function pred_list(){

$data['title'] = 'Uploaded Predictions';

$this->load->view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon', 'image/ico');

$this->load->view('templates/menu');

if (isset($_SESSION['username'])) {

$this->load->model('prediction_model');

$allPredictions = $this

->prediction_model

->get_AllPredictions();
```

```
$predCount = $this

->prediction_model

->get_AllPredictionsCount();


$data['predictionCount'] = $predCount;


$data['allPredictions'] = $allPredictions;

$this->load->view('pages/predict', $data);

}else{

$data['notifType'] = 'invalid-access';


$this->load->view('pages/notifications', $data);


}


$this->load->view('templates/footer');

}


public function readPredictionFiles($path, $name){


$viewData = array();


$this->load->helper('file');


$viewData['predname'] = $name;


$viewData['predImage'] = base_url().$path."prediction.jpg";


$viewData['predSynthImage'] =
base_url().$path."prediction_synthesis.jpg";

$viewData['predSynthFile'] =
base_url().$path."prediction_synthesis.pksf";


$viewData['predMolFile'] =
base_url().$path."prediction.mol";


$viewData['predSMILES'] = read_file($path."prediction.smi");


$viewData['predDescr'] =
read_file($path."prediction_descr.txt");


return $viewData;

}

public function rateprediction($user_id, $pred_id,
$predName){


$data['title'] = 'Rate "'.$predName.'"';


$data['pred_id'] = $pred_id;


$data['user_id'] = $user_id;

$data['predName'] = $predName;


if (isset($_SESSION['username'])) {

$this->load->library('form_validation');

$this->form_validation->set_rules('rating', 'Rating',
'required|numeric');

//checks if the current user already rated the compound


$viewingUser = $_SESSION['username'];


$userNameQuery = (array)$this->db->query("SELECT uname
FROM users WHERE uname = '$viewingUser';")->row();
```

```
$userPredAuthor = $userNameQuery['uname'];


$data['userPredAuthor'] = $userPredAuthor;


$recordSearch = (array)$this->db->query("SELECT rating_score FROM rating WHERE related_prediction = $pred_id AND rating_source_user = '$userPredAuthor';")->row();

//echo $_SESSION['SOURCE_URI'];


if ($this->form_validation->run() == FALSE)

{


$this->load->database();

//checks the name of the author of the prediction


$userNameQuery = (array)$this->db->query("SELECT uname FROM users WHERE user_id = $user_id;")->row();


$data['userPredAuthor'] = $userNameQuery['uname'];



if(sizeof($recordSearch) > 0){

//edit records

$data['rating'] = $recordSearch['rating_score'];

$data['title'] = 'Edit rating for "'.$predName.'"';


}


$this->load->view('templates/header', $data);


echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon', 'image/ico');


$this->load->view('templates/menu');

$this->load->view('regular/ratingform', $data);


$this->load->view('templates/footer');

}else{


$this->load->database();


$commentEntry = array();


if(sizeof($recordSearch) > 0){

//edit records

$commentEntry['rating_score'] = $_POST['rating'];

$this->db->where('related_prediction', $pred_id)->where('rating_source_user', $_SESSION['username'])->update('rating', $commentEntry);


}else{

//store to database


$commentEntry['related_prediction'] = $pred_id;


$commentEntry['rating_source_user'] = $_SESSION['username'];


$commentEntry['rating_score'] = $_POST['rating'];


$this->db->insert('rating', $commentEntry);
```

```
}


redirect(site_url()."/reguser/viewprediction/".$user_id."/".$pred_id."/".$predName, 'refresh');

}


}else{

$data['notifType'] = 'invalid-access';

$this->load->view('templates/header', $data);


echo link_tag('css/style.css');


echo link_tag('images/favicon.jpg', 'shortcut icon', 'image/ico');


$this->load->view('templates/menu');

$this->load->view('pages/notifications', $data);

$this->load->view('templates/footer');


}

}


public function commentprediction($user_id, $pred_id, $predName){


$data['title'] = 'Comment Form';


$data['pred_id'] = $pred_id;


$data['user_id'] = $user_id;


$data['predName'] = $predName;

if (isset($_SESSION['username'])) {

$this->load->library('form_validation');

$this->form_validation->set_rules('comment', 'Comment', 'required');


//echo $_SESSION['SOURCE_URI'];


if ($this->form_validation->run() == FALSE)

{


$this->load->database();


$userNameQuery = (array)$this->db->query("SELECT uname FROM users WHERE user_id = $user_id;")->row();


$data['userPredAuthor'] = $userNameQuery['uname'];


$this->load->view('templates/header', $data);


echo link_tag('css/style.css');


echo link_tag('images/favicon.jpg', 'shortcut icon', 'image/ico');


$this->load->view('templates/menu');

$this->load->view('regular/commentform', $data);


$this->load->view('templates/footer');

}else{


//store to database
```

```
$this->load->database();

$commentEntry = array();

$commentEntry['related_prediction'] = $pred_id;

$commentEntry['comment_author'] =
$_SESSION['username'];

$commentEntry['comment_text'] = $_POST['comment'];

$this->db->insert('comment', $commentEntry);

redirect(site_url()."/reguser/viewprediction/".$user_id."/".$
pred_id."/".$predName, 'refresh');

}

}else{

$data['notifType'] = 'invalid-access';

$this->load->view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');

$this->load->view('templates/menu');

$this->load->view('pages/notifications', $data);

$this->load->view('templates/footer');

}

}}?>
```

…/controllers/user_profile.php

```
<?php/**
* @author ajmendoza
* @copyright 2012
*/

class User_profile extends CI_Controller{
public function __construct()

{

parent::__construct();

if(!isset($_SESSION))

session_start();}
public function index(){

$data['title'] = 'User Profile';

$this->load->view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon',
'image/ico');
$this->load->view('templates/menu');

if (isset($_SESSION['username']) ) {

$this->load->model('user_model');

$loggedUser = $this

->user_model

->get_user(

$_SESSION['username']

);

$userdata = (array)$loggedUser;

$this->load->model('prediction_model');
```

```
$data['totalPreds'] = $this->prediction_model->getTotalPredictions($userdata['user_id']);

$data = array_merge($data, $userdata);$this->load->view('pages/user_profile_view', $data);

if($data['user_id'] == 1){


//this means that the user is the administrator


$this->load->view('admin/dashboard', $data);

}else{


$this->load->view('regular/dashboard', $data);

}}

$this->load->view('templates/footer');}
}
?>
```

…/controllers/usersform.php

```
<?php/**
* @author ajmendoza
* @copyright 2012
*/

class Usersform extends CI_Controller {

public function __construct()
{

parent::__construct();

if(!isset($_SESSION))


session_start();


}

function index()
{


if ( isset($_SESSION['username']) ) {

redirect('user_profile');

}


$this->load->library('form_validation');


$this->form_validation->set_rules('uname', 'Username', 'required');


$this->form_validation->set_rules('password', 'Password', 'required');if ($this->form_validation->run() == FALSE)
{

$page = 'Users';

$data['title'] = ucfirst($page); // Capitalize the first letter
$this->load->view('templates/header', $data);

echo link_tag('css/style.css');

echo link_tag('images/favicon.jpg', 'shortcut icon', 'image/ico');

$this->load->view('templates/menu', $data);$this->load->view('pages/users');
$this->load->view('templates/footer', $data);
}
else
{
// then validation passed. Get from db

$this->load->model('user_model');

$res = $this

->user_model

->verify_user(


$this->input->post('uname'),


$this->input->post('password')

);
if ( $res != false ) {


$_SESSION['username'] = $this->input->post('uname');
```

```
redirect('user_profile');

}else{


redirect('users');

}}}
public function logout()

{


session_destroy();


redirect('users');

}}
?>
```

…/models/prediction_model.php

```
<?php


/**

* @author ajmendoza

* @copyright 2012

*/


class Prediction_model extends CI_Model {

function __construct()

{

$this->load->database();


}


public function getTotalPredictions($id){

$q = $this

->db

->where('user_id', $id)

->get('prediction');


return $q->num_rows();


}


public function getID($predictionData){

$q = $this

->db

->where('user_id', $predictionData['user_id'])

->where('pred_name', $predictionData['pred_name'])

->get('prediction');

$x = (array)$q->row();

return $x['pred_id'];


}


public function getAuthor($user_id){

$q = $this

->db

->where('user_id', $user_id)

->get('users');

$x = (array)$q->row();

return $x['uname'];


}



public function getComments($pred_id){

$q = $this
```

```
->db

->where('related_prediction', $pred_id)

->order_by("comment_date", "desc")->get('comment');

if ( $q->num_rows > 0 ) {

return $q;

}

return false;



}



public function get_AllCommentsCount(){

$q = $this

->db

->get('comment');

if ( $q->num_rows > 0 ) {

return $q->num_rows();

}

return false;



}





public function get_AllComments(){

$this->db->select('*');

$this->db->from('comment');

$this->db->join('prediction', 'comment.related_prediction =
prediction.pred_id');

$q = $this->db->get();

if ( $q->num_rows > 0 ) {

return $q;

}

return false;



}



public function getCommentsCount($pred_id){

$q = $this

->db

->where('related_prediction', $pred_id)

->get('comment');

if ( $q->num_rows > 0 ) {

return $q->num_rows();

}

return false;



}



public function get_AllPredictions(){

$this->db->select('*');

$this->db->from('prediction');

$this->db->join('users', 'users.user_id = prediction.user_id');

$this->db->order_by("pred_name", "asc");

$query = $this->db->get();

if ( $query->num_rows > 0 ) {

// person has account with us

return $query;

}

return false;

}



public function get_AllPredictionsCount(){

$this->db->select('*');
```

```
$this->db->from('prediction');
$this->db->join('users', 'users.user_id = prediction.user_id');
$query = $this->db->get();
if ( $query->num_rows > 0 ) {
// person has account with us
return $query->num_rows();
}
return false;
}


public function get_AllPredictionsBy($user_id){
$this->db->select('*');
$this->db->from('prediction');
$this->db->where('users.user_id', $user_id);
$this->db->join('users', 'users.user_id = prediction.user_id');
$this->db->order_by("pred_name", "asc");
$query = $this->db->get();
if ( $query->num_rows > 0 ) {
// person has account with us
return $query;
}
return false;
}

public function get_AllPredictionsByCount($user_id){
$this->db->select('*');
$this->db->from('prediction');
$this->db->where('users.user_id', $user_id);
$this->db->join('users', 'users.user_id = prediction.user_id');
$query = $this->db->get();
if ( $query->num_rows > 0 ) {
// person has account with us
return $query->num_rows();
}
return false;
}


}
?>
```

…/models/user_model.php

```
<?php

/**
* @author ajmendoza
* @copyright 2012
*/

class User_model extends CI_Model {
function __construct()
{
$this->load->database();


}

public function verify_user($uname, $password)
{
$q = $this
->db
->where('uname', $uname)
->where('password', sha1($password))
->where('approved', 1)
```

```
->limit(1)

->get('users');


if ( $q->num_rows > 0 ) {

// person has account with us

return $q->row();

}

return false;

}


public function get_user($uname)

{

$q = $this

->db

->where('uname', $uname)

->where('approved', 1)

->limit(1)

->get('users');


if ( $q->num_rows > 0 ) {

// person has account with us

return $q->row();

}

return false;

}


public function get_userID($id)

{

$q = $this

->db

->where('user_id', $id)

->where('approved', 1)

->limit(1)

->get('users');


if ( $q->num_rows > 0 ) {

// person has account with us

return $q->row();

}

return false;

}


public function get_unauthUserList(){

$q = $this

->db

->where('approved', 0)

->get('users');


if ( $q->num_rows > 0 ) {

// person has account with us

return $q;

}

return false;

}


public function get_unauthUserListCount(){

$q = $this

->db

->where('approved', 0)

->get('users');


if ( $q->num_rows > 0 ) {
```

```
// person has account with us
return $q->num_rows();
}
return false;
}
}
?>
```

…/views/admin/allPKNP.php

```
<?php

/**
* @author ajmendoza
* @copyright 2013
*/
$this->load->library('table');
?><div id="main-body">

<h3><?php echo $title; ?></h3>
<?php echo br(1);

if(sizeof($pknpRecords) > 0){

echo "<p>NOTE: Once a natural product is deleted, its associated synthase(s) will be deleted as well.</p>";
echo "<table class = 'pkrecords' id='pkrecords-table'>";
echo "<thead><th> </th><th>Polyketide Name</th><th>Action</th></thead>";

foreach ($pknpRecords as $row)
{
$row = (array)$row;
echo "<tr>";
$id = $row['natprod_name'];
echo "<th width=20% bgcolor=white><img width=100% src=".base_url()."polyketide_images/".$row['pknpImage'].">  </th>";
echo "<th>".$id."</th>";
echo "<th>".anchor("editpolyketide/$id", "edit")." ".anchor("deletepolyketide/$id", "delete")."</th>";
echo "</tr>";
}

echo "</table>";
}else{
echo "<p>There are currently no polyketide natural product.</p>";
}
?>

</div>
```

…/views/admin/dashboard.php

```
<div id="main-body">

<h3>Administrator's Dashboard</h3>
<?php echo br(1); ?>
<table id="admin-dashboard">
<tr>
<td width="33%">
<?php echo anchor('admin/pending', '<img src = '.base_url().'images/user-add.png><br />View pending account requests.'); ?>
</td>
<td width="33%">
<?php echo anchor('addpolyketide', '<img src = '.base_url().'images/addPKNP.png><br />Add new polyketide record'); ?>
```

```
</td>
<td width="33%">
<?php echo anchor('moderatepolyketide', '<img src = '.base_url().'images/addPKNP.png><br />Moderate existing polyketide record'); ?>
</td>
</tr>
<tr>
<td width="33%">
<?php echo anchor('addpksrecord', '<img src = '.base_url().'images/addPKS.gif><br />Add new polyketide synthase record.'); ?>
</td>
<td width="33%">
<?php echo anchor('moderatepks', '<img src = '.base_url().'images/addPKS.gif><br />Moderate existing polyketide synthase record'); ?>
</td>
<td width="33%">
<?php echo anchor('moderatepredictions', '<img src = '.base_url().'images/addPKNP.png><br />Moderate predictions'); ?>
</td>
</tr>
</table>
</div>
```

…/views/admin/moderatecomments.php

```
<?php


/**
* @author ajmendoza
* @copyright 2013
*/
$this->load->library('table');
?><div id="main-body">


<h3><? echo $title; ?></h3>
<?php echo br(1);


echo "<p align=center><table width=70%><th width=50%>".anchor("moderatepredictions", "Moderate predictions");
echo "</th><th width=50%>".anchor("moderatecomments", "Moderate comments")."</table></p>";


if($resultsTotal > 0){


echo "<table class = 'pending' id='pending-users' width=80%>";
echo "<thead><th width=5%>Date</th><th width=15%>Author</th><th width=15%>Prediction</th><th width=50%>Comment</th><th width=10%>Action</th></thead>";
foreach ($comments->result_array() as $row)
{
echo "<tr>";
$id = $row['comment_id'];
echo "<th>".$row['comment_date']."</th>";
echo "<th>".$row['comment_author']."</th>";
echo "<th>".$row['pred_name']."</th>";
echo "<th>".$row['comment_text']."</th>";
echo "<th>".anchor("deletecomment/$id/", "delete")."</th>";
echo "</tr>";
}


echo "</table>";
```

```
}else{
echo "<p>There are no comments from the users.</p>";
}
?>

</div>
```

…/views/admin/moderatepreds.php

```
<?php

/**
* @author ajmendoza
* @copyright 2013
*/
$this->load->library('table');


?><div id="main-body">

<h3><? echo $title; ?></h3>
<?php echo br(1);

echo "<p align=center><table width=70%><th width=50%>".anchor("moderatepredictions", "Moderate predictions");
echo "</th><th width=50%>".anchor("moderatecomments", "Moderate comments")."</table></p>";

if($resultsTotal > 0){

echo "<table class = 'pending' id='pending-users'>";
echo "<thead><th>#</th><th>Name</th><th>Submission date</th><th>Action</th></thead>";
foreach ($predictions->result_array() as $row)
{
echo "<tr>";
$id = $row['pred_id'];
$predName = $row['pred_name'];
echo "<th>".$id."</th>";
echo "<th>".$predName."</th>";
echo "<th>".$row['submission_date']."</th>";
$author = $row['user_id'];
echo "<th>".anchor("deletepred/$author/$id/$predName", "delete")."</th>";
echo "</tr>";
}

echo "</table>";
}else{
echo "<p>No prediction is uploaded by user.</p>";
}
?>

</div>
```

…/views/admin/pendingUsersPage.php

```
<?php

/**
* @author ajmendoza
* @copyright 2013
*/
$this->load->library('table');
?><div id="main-body">
```

```
<h3>List of all pending users</h3>

<?php echo br(1);


if($resultsTotal > 0){


echo "<table class = 'pending' id='pending-users'>";

echo "<thead><th>Username</th><th>Name</th><th>Occupation</th><th>Email</th><th>Action</th></thead>";

foreach ($pendingUsers->result_array() as $row)

{

echo "<tr>";

$id = $row['user_id'];

echo "<th>".$row['uname']."</th>";

echo "<th>".$row['fname']." ".$row['surname']."</th>";

echo "<th>".$row['occupation']."</th>";

echo "<th>".$row['email']."</th>";

echo "<th>".anchor("admin/approveUser/$id", "approve")." ".anchor("admin/deleteUser/$id", "delete")."</th>";

echo "</tr>";

}


echo "</table>";

}else{

echo "<p>There are no users waiting for approval.</p>";

}

?>


</div>
```

…/views/admin/remPKS.php

```
<?php


/**

* @author ajmendoza

* @copyright 2013

*/

$this->load->library('table');



?><div id="main-body">


<h3><?php echo $title; ?></h3>

<?php echo br(1);


if(sizeof($pknpRecords) > 0){



echo "<p>Listed below are polyketide synthases that are not part of a polyketide natural product. This form only allows removal of the selected record.</p>";

echo "<table class = 'pkrecords' id='pkrecords-table'>";

echo "<thead><th>Polyketide Name</th><th>Action</th></thead>";

foreach ($pknpRecords as $row)

{

$row = (array)$row;

echo "<tr>";

$id = $row['PKS_id'];

echo "<th>".$row['PKS_name']."</th>";

echo "<th>".anchor("deletepks/$id", "delete")."</th>";

echo "</tr>";

}
```

```
echo "</table>";
}else{
echo "<p>All polyketide synthases are already associated with a natural product.</p>";
}
?>
</div>
```

…/views/pages/about.php

```
<?php


/**
* @author ajmendoza
* @copyright 2012
*/



?>
<div id="main-body">

<h3>About</h3>
<p>This system is composed of two main parts: <strong>PKDB</strong> or the Polyketide Database, and <strong>Predyketide</strong> or the Polyketide Prediction Software.</p>
<? echo br(2); ?>
<p>The PKDB component is an online database system where users can navigate on the existing Type I modular polyketides. The data that can be viewed from this are the synthases, the biosynthesis, the structure, and the modules involved in the creation of the natural product.</p>
<? echo br(2); ?>
<p>Predyketide, is a JAR executable desktop application with graphical user interfaces. This component allows user to create a Type I modular polyketide biosynthesis on a dry laboratory and even with no internet.</p>
<? echo br(3); ?>
</div>
```

…/views/pages/contact.php

```
<?php


/**
* @author ajmendoza
* @copyright 2012
*/


?><div id="main-body">

<h3>Contact</h3>
<p>For comments and/or suggestions, kindly fill-out the form below. Thank you!</p>
<?php
echo "<span class='form-validations'>".validation_errors()."</span>";
echo form_open('contactform');
?>

<h5>Name</h5>
<input type="text" name="name" <?php if(isset($_SESSION['username'])) echo "disabled"; ?> value="<?php echo (isset($_SESSION['username'])) ? $loggedUser['fname']." ".$loggedUser['surname']: set_value('name'); ?>" size="50" />
<?php echo br(2); ?>
<h5>Email Address</h5>
<input type="text" name="email" <?php if(isset($_SESSION['username'])) echo "disabled"; ?>
```

```
value="<?php echo (isset($_SESSION['username'])) ? $loggedUser['email']: set_value('email'); ?>" size="50" />

<?php echo br(2); ?>

<h5>Message</h5>

<textarea name="message" rows="8" cols="50" /><?php echo set_value('message'); ?></textarea>

<?php echo br(2);



echo "<div style='margin: 10px 10px 10px 110px'>".recaptcha_get_html("6Ldtc9wSAAAAANwwkWJxTLJTbkK_7gE5lSzOAIbf")."</div>";

?>

<div><input type="submit" value="Submit" /></div>

<?php echo br(3); ?>

</form>

</div>
```

…/views/pages/create_user_form.php

```
<?php



/**

* @author ajmendoza

* @copyright 2012

*/





?><div id="main-body">



<h3>Create New Account</h3>

<p>All fields with (*) are required for the site authorized user account application.</p>

<?php

echo "<span class='form-validations'>".validation_errors()."</span>";

echo form_open('create_user');

?>


<h5>Username (max of 12 chars)*</h5>

<input type="text" name="uname" value="<?php echo set_value('uname'); ?>" size="30" />

<?php echo br(2); ?>

<h5>Password*</h5>

<input type="password" name="password" size="30" />

<?php echo br(2); ?>

<h5>Confirm Password*</h5>

<input type="password" name="password2"  size="30" />

<?php echo br(2); ?>

<h5>First name*</h5>

<input type="text" name="fname" value="<?php echo set_value('fname'); ?>" size="30" />

<?php echo br(2); ?>

<h5>Surname*</h5>

<input type="text" name="surname" value="<?php echo set_value('surname'); ?>" size="30" />

<?php echo br(2); ?>

<h5>Email address*</h5>

<input type="text" name="email" value="<?php echo set_value('email'); ?>" size="30" />

<?php echo br(2); ?>

<h5>Home address</h5>

<input type="text" name="home_address" value="<?php echo set_value('home_address'); ?>" size="30" />

<?php echo br(2); ?>

<h5>Contact number</h5>

<input type="text" name="contact_num" value="<?php echo set_value('contact_num'); ?>" size="30" />

<?php echo br(2); ?>
```

```
<h5>Occupation*</h5>

<input type="text" name="occupation" value="<?php echo
set_value('occupation'); ?>" size="30" />

<?php echo br(2); ?>

<h5>Short Bio (300 characters)</h5>

<textarea name="short_bio" rows="8" cols="50"
maxlength="300"/><?php echo set_value('short_bio');
?></textarea>

<?php echo br(2);



echo "<div style='margin: 10px 10px 10px
110px'>".recaptcha_get_html("6Ldtc9wSAAAAANwwkWJxTLJ
TbkK_7gE5lSzOAIbf")."</div>";

?>

<div><input type="submit" value="Submit" /></div>

<?php echo br(3); ?>

</form>

</div>
```

…/views/pages/createpks.php

```
<?php



/**

* @author ajmendoza

* @copyright 2012

*/



$selectExtenderUnitForm = "<select name='pksunits[]'>";

$selectStarterUnitForm = "<select name='pksunits[]'>";

foreach ($modUnits->result_array() as $row)

{

if($row["module_type"]=="extender")

$selectExtenderUnitForm.="<option
value=".$row['module_id'].">(Extender)
".$row['module_name']."</option>";

else

$selectStarterUnitForm.="<option
value=".$row['module_id'].">(Starter)
".$row['module_name']."</option>";



}



$selectExtenderUnitForm.="</select>".nbs(10);

$selectStarterUnitForm.="</select>".nbs(10);

$selectExtenderUnitForm.="<select name='domainseq[]'>
<option value='KS-AT-ACP'>KS-AT-ACP</option> <option
value='KS-AT-KR-ACP'>KS-AT-KR-ACP</option> <option
value='KS-AT-DH-KR-ACP'>KS-AT-DH-KR-ACP</option>
<option value='KS-AT-DH-ER-KR-ACP'>KS-AT-DH-ER-KR-
ACP</option>  </select>";

$selectStarterUnitForm.="<select hidden
name='domainseq[]'><option value='' selected>-
</option></select>";

?>

<script>



function addExtender(divName){

var newDiv=document.createElement('div');



newDiv.innerHTML= <? print
"\"$selectExtenderUnitForm\""?>;

document.getElementById(divName).appendChild(newDiv);

}



function addStarter(divName){

var newDiv=document.createElement('div');



newDiv.innerHTML= <? print
"\"$selectStarterUnitForm\""?>;
```

```
document.getElementById(divName).appendChild(newDiv);
}
</script>
<div id="main-body">

<h3><? echo $title; ?></h3>
<p>Create a new polyketide synthase (PKS) using the form below.</p>
<?php
echo "<span class='form-validations'>".validation_errors()."</span>";
echo form_open('addpksrecord');
?>

<h5>PKS Name</h5>
<input type="text" name="PKS_name" value="<?php echo set_value('PKS_name'); ?>" size="50" />
<?php echo br(2); ?>
<h5>Source Gene</h5>
<input type="text" name="source_gene" value="<?php echo set_value('source_gene'); ?>" size="50" />
<?php echo br(2); ?>
<h5>Modular Units
<input type="button" style="margin: 5px;" value="Add Starter" onclick="addStarter('dynamicInput');" />
<input type="button" style="margin: 5px;" value="Add Extender" onclick="addExtender('dynamicInput');" /></h5>
<div id="dynamicInput" style="margin: 10px 10px 10px 150px;"></div>
<?php echo br(2); ?>
<h5>Supporting Resources</h5>
<textarea name="resources" rows="4" cols="50" /><?php echo set_value('resources'); ?></textarea>
<?php echo br(2); ?>
<div><input type="submit" value="Submit" /></div>
<?php echo br(3); ?>
</form>
</div>
```

<u>…/views/pages/dbase.php</u>

```
<?php

/**
* @author ajmendoza
* @copyright 2012
*/


?>
<div id="main-body">
<h3><?php echo $title;?></h3>

<table width = 100%>
<?
$counter = 0;
foreach ($pkdbRecords->result_array() as $row)
{
if($counter%4 == 0)
{
print "\n<tr>\n";
}

print "<td width=25%>".anchor("viewPKNP/".$row['natprod_name'], $row['natprod_name'])."</td>\n";

if($counter%4 == 0)
```

```
{

print "\n</tr>\n";

}

}

?>

</table>

</div>
```

…/views/pages/notifications.php

```
<?php



/**

* @author ajmendoza

* @copyright 2013

*/



$notifMessage = "";



switch($notifType){

case "create-user-pending":

$notifMessage = "Your account has already been created. Kindly wait for administrator's review.";

break;

case "invalid-access":

$notifMessage = "You do not have the rights to access this page.";

break;

case "pks-add-success":

$notifMessage = "The new PKS is successfully added to the database.".br(3)."To add a new record, click ".anchor("./addpksrecord", "here").".";



break;

}



?><div id="main-body">



<h3><?php echo $title?></h3>



<?php echo br(1); ?>



<p><?echo $notifMessage; ?></p>



<?php echo br(2); ?>

</form>

</div>
```

…/views/pages/PKNPForm.php

```
<?php



/**

* @author ajmendoza

* @copyright 2013

*/





?>

<script type="text/javascript" src="<?php echo base_url();?>js/selectbox.js" ></script>

<div id="main-body">



<h3><? echo $title;?></h3>
```

```
<p>Create a new polyketide natural product (PKNP) using the form below. Kindly take note of the PKS order.</p>

<?php

echo "<span class='form-validations'>".validation_errors()."</span>";

echo "<span class='form-validations'>".$error."</span>";

echo form_open_multipart('addpolyketide');

?>



<h5>PKNP Name</h5>

<input type="text" name="pknpName" value="<?php echo set_value('pknpName'); ?>" size="50" />

<?php echo br(2);?>



<h5>Host Organism</h5>

<input type="text" name="host_organism" value="<?php echo set_value('host_organism'); ?>" size="50" />

<?php echo br(2);?>




<h5>Natural Product Image*</h5>

<input type="file" name="userfile" size="30" />

<?php echo br(2);?>



<h5>PKS Sequence*</h5>

<table border="0" style="margin: 10px 10px 10px 110px;">

<tr>

<td>

<select name="pksavailable" size="10" multiple="multiple" style="width: 100px;">

<?

foreach ($pksRecords->result_array() as $row)

{

echo "<option value='".$row['PKS_id']."'>".$row['PKS_name']."</option>\n";

}

?>



</select>

</td>



<td>

<input type="button" class="transportButtons" value="--&gt;" onclick="moveOptions(this.form.pksavailable, this.form.pksTosubmit);" /><br />

<input type="button" class="transportButtons" value="&lt;--" onclick="moveOptions(this.form.pksTosubmit, this.form.pksavailable);" />

</td>



<td>

<select name="pksTosubmit" size="10" multiple="multiple" style="width: 100px;">

</select>

</td>

</tr>

</table>



<h5>Description</h5>

<textarea name="descr" rows="4" cols="50" /><?php echo set_value('descr');?></textarea>

<?php echo br(2)?>



<textarea hidden name="pkssubmitted" id="pkssubmitted"></textarea>



<div><input type="submit" value="Submit" /></div>
```

```
<?php echo br(3);?>

</form>
</div>
```

__…/views/pages/PKNPTemplate.php__

```
<?php

/**
* @author ajmendoza
* @copyright 2012
*/

$pksCount = sizeof($pknpRecord);

?>
<div id="main-body" style="overflow: auto;">

<div style="width: 50%;  margin: 0px auto; position: relative; background: #ffffff"><img src = "<?php echo base_url().'polyketide_images/'.$basicInfo['pknpImage'];?>" width = 100%/> </div>

<h3><?php echo $basicInfo['natprod_name'];?></h3>
<h5><i><?php echo $basicInfo['organism_specie'];?></i></h5>

<p><?php echo $basicInfo['NP_desc'];?></p>
<div style="width: 300px; margin-left: 110px; margin-top: 5px; font-size: 12px; float: left; position: relative; word-wrap: break-word;"><strong  style="color: #7eae19;">POLYKETIDE SYNTHASES</strong>
</div>
<?php
for($count = 0; $count < $pksCount; )
{
$row = (array)$pknpRecord[$count];
$pksName = $row['PKS_name'];
print "
<
div style='width: 350px; margin-left: 150px; margin-top: 5px; font-size: 12px; float: left; word-wrap: break-word;'>";
print "
<
div ><strong  style='color: #7eae19;'>PKS Name:</strong> ".$pksName."</div>";
print "
<
div><strong  style='color: #7eae19;'>Gene:</strong> ".$row['gene']."</div>";
print "<div style='margin-left: 15px;'>";
print "

<t
able width='90%'>";
while($pksName == $row['PKS_name']){
print "<tr>";
print "
<t
h width = 60% style='text-align: left; font-weight: normal; font-size: 12px'><strong>".$row['module_name']."</strong></th>";
print "
<t
h width = 40% style='font-weight: normal; font-size: 12px; text-align: left;'>[".$row['domain_seq']."]</th>";
print "</tr>";
$count++;
if($count < $pksCount)
$row = (array)$pknpRecord[$count];
else{
if($count == $pksCount){
print "<tr>";
```

```
print "

<t

h width = 60% style='text-align: left;'><strong>------
</strong></th>";

print "

<t

h width = 40% style='font-weight: normal; text-align:
left;'>["."TE"."]</th>";

print "</tr>";

}

break;

}

}

print "     </table>";

print "</div>";

print "</div>";

}

?>

</div>
```

<u>…/views/pages/PKSForm.php</u>

```
<?php


/**

* @author ajmendoza

* @copyright 2012

*/





?>

<div id="main-body" style="overflow: auto;">

<div style="width: 30%;  margin: 0px auto; position: relative;
background: #ffffff"><img src =
"/polykase/polyketide_images/Erythromycin.png" width =
100%/> </div>



<div style="width: 300px; margin-left: 110px; margin-bottom:
10px; margin-top: 10px; float: left; position: relative; font-
size: 20px; color: #7eae19;">Erythromycin</div>


<div style="width: 300px; margin-left: 110px; margin-top:
5px; font-size: 12px; float: left; position: relative; word-wrap:
break-word;"><strong style="color: #7eae19;">Host
Organism:</strong>&nbsp;<i>Saccharopolyspora
erythraea</i>

</div>



<div style="width: 300px; margin-left: 110px; margin-top:
5px; font-size: 12px; float: left; position: relative; word-wrap:
break-word;"><strong  style="color: #7eae19;">POLYKETIDE
SYNTHASES</strong>

</div>


<div style="width: 350px; margin-left: 150px; margin-top:
5px; font-size: 12px; float: left; position: relative; word-wrap:
break-word;">

<div ><strong  style="color: #7eae19;">PKS Name:</strong>
eryth_001</div>

<div><strong  style="color: #7eae19;">Gene:</strong>
eryAI</div>

<div style="margin-left: 15px;">

<table width="90%">

<tr>

<th width = 60% style="text-align: left;"><strong>Propionyl-
CoA</strong></th>

<th width = 40% style="font-weight: normal; text-align:
left;">[AT-ACP]</th>
```

```
</tr>
<tr>
<th width = 60% style="text-align: left;"><strong>Propionyl-CoA</strong></th>
<th width = 40% style="font-weight: normal; text-align: left;">[AT-ACP]</th>
</tr>

</table>
</div>
</div>
</div>
```

…/views/pages/polyketideTemplate.php

```
<?php

/**
* @author ajmendoza
* @copyright 2012
*/

?>
<div id="main-body" style="overflow: auto;">

<div style="width: 30%;  margin: 0px auto; position: relative; background: #ffffff"><img src = "/polykase/polyketide_images/Erythromycin.png" width = 100%/> </div>
<div style="width: 300px; margin-left: 110px; margin-bottom: 10px; margin-top: 10px; float: left; position: relative; font-size: 20px; color: #7eae19;">Erythromycin</div>

<div style="width: 300px; margin-left: 110px; margin-top: 5px; font-size: 12px; float: left; position: relative; word-wrap: break-word;"><strong style="color: #7eae19;">Host Organism:</strong>&nbsp;<i>Saccharopolyspora erythraea</i>
</div>

<div style="width: 300px; margin-left: 110px; margin-top: 5px; font-size: 12px; float: left; position: relative; word-wrap: break-word;"><strong  style="color: #7eae19;">POLYKETIDE SYNTHASES</strong>
</div>

<div style="width: 350px; margin-left: 150px; margin-top: 5px; font-size: 12px; float: left; position: relative; word-wrap: break-word;">
<div ><strong  style="color: #7eae19;">PKS Name:</strong> eryth_001</div>
<div><strong  style="color: #7eae19;">Gene:</strong> eryAI</div>
<div style="margin-left: 15px;">
<table width="90%">
<tr>
<th width = 60% style="text-align: left;"><strong>Propionyl-CoA</strong></th>
<th width = 40% style="font-weight: normal; text-align: left;">[AT-ACP]</th>

</tr>
<tr>
<th width = 60% style="text-align: left;"><strong>Propionyl-CoA</strong></th>
<th width = 40% style="font-weight: normal; text-align: left;">[AT-ACP]</th>
```

```
</tr>

</table>
</div>
</div>
 </div>
```

…/views/pages/predict.php

```
<?php

/**
* @author ajmendoza
* @copyright 2012
*/
?><div id="main-body">

<h3><? echo $title; ?></h3>
<?php echo br(1);

if($predictionCount > 0){

echo "<table class = 'pending' id='pending-users'>";
foreach ($allPredictions->result_array() as $row)
{
echo "<tr>";
$user_id = $row['user_id'];
$pred_id = $row['pred_id'];
$predName = $row['pred_name'];
echo "<th style='width: 50px;'><img width=100% src =
".base_url()."predictions/$user_id/$pred_id/prediction.jpg".
"></th>";
echo
"<th>".anchor("reguser/viewprediction/$user_id/$pred_id/$
predName/", $predName)." by ".$row['uname']."</th>";
if(isset($isOwnAccount) && $isOwnAccount){
echo
"<th>".anchor("deletepred/$user_id/$pred_id/$predName/"
, "delete")."</th>";
}
echo "</tr>";

}

echo "</table>";
}else{
echo "<p>Nobody shared any prediction.</p>";
}
?>

</div>
```

…/views/pages/predictionTemplate.php

```
<?php

/**
* @author ajmendoza
* @copyright 2012
*/


$atts = array(
'width'      => '800',
'height'     => '600',
```

```
'scrollbars' => 'yes',
'status'     => 'yes',
'resizable'  => 'yes',
'screenx'    => '0',
'screeny'    => '0'
);


$attsSynth = array(
'width'      => '800',
'height'     => '400',
'scrollbars' => 'yes',
'status'     => 'yes',
'resizable'  => 'yes',
'screenx'    => '0',
'screeny'    => '0'
);
?>
<script language="javascript">
function toggle() {
var ele = document.getElementById("toggleText");
var text = document.getElementById("displayText");
if(ele.style.display == "block") {
ele.style.display = "none";
text.innerHTML = "View comments";
}
else {
ele.style.display = "block";
text.innerHTML = "Hide comments";
}
}
</script>
```

```
<div id="main-body" style="overflow: auto;">


<div style="width: 300px; margin: 10px 10px 10px 110px;
font-size: 20px; float: left; position: relative; color: #7eae19;"
><? echo $predname; ?></div>

<div style="width: 300px; margin-left: 110px; margin-top:
5px; font-size: 12px; float: left; position: relative; word-wrap:
break-word;"><strong style="color: #ffffff;"><? echo "by
".$author; ?></strong></div>

<div style="width: 200px; height: 200px;  margin-right:
110px; float: right; position: relative;"><img src = "<? echo
$predImage; ?>" width = 100%/> </div>


<div style="width: 300px; margin-left: 110px; margin-top:
5px; font-size: 12px; float: left; position: relative; word-wrap:
break-word;"><strong style="color: #7eae19;">Average
Rating:</strong>&nbsp;<? echo $AveRating; ?>

</div>

<div style="width: 300px; margin-left: 110px; margin-top:
5px; font-size: 12px; float: left; position: relative; word-wrap:
break-word;"><strong style="color:
#7eae19;">SMILES:</strong>&nbsp;<? echo $predSMILES; ?>

</div>


<div style="width: 300px; margin-left: 110px; margin-top:
5px; font-size: 12px; float: left; position: relative;"><strong
style="color: #7eae19;">Molfile:&nbsp;</strong>

<?php echo anchor_popup($predMolFile, 'Download here',
$atts);?>

</div>


<div style="width: 300px; margin-left: 110px; margin-top:
5px; font-size: 12px; float: left; position: relative;"><strong
style="color: #7eae19;">Descriptors:&nbsp;</strong>

<pre style="margin-left: 15px;"><? echo $predDescr; ?>

</pre>

</div>
```

```
<div style="width: 200px; margin-right: 110px; margin-top: 30px; font-size: 10px; float: right; position: relative; word-wrap: break-word;">

<?php echo anchor_popup($predSynthImage, '<img src ='.$predSynthImage.' width = 100%/>', $attsSynth);?>

<br />

<?php echo anchor_popup($predSynthFile, 'Download synthesis file', $atts);?>

</div>



<div style="width: 300px; margin-left: 110px; margin-top: 5px; font-size: 12px; float: left; position: relative; word-wrap: break-word;">

<strong style="color: #7eae19;"><a href="<?php echo site_url()."/rate/".$user_id."/".$pred_id."/".$predName; ?>">Rate!</a></strong>&nbsp;&nbsp;&nbsp;&nbsp;

<strong style="color: #7eae19;"><a id="displayText" href="javascript:toggle();">View comments</a></strong>&nbsp;&nbsp;&nbsp;&nbsp;

<strong style="color: #7eae19;"><a href="<?php echo site_url()."/comment/".$user_id."/".$pred_id."/".$predName; ?>">Add comment</a></strong>&nbsp;&nbsp;&nbsp;&nbsp;

</div>

<div id="toggleText" style="width: 300px; margin-left: 110px; margin-top: 5px; float: left; position: relative; word-wrap: break-word; display: none;">

<?

if($commentsCount){

foreach ($comments->result_array() as $row)

{

echo "<div style='display: block; width: 300px; margin-left: 20px; margin-top: 5px; float: left; position: relative; word-wrap: break-word;'><strong>".$row['comment_author']."</strong>&nbsp;on ".$row['comment_date']."<br />".$row['comment_text']."</div>\n";

}

}else{

echo "<div style='display: block; width: 300px; margin-left: 110px; margin-top: 5px; float: left; position: relative; word-wrap: break-word;'>There are no comments in this prediction.</div>\n";

}

?>

</div>

</div>
```

…/views/pages/user_profile.php

```
<?php



/**

* @author ajmendoza

* @copyright 2012

*/

$title = "User Profile";

?>

<div id="main-body">

<?php



if (isset($_SESSION['username']) ) {

print "<h5>User profile</h5>";

print br(3);



}else{

print "<p>User not found in database.</p>";

}

?>

</div>
```

…/views/pages/users.php

```
<?php


/**

* @author ajmendoza

* @copyright 2012

*/



?><div id="main-body">


<h3>Users</h3>

<p>Kindly log-in with your system credentials.<br />

<?php echo anchor('create_user', 'Not yet a user?'); ?></p>

<?php


echo "<span class='form-validations'>".validation_errors()."</span>";

echo form_open('users');

?>


<h5>Username</h5>

<input type="text" name="uname" value="<?php echo set_value('uname'); ?>" size="50" />

<?php echo br(2); ?>

<h5>Password</h5>

<input type="password" name="password" size="50" />

<?php echo br(2); ?>

<div><input type="submit" value="Submit" /></div>

<?php echo br(3); ?>

</form>

</div>
```

…/views/regular/commentform.php

```
<?php


/**

* @author ajmendoza

* @copyright 2012

*/



?><div id="main-body">


<h3><? echo $title; ?></h3>

<p>Use this form to give comments on the prediction <? echo $predName; ?> by <? echo $userPredAuthor; ?></p>

<?php echo br(2); ?>

<?php

echo "<span class='form-validations'>".validation_errors()."</span>";

echo form_open('comment/'.$user_id."/".$pred_id."/".$predName);

?>


<h5>Comment</h5>

<textarea name="comment" rows="8" cols="50" /><?php echo set_value('comment'); ?></textarea>

<?php echo br(2); ?>

<div><input type="submit" value="Submit" /></div>

<?php echo br(3); ?>

</form>

</div>
```

…/views/regular/dashboard.php

```
<?

$MAXPREDICTIONS = 3;
```

```
?>

<div id="main-body">



<h3>User's Dashboard</h3>

<?php echo br(1); ?>



<p><?php echo anchor('dbase', 'View existing polyketide records.'); //redirect to db page ?></p>

<p><?php echo anchor('reguser/viewpredictionslist', 'View predictions shared by co-users.');  //redirect to pred page?></p>

<?php

if($totalPreds){

echo "<p>".anchor('reguser/viewpredictionsby/'.$user_id.'/'.$uname.'/', 'View uploaded predictions.')."</p>";

}

if($MAXPREDICTIONS > $totalPreds){

echo "<p>".anchor('reguser/addpredictions/'.$user_id, 'Share one of your predictions.')."</p>";

}

?>

</div>
```

…/views/regular/pred_form.php

```
<?php



/**

* @author ajmendoza

* @copyright 2013

*/

?><div id="main-body">



<h3><?php echo $title; ?></h3>

<p>All fields with (*) are required for the site authorized user account application.</p>

<?php

echo "<span class='form-validations'>".validation_errors()."</span>";

echo "<span class='form-validations'>".$error."</span>";

$formPath = 'reguser/addpredictions/'.$user_id;

echo form_open_multipart($formPath);

?>


<h5>One-word name of prediction*</h5>

<input type="text" name="predname" value="<?php echo set_value('predname'); ?>" size="30" />

<?php echo br(2); ?>

<h5>The polyketide zip file exported from Predyketide(.pk.zip)*</h5>

<input type="file" name="userfile" size="30" />

<?php echo br(3); ?>

<div><input type="submit" value="Submit" /></div>

<?php echo br(3); ?>

</form>

</div>
```

…/views/regular/ratingform.php

```
<?php



/**

* @author ajmendoza

* @copyright 2012

*/



?>

<div id="main-body">
```

```
<h3><? echo $title; ?></h3>
<p>Use this form to rate or edit your ratings for <? echo $predName." by ".$userPredAuthor; ?>.</p>
<?php echo br(2); ?>
<?php
echo "<span class='form-validations'>".validation_errors()."</span>";
echo form_open('rate/'.$user_id."/".$pred_id."/".$predName);
?>

<h5>Rank this prediction (0-10) with 0 as the lowest, and 10 as the highest.</h5>
<input type="number" name="rating" width="20px" min=0 max=10 value="<?php if(isset($rating)) echo $rating;?>"/>
<?php echo br(2); ?>
<div><input type="submit" value="Submit" /></div>
<?php echo br(3); ?>
</form>
</div>
```

…/views/template/footer.php

```
<?php


/**
* @author ajmendoza
* @copyright 2012
*/
?>
<div id="footer">
Special problem of John Althom Aduna-Mendoza <br />
Copyright 2013<br />
All rights reserved.
</div>
</div>
</body>
</html>
```

…/views/template/header.php

```
<?php


/**
* @author ajmendoza
* @copyright 2012
*/


if(!isset($_SESSION))
session_start();


?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link href="style.css" rel="stylesheet" type="text/css" />
<title><?php echo $title ?></title>
</head>


<body>
<div id="container">
```

…/views/template/menu.php

```
<?php

/**
* @author ajmendoza
* @copyright 2012
*/
?>
<div id="menu">
<ul>
<li><a href="<? echo base_url();?>">Home</a></li>
<li><a href="<? echo site_url("dbase");?>">Database</a></li>
<li><a href="<? echo site_url("reguser/viewpredictionslist");?>">Prediction</a></li>
<li><a href="<? echo site_url("about");?>">About</a></li>
<li><a href="<? echo site_url("contactform");?>">Contact</a></li>
<li><a href="<? echo site_url("users");?>">Users</a></li>
</ul>
</div>
<div id="greeting">
<?php
if(isset($_SESSION['username'])){
echo "<p>Hello, ".$_SESSION['username']."!</p>";
}
?>
</div>
```

XI. Acknowledgement

Sa wakas! Ito an ang pinakahinihintay kong isulat na bahagi. Bukod sa dito ko kasi ililimbag ang aking mga pasasalamat, ito lang din ang natatanging lugar para sa akin upang mag-Tagalog.

Pinakauna kong pasasalamatan ang ating Panginoong Hesukristo na gumabay at nagbigay sa akin ng talino't kasipagan upang tapusin itong aking Special Problem (SP). Kung wala ang Kaniyang presensiya, tiyak naman na sa kawalan din ako marahil mapupulot ngayon. Maraming-maraming salamat po sa Inyo. Ngunit, higit pa sa pagtulong sa aking SP, Kayo po ay laging nandiyan para sa akin. Hindi ko na po alam kung papaano ko pa po Kayo mapapasalamatan. Sana, ang pagsasabuhay ng Inyong mga gawa at aral ay maging sapat na.

Malaki rin ang aking pasasalamat sa mga mahahalagang tao sa likod ng tagumpay nitong proyekto na ito. Hindi magiging possible ang topic na ito kung wala ang gabay ng aking adviser na si Ms. Ma. Sheila A. Magboo. Ma'am, kahit kaunti lang ang ating personal at harapan na advising session dahil sa medyo hindi swak na schedule, lagi naman kayong handa magreply via text at e-mail. Salamat ng marami Ma'am kasi kung hindi niyo po ako na-persuade out of my original topic, baka po hindi pa ako tapos hanggang ngayon. At, may bonus pa! Thanks ma'am ng maraming-marami. Taus-puso rin ang aking pasasalamat kay Sir/Kuya Jonnel na nagpakilala sa akin sa topic na ito. Habambuhay kong tatanawing utangna-loob ang inyong pagtuturo at tulong. Maraming-maraming salamat po. Maraming salamat rink ay Doc Noel Quiming dahil siya ang palagi kong kinukulit sa data and results verification. Maraming salamat din sa aking editor na si Ms. Bea Samiano na buong tiyagang binasa at inayos ang mga grammar at ambiguity ng aking SP document.

Maraming salamat din sa aking mga naging guro at kamag-aral. Maraming salamat kay Ate Eden dahil palagi kang nandito para sa aming mga BS Comsci undergrads. Maraming salamat sa mga pinakamalalapit kong blockmates – LoAnn, Sosa, Jennifer, Blessie, Simon, Jhesed, Zach, at Mia. Mula second year pa lamang, nagiging masaya na ang araw-araw na lakbayin at aralin dahil kayo ang kasama ko. Masyadong maikli ang papel na ito para sambitin ko lahat ng aking nais sa inyo kaya't hintayin niyo na lang ang blog ko. Salamat rin sa mga "underclassmen" na sumusuporta sa akin, lalo na ang Batch 2011. Marami pa akong dapat pasalamatan gaya ng FBC batchmates ko, Biochem 2010, classmates ko sa Speech11 at PE (SD/MJ) pero kulang talaga itong papel eh, so for the mean time, "THANK YOU ALL" muna!

Bb. Rosemarie Albatana, huwag mong sasabihin na wala kang naitulong sa SP (thesis) ko na ito. Napakalaki ng inspirasyong idinulot mo para simulant at tapusin ko ito. At sa totoo lang, kung wala ang inspirasyong hatid mo, baka hindi ko pa rin nasisimulan itong proyekto na ito. Salamat! Ikaw naman next year, Enginneer! Don't worry, alam mo namang nandito lang naman ako palagi sa iyo. Kaya mo yan! Ikaw pa!

So it all comes down to this. Huli kong pasasalamatan ang dalawang pamilya ko – ang aking LRC family at ang aking biological family – at ang taong may "sala" kung bakit ako ngayon nasa UP. Plano at pangarap ko man na mag-aral sa UP noong 4th year high school, tinanggap ko na noon pa man na hindi kakayanin ng aming pamilya. Mahirap ang buhay. Ayaw ko na itong dagdagan pa. Subalit, si Bb. Arlene Coralde-Magtira, ang aking guro sa Kasaysayan I noong first year high school at SSG adviser, ay gumawa ng paraan para maging possible na makapag-exam ako. Kinausap niya ang Nanay ko. Mahabang kwento ito at alam kong medyo vague ang pagkakasabi ko rito pero sa simpleng salita, siya ang pinakainuutangan ko ng loob sa aking pagpasok sa UP.

Nasa loob na ako ng UP Manila. Mahirap makisama at mamuhay nang halos walang kilala. Oo, inaamin ko na kakaunti lang ang naging kaibigan ko noong unang taon ko. May problema rin kasi ako sa pagsosocialize eh. Pero, everything changed noong natagpuan ko ang office sa second floor ng Joaquin Gonzales Building. Actually, hindi nga office ang nakita ko eh. Nakakita ako ng isang pamilyang nag-aruga at gumabay sa akin mula noong second year ko hanggang ngayon…hanggang bukas. Hindi ko alam kung paano ko kayo pasasalamatan. Habang buhay namamalagi sa aking puso ang lahat ng ala-ala at aral na aking pinagdaanan kasama kayo. Maraming salamat sa aking pamilya dito sa UP Manila. Nanay Tin, Ate Len, Ms. Mae, Aubrey, Sarah, Roselle, Ate Joy, Job, Portia, Viel, Bea, LoAnn, at Sir Harold, maraming salamat! Paulit-ulit ko iyang sasambitin sa inyo. Hayaan niyo, I have the rest of my lifetime to spend with you all naman eh. Asa namang magtatapos dito ang lahat? Nope! You can't get rid of me now that you have me in your lives. Gusto ko lang na palagi niyong maalala na walang hanggan ang aking pasasalamat sa inyo dahil alam kong wala ako ngayon dito kung wala kayo sa buhay ko. Aww cheesy! Hintayin niyo blog ko.

Para sa aking Nanay, Inang at mga kapatid, mag-usap na lang tayo. Ayokong dito sa papel na ito lumabas ang kadramahan natin eh. Basta't alamin niyo na di matatapos ang aking pasasalamat sa suporta na inyong binibigay sa akin. Mahal ko kayong lahat.

At sa lahat ng taong naging bahagi ng buhay ko sa UP Manila, maraming salamat sa inyong lahat!