UPTEC X 10033

Examensarbete 30 hp
Januari 2011

Pharmaceutical knowledge retrieval through reasoning of ChEMBL RDF

Annsofie Andersson

UPPSALA
UNIVERSITET

Bioinformatics Engineering Program

Uppsala University School of Engineering

UPTEC X 10 033

Date of issue 2011-01

Author

Annsofie Andersson

Title (English)

Pharmaceutical knowledge retrieval through reasoning of ChEMBL RDF

Title (Swedish)

Abstract

Development of new pharmaceuticals include numerous of tests and analyses, which unfortunately are both expensive and time consuming. The following study therefore aims to contribute to the area with development of new computer tools that uses the powerful Semantic Web and the ChEMBL database in a user-friendly environment like Bioclipse. In addition, relevant kinase studies utilizing these applications are performed to expand the knowledge in this area and to show the use of the developed tools.

Keywords

Bioclipse, ChEMBL, ligand, MoSS, RDF, Semantic Web, SPARQL, kinase proteins

Supervisors

Egon Willihagen

Scientific reviewer

Jarl Wikberg

Project name	Sponsors
Language **English**	Security
ISSN 1401-2138	Classification
Supplementary bibliographical information	Pages **39**

Biology Education Centre Biomedical Center Husargatan 3 Uppsala
Box 592 S-75124 Uppsala Tel +46 (0)18 4710000 Fax +46 (0)18 471 4687

UPPSALA
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Pharmaceutical knowledge retrieval through reasoning of ChEMBL RDF

Annsofie Andersson

The important research in drug development constantly struggles with finding suitable organic compounds with good properties that can affect pathogenic processes. However many parameters, such as few side effects and low toxicity, are only some of the conditions that must be met to obtain proper and approved drugs. Development of new pharmaceuticals therefore include numerous of tests and analyses, which unfortunately are both expensive and time consuming. The following study therefore aims to contribute to the area with development of new computer tools that uses the powerful Semantic Web and the ChEMBL database in a user-friendly environment like Bioclipse. In addition, relevant kinase studies utilizing these applications are performed to expand the knowledge in this area and to show the use of the developed tools.

Handledare: Egon Willighagen
Ämnesgranskare: Jarl Wikberg
Examinator: Margareta Krabbe
ISSN: 1401-2138, UPTEC X 10033

Pharmaceutical knowledge retrieval through reasoning of ChEMBL RDF

Annsofie Andersson

Sammanfattning

Den betydelsefulla forskningen inom läkemedelsutveckling kämpar ständigt med att finna lämpliga organiska föreningar som med goda egenskaper kan påverka sjukdomsframkallande processer. Genom att detektera molekyler som binder väl till orsakande receptorer med önskade effekter skapar möjligheter att bota, förebygga och lindra sjukdomar. Upptäckter av detta slag är således ytterst åtråvärda och väldigt väsentliga inom bl.a. sjukvården. Dock präglas denna forskning av svårigheter. Många parametrar, såsom t.ex. stark affinitet, få biverkningar och låg toxicitet, är enbart några av de förhållanden som måste vara uppfyllda för att få fram ett välfungerande och godkänt läkemedel. Utveckling av nya mediciner omfattas därför utav noggranna tester och analyser, vilka är både tids- och kostnadskrävande. Detta leder till icke-triviala förhållanden inom läkemedelsforskningen. Följande studie har följaktligen i syfte att bidra med utveckling av datorverktyg och relevanta kinasundersökningar för att utvidga kunnandet inom detta område. Två applikationer har utvecklats i Bioclipse-plattformen tillsammans med ChEMBL-databasen och den semantiska webben. Studien påvisar att potentiella läkemedelskandidater kan hittas tämligen enkelt med de utvecklade applikationerna. De bidragande kinasundersökningar hade som mål att utöka kunskapen om föreningar vilka binder till kinasreceptorer samt att påvisa funktionaliteten hos de nyligen utvecklade metoderna. Detta utfördes med analyser, kallade ”molecular substructure mining”, med hjälp utav den Bioclipse-integrerade applikationen MoSS. MoSS är en programvara för att upptäcka signifikanta substrukturer hos en grupp av utvalda molekyler. Tydligt förekommande fragment har påträffats i undersökningen. Sådana strukturer observerades mellan biologiskt aktiva och inaktiva föreningar bindande till olika kinasfamiljer. I synnerhet detekterades intressanta fragment hos ligander vilka binder till Tyrosin kinaser när jämförelser mot övriga kinasfamiljer utfördes.

Examensarbete 30hp
Civilingenjörsprogrammet Bioinformatik
Uppsala universitet, november 2010

Abstract

The important research in drug development constantly struggles with finding suitable organic compounds with good properties that can affect pathogenic processes. By detecting molecules that bind well to disease-causing receptors with desired effects creates opportunities to cure, prevent and alleviate diseases. Discoveries of this kind are therefore highly desirable and very important in areas such as healthcare. However, this research is characterized by difficulties. Many parameters, such as strong affinity, few side effects and low toxicity, are only some of the conditions that must be met to obtain proper and approved drugs. Development of new drugs therefore include numerous of tests and analyses, which unfortunately are both expensive and time consuming. This leads to non-trivial conditions in the pharmaceutical research. The following study therefore aims to contribute to the area with development of new computer tools and relevant kinase studies to expand the knowledge in this area. Two applications have been developed in the Bioclipse platform together with the ChEMBL database and the Semantic Web. The paper shows that potential drug candidates can be found quite easy with the created applications. The contributing kinase studies had as ambition to give more understanding about compounds that bind to kinase receptors and to show the use of the developed methods. This was carried out with so-called ‘molecular substructure mining’ analysis with the Bioclipse-integrated application MoSS. Moss is a program that discovers significant substructures in a group of chosen molecules. Distinct fragments were found in the research. Frequently occurring fragments were observed between biologically active and inactive compounds binding to various kinase families. Especially, interesting substructures were detected in ligands that primarily bind to Tyrosine kinases when comparisons against the other kinase families were performed.

Table of Contents

Glossary

activity type	A measure deciding the effect a ligand has on the target, for instance IC_{50} or k_i.
affinity	A measurement on how strong a ligand-receptor interaction is. Additionally, described as the equilibrium between the unbound components and the resulting ligand-target complex.
AGC	Kinase family: containing PKA, PKG and PKC kinase families.
API	Application Programming Interface.
assay	Procedure testing the activity of a potential drug.
CAMK	Kinase family: Calcium/calmodulin-dependent protein kinase family.
ChEMBL	Medicinal chemistry database.
ChEBI	Chemical Entities of Biological Interest.
CK1	Kinase family: Casein kinase 1 family.
CMGC	Kinase family: containing CDK, MAPK, GSK3 and CLK kinase families.
Compound/ligand	A small chemical molecule that binds to a receptor.
FTP	File Transfer Protocol.
HTS	High-Throughput Screening.
IC_{50}	Half maximal inhibitory concentration.
IDE	Integrated Development Environment.
k_i	Affinity measure between a receptor and its binding ligand.
OWL	Web Ontology Language.
PCM	Proteochemometrics.
pharmaceutical drug	Contains a chemical substance that acts on and modifies a target to cure or treat a disease.
QSAR	Quantitative Structure-Activity Relationship.
RCP	Rich Client Platform.
RDF	Resource Description Framework.
RDFS	Resource Description Framework Schema.
SAR	Structure-Activity Relationship.
SMILES	Simplified Molecular Input Line Entry System.
SPARQL	SPARQL Protocol and RDF Query Language.
URI	Uniform Resource Identifier.
WWW	World Wide Web.
XML	eXtensible Markup Language.
target	An object in the body a compound is supposed to bind to. This study refers a target to a protein.
TK	Kinase family: Tyrosine kinase family.
TKL	Kinase family: Tyrosine kinase-like family.
STE	Kinase family: Homologs of yeast Sterile 7, Sterile 11 and Sterile 20 kinase families.
W3C	World Wide Web Consortium.

1 Introduction

Background: Drug discovery research is a widespread and highly attractive field. Time-consumption and extreme expenses are unfortunately two limiting factors in the development of functional medicines. Several used techniques to find pharmaceuticals have actually been compared to finding a needle in a haystack, more information is required. A way to gain knowledge about pharmaceuticals is through molecular substructure mining, which focuses on identifying discriminative fragments and regularities in compounds in given areas. Therefore, locating patterns in compounds is a way to retrieve awareness about molecular properties and behavior. For instance implications of an acid environment could easily be drawn if a carboxylic group is frequently occurring in a set of molecules. This generates the possibility to make modifications in activities and hence behavior of potential drugs to make them fit a target good. Molecular substructure mining is thus an important approach to find such atom groups relevant to a molecule's property.

The second major obstacle in drug discovery is the lack of free and easily accessible data with chemical and biological properties of drugs and drug-like molecules. Such information is needed to statistically predict these properties for new potential pharmaceuticals. While Open Data is increasingly making the data freely available, the lack of Open Standards is a remaining issue to be solved. A major tool that changed the essence of bioinformatics was the Internet, and in particular the recently introduced Semantic Web. These technologies provide functionalities like no other tool had done before by: simplifying access and transfer of data, creating fast communications between scientist and online tools and databases. Pharmaceutical research will benefit from an open network of biological and chemical data that is easy accessible and where contributions can be made in a simple manner. Actually, these contributions can be detected in ongoing research with the help from the Semantic Web. The Semantic Web is simply an extension to the WWW that goes beyond the human intellect and further into machine learning. More active help from computers is possible with the use of the Semantic Web as well as easy connections between biological and chemical databases thus enabling an easier platform for sharing and spreading knowledge.

Goal: The goal of this study is to contribute to drug discovery research with the use of Semantic Web techniaques[16] and molecular substructure mining using the Bioclipse-integrated software MoSS[10][13]. Two approaches were taken in order to carry out this task. First, to utilize the Semantic Web standards such as RDF[22] and SPARQL[30] to develop two user-friendly applications in the life science platform Bioclipse[11][12] in purpose of facilitating the search for pharmaceutical information. Second, to use these applications together with MoSS to find suitable candidates for substructure mining in order to discover interesting substructures. As experimental data molecules that binds to kinase proteins will be used to find pattern similarities. Also, Human Computer Interaction (HCI) theories will be applied due to the requirements of well-working workflows advance data analysis have. The following research question was formed for this study –*"How can the Semantic Web and molecular substructure mining contribute to drug discovery research? As benchmark data kinase protein ligands extracted from the ChEMBL database is used.*

Overview: The important preliminaries of the study are introduced in Section 2 including the existing technologies and theories such as the background of drug discovery, the Semantic Web and Bioclipse etc. The combination of existing technologies and the development of novel methods are further given in Section 3. The results from the study are presented in Section 4 while Section 5 analyze, discuss and take a look at the outcome in the field of study. Finally a summarization with conclusions is given in Section 6. Additionally, there is an Appendix with more detailed information containing among others method descriptions and wizard manuals.

2 Background

An introduction of the important preliminaries is given in the following section, including among others the background of drug discovery, Semantic Web and Human Computer Interaction. Open data, Open Source and Open Standards are three central concepts for this study and are also presented in this section. These technologies contain suitable qualities to create applications that can simplify findings of chemical molecules in pharmaceutical development like for instance molecular substructure mining, which is one approach for this paper. The following research question is to be answered *–"How can the Semantic Web and molecular substructure mining contribute to drug discovery research? As benchmark data kinase protein ligands extracted from the ChEMBL database is used.*

2.1 Drug discovery

Production of pharmaceuticals is complex and expensive involving numerous analyses and tests. Drug discovery contain two founding aspects: to locate disease-causative processes i.e. targets and to discover qualitative compounds intervening with a certain target forcing it to act correct again[1][2]. This study primarily concentrates on receptor-ligand interactions. Affected genes are located with techniques like radioligand binding and microarrays etc. Likewise, potential drug candidates could be found with analyses such as High-Throughput Screening (HTS) and structure-based analysis. HTS is one of the most commonly used methods where automatic laboratory techniques screen over thousands of potential candidates investigating how compounds act on targets. Various types of measurements depending on assays are used. This study refers such measures to activity types and the main activity type in focus is an inhibitory concentration, IC_{50}. IC_{50} is a measure on how fast a compound inhibits a biological process by half. Affinity (k_i) is however another form of activity type that measures how strongly a compound binds to a target. A high affinity value suggests a strong compound-target interaction, the higher the better. It is important to distinguish k_i from IC_{50}, even though they are mathematical convertible into each other and are both classified as activity types. Further, as mentioned knowledge of pathogenic receptors is required. With that notice search for compounds could begin which involves both *in vitro* and *in vivo* analysis that hopefully generates a good final product. Candidates with undesired abilities are removed from the research and hopefully a strong candidate is finally ready to reach the market. A summary with conclusions is that drugs are often discovered by accidents like for instance penicillin[3], Flemming (1928). Sir Alexander Flemming discovered Penicillium chrysogenum when he returned from holidays and found that a fungus also known as penicillin had inhibited the growth of the bacterial colonies that he accidently forgotten in the lab. Adding that with expenses and long research periods further analysis and improved methods are highly desirable to create a faster and more understandable research.

2.1.2 Kinase proteins

As mentioned in Section 2.1 a part of drug discovery involves generating understanding about pathogenic processes. A very interesting group of receptors are the kinase proteins that are a part of cell regulation. Kinase proteins regulate cells with phosphorylation causing them to move between active and inactive stages. They are thus part of controlling processes like cell growth and transmission of signals in cells. There are seven known families: Tyrosine kinase (TK), Containing PKA, PKG and PKC kinase families (AGC), Calcium/calmodulin-dependent protein kinase family (CAMK), casein kinase 1 family (CK1), Containing CDK, MAPK, GSK3 and CLK kinase families (CMGC), Tyrosine kinase-like family (TKL) and Homologs of yeast Sterile 7, Sterile 11 and Sterile 20 kinase families (STE). These families are used as benchmark data to locate discriminative fragments and regularities in compounds interacting with kinase proteins. Table 1 shows the chosen activity interval and the number of active and inactive compounds found within that range for each family. Each family was studied separately to find a boarder dividing the molecules in to active and inactive sets. The data was continuously used in the kinase studies performed in this paper.

2.2 ChEMBL and Kinase SARfari

ChEMBL is an Open database[4][5] containing information about bioactive compounds and their interaction with one or many receptors. The data is taken from published articles where experiments similar or like the ones described in Section 2.1 have been performed. The Chemical Entities of Biological Interest (ChEBI)[6] database consists of small chemical compounds and their properties. ChEMBL is now embedded in ChEBI and share important properties like for instance identification of compounds. Thereby it is easy to search for additional information of a compound found in ChEMBL in the ChEBI database. The primary use of ChEMBL exists in Structure- Activity Relation (SAR) where drug discovery is a big aim. Further, a side-project to the ChEMBL database is the SARfari database. The SARfari databases consist of the same type of data as ChEMBL but concentrates on incorporating data originating from certain related gene families. The used Kinase SARfari[7] database thus contains information about ligands interacting with kinase families.

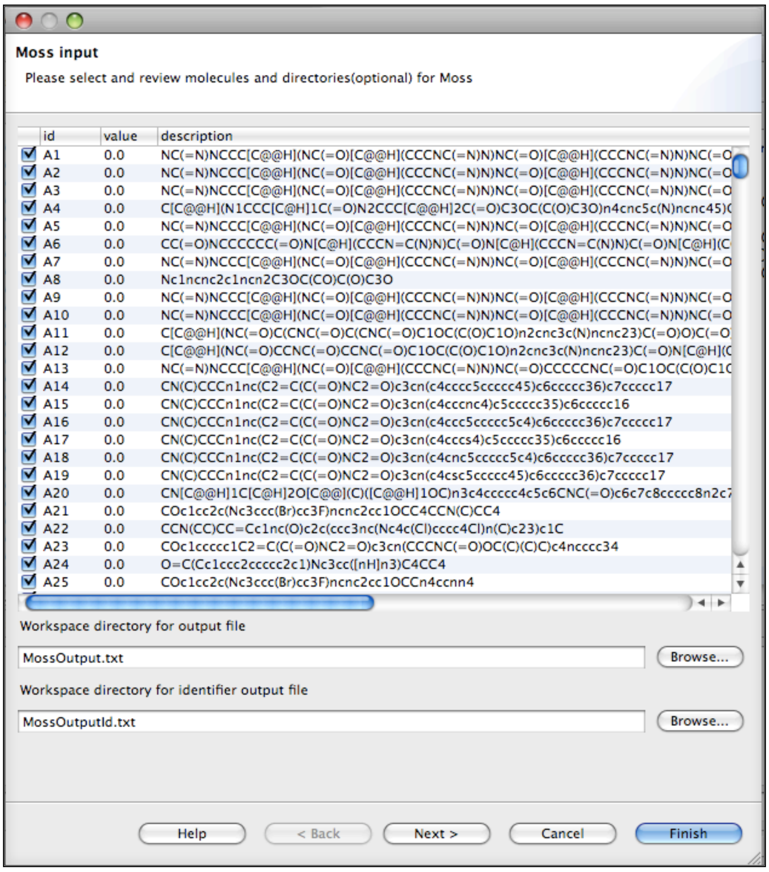
Table 1. The number of active and inactive molecules binding to kinase proteins with IC_{50} in a certain activity interval is shown for each family. The compounds, which are retrieved from the ChEMBL database via the Bioclipse platform, were studied in order to decide the boarder that divides them into active and inactive groups. Below is the number of molecules per category shown. The found molecules are used in the paper as assisting data for programming purpose and more importantly in the molecular substructure mining analyses.

Protein family	Active (#)	Inactive (#)	Activity interval (nmol)
CK1	68	15	1-50000
STE	836	37	1-40000
TKL	1087	101	1-30000
CAMK	2623	55	1-560000
AGC	5729	233	1-1000000
CMGC	9408	592	1-100000
TK	9713	287	1-5460000

Moreover, the ChEMBL database is available online[5] as well as for download[8]. The ChEMBL version used in this study was ChEMBL 02, released in March of 2010. The ChEMBL and Kinase SARfari database were used to obtain information about compounds that could be good candidates in drug development. In particular, compounds interacting with kinase proteins were observed to create awareness about potential important discriminative structures. The benchmark data was very helpful during the design of the wizards but was primarily used in the kinase research.

2.3 Molecular Substructure Mining and MoSS

Molecular substructure mining is a study used to locate regularities within a set of molecules, which usually originates from a given field of interest. Substructure mining finds discriminative fragments between for example active and inactive molecules to locate fragments that are of importance in for instance drug discovery or other areas where small fragments play important roles. Frequently occurring substructures most likely contain important functionalities that could give understandings about binding sites between ligands and receptors or perhaps interesting facts about why some molecules are less desirable. Moreover, Molecular Substructure Miner (MoSS)[9][10] is an Open Source software that locates discriminative substructures in molecule sets.



Moss input

Please select and review molecules and directories(optional) for Moss

	id	value	description
☑	A1	0.0	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCNC(=N)N)NC(=O
☑	A2	0.0	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCNC(=N)N)NC(=O
☑	A3	0.0	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCNC(=N)N)NC(=O
☑	A4	0.0	C[C@@H](N1CCC[C@H]1C(=O)N2CCC[C@@H]2C(=O)C3OC(C(O)C3O)n4cnc5c(N)ncnc45)(
☑	A5	0.0	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCNC(=N)N)NC(=O
☑	A6	0.0	CC(=O)NCCCCCC(=O)N[C@H](CCCN=C(N)N)C(=O)N[C@H](CCCN=C(N)N)C(=O)N[C@H](C
☑	A7	0.0	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCNC(=N)N)NC(=O
☑	A8	0.0	Nc1ncnc2c1ncn2C3OC(CO)C(O)C3O
☑	A9	0.0	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCNC(=N)N)NC(=O
☑	A10	0.0	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCNC(=N)N)NC(=O
☑	A11	0.0	C[C@@H](NC(=O)C(CNC(=O)C(CNC(=O)C1OC(C(O)C1O)n2cnc3c(N)ncnc23)C(=O)O)C(=O)
☑	A12	0.0	C[C@@H](NC(=O)CCNC(=O)CCNC(=O)C1OC(C(O)C1O)n2cnc3c(N)ncnc23)C(=O)N[C@H](C
☑	A13	0.0	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)CCCCCNC(=O)C1OC(C(O)C1C
☑	A14	0.0	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4cccc5ccccc45)c6ccccc36)c7ccccc17
☑	A15	0.0	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4cccnc4)c5ccccc35)c6ccccc16
☑	A16	0.0	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4ccc5ccccc5c4)c6ccccc36)c7ccccc17
☑	A17	0.0	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4cccs4)c5ccccc35)c6ccccc16
☑	A18	0.0	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4cnc5ccccc5c4)c6ccccc36)c7ccccc17
☑	A19	0.0	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4csc5ccccc45)c6ccccc36)c7ccccc17
☑	A20	0.0	CN[C@@H]1C[C@H]2O[C@@](C)([C@@H]1OC)n3c4ccccc4c5c6CNC(=O)c6c7c8ccccc8n2c7
☑	A21	0.0	COc1cc2c(Nc3ccc(Br)cc3F)ncnc2cc1OCC4CCN(C)CC4
☑	A22	0.0	CCN(CC)CC=Cc1nc(O)c2c(ccc3nc(Nc4c(Cl)cccc4Cl)n(C)c23)c1C
☑	A23	0.0	COc1ccccc1C2=C(C(=O)NC2=O)c3cn(CCCNC(=O)OC(C)(C)C)c4ncccc34
☑	A24	0.0	O=C(Cc1ccc2ccccc2c1)Nc3cc([nH]n3)C4CC4
☑	A25	0.0	COc1cc2c(Nc3ccc(Br)cc3F)ncnc2cc1OCCn4ccnn4

Workspace directory for output file

MossOutput.txt

Workspace directory for identifier output file

MossOutputId.txt

Figure 1. A screenshot of the MoSS wizard taken from inside the Bioclipse workspace. The figure displays the first out of four pages in the wizard. Additionally, the picture shows a table containing a dataset of molecules that has been inserted to be further analyzed.

The program originates from a project by Borgelt[10] and was implemented in Bioclipse[11][12] by the author in 2008. The implemented MoSS version was 5.3. MoSS builds on the MoFa[13] and Eclat[14] algorithms that control the parsing of molecules in order to find discriminative fragments. Mining analysis depend on parameters including minimal and maximal support, ring extensions, ignorance of bonds or atom, pruning and so on. Parameters are set by users to optimize and regulate their mining studies. The MoSS application in Bioclipse consists of a wizard with four pages of setting choices. Figure 1 displays the first page in that wizard, which also shows a table of molecules that are to be analyzed.

2.4 The Semantic Web

Instead of using ordinary relational databases the Semantic Web[15][16] was applied because of its powerful way of storing and retrieving information. The following section introduces the background, concepts and standards of the Semantic Web. The used techniques are all Open Standards and recommendations of World Wide Web Consortium (W3C)[17]. Before describing the Semantic Web, understandings about the Internet, syntax and semantics should be clear. Internet was among other things designed to make computers communicate with each other. Most people use Internet through the World Wide Web (WWW), where it is possible to view, store and retrieve information from for example webpage's or databases around the world. Unfortunately computers do not understand the given information from a web document as humans do. Computers mainly understand the syntax behind a document hence a webpage can be shown. However, they do not know how to interpret the meaning of a webpage i.e. understand the semantics. Syntax is the way to write a sentence and can be done in numerous ways. Semantics on the other hand is the meaning behind such a sentence. For example, “I love” and “I ♥ ” have different syntax but the same semantics. Moreover, the Semantic Web initiative solved this deduction by creating techniques assisting computers to interpret syntax from the WWW in order to understand the semantics. To be clear, the Semantic Web is not a completely new web but an extension to the already existing WWW. Consequently, computers can contribute to more active help establishing even more efficient workflows. Also, knowledge can be found with ease since linking between different open databases is possible. Projects in use of the Semantic Web include Dbpedia[18], Bio2RDF[19], chem2Bio2RDF[20] and Semantic Web Health Care and Life Science (HCLS) Interest group[21]. Dbpedia is a knowledgebase consisting of information that has been taken from Wikipedia. Both Bio2RDF and chem2Bio2RDF are frameworks that link various databases together. Bio2RDF primarily focus on linking life-scince information while chem2Bio2RDF connect chemogenomics and other chemical biology data. HCLS on the other hand is a big project aiming to support health care and life science research through the Semantic Web.

2.4.1 RDF

This paragraph introduces the Resource Description Framework (RDF)[22][23], an Open Standard of the Semantic Web. RDF is a metadata for expressing information about objects on the web. In this case an object is a resource in form of a Unique Resource Identifier (URI). RDF models information by structuring it after the existing concepts generating a representation of data that machines can understand. There are different ways to serialize RDF, examples of notation languages are RDF/XML[24], Turtle[25], Notation 3[26] and N-triples[27] etc. Further, information is stored in statements entitled RDF expressions. This term is built on so-called triples: *subject-predicate-object*.

- *Subject*- is the first fraction of a statement that defines the resource that is to be described.
- *Predicate*- is a second resource fraction in a statement that identifies the character of the *subject* and defines the relation between the *subject* and the *object*.
- *Object*- is the last fraction in a statement that either is a resource, blank node or a literal describing what character the *subject* has.

Figure 2 displays a simple RDF statement built with these three concepts. Connecting various triples generates something called RDF graphs, which are extensive networks of many RDF statements. Figure 3 exemplifies a somewhat small RDF graph. The ChEMBL data presented in Section 2.2 was converted into RDF because of its lacking support and is thus built with an extensive RDF graph.

Additionally, RDF does not have an extensive functionality to express triples. A subject can only be described by a predicate with few terms such as being a *type* or *property* etc. If predicates are provided with more detailed identities like being a *class*, *subclass*, *label*, *sameAs*, *differentFrom* and so on more meaningful statements could be formed. As mentioned RDF does not handle such functionality but the two standards; *RDF Schema* (RDFS)[28] and *Web Ontology Language* (OWL)[29] on the other hand do. To explain their power an introduction of two concepts are necessary: *vocabulary* and *ontology*. A *vocabulary* is comparable to a glossary as seen in the Glossary section in the beginning of the paper. A glossary has no direct structure, it mainly provides description to a number of words in a certain domain of interest. *Ontology* on the other hand controls a vocabulary by hierarchically structuring the relationships between the objects thus generating a controlled model of relations consisting of an extensive *vocabulary*. Classification of objects is done with *domains*, which is a region of certain selection of properties. Only words with correct qualifications is found in a given *domain*. *Ontologies* basically define *classes* and *properties* with the use of a hierarchical structure of a vocabulary and domains. The use of broad *vocabularies* and *ontologies* from RDFS and OWL helps generating an extensive and controlled meaning to RDF statements and graphs.

Further, to simplify write and readability in RDF syntax *prefixes* (*rdf*) can be used instead of long URI namespaces (*<http://www.w3.org/1999/02/22-rdf-syntax-ns#/type>*). Commonly used *prefixes* in this study can be seen in Table 2 while Figure 2 shows how such prefixes could be utilized in an RDF statement.

```
PREFIX db: <http://www.dbpedia.org/property>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

db:isbn/978-0-470-19510-9  dc:title  "Drugs from discovery to.."
```

Figure 2. A simple example of an RDF statement about a book is shown. The statement is built up with triples (subject, predicate and object) together with the two prefixes: db and dc.

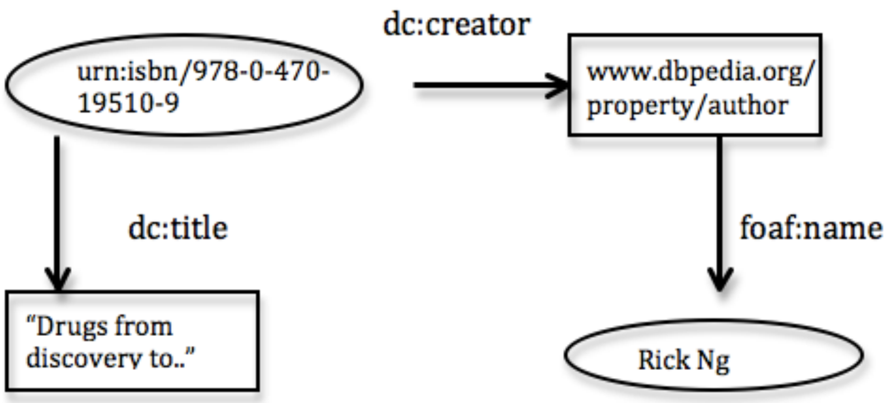

urn:isbn/978-0-470-19510-9
dc:creator
www.dbpedia.org/property/author
dc:title
foaf:name
"Drugs from discovery to.."
Rick Ng

Figure 3. The figure shows a quite small RDF graph over the relationship between a book, its title and author.

Table 2. Commonly used namespaces with corresponding prefixes for this study is shown. A brief description for every namespace is also provided.

Prefix	Namespace	Description
rdf	<http://www.w3.org/1999/02/22-rdf-syntax-ns#>	RDF vocabulary
rdfs	<http://www.w3.org/2000/01/rdf-schema#>	RDF Schema vocabulary
owl	<http://www.w3.org/2002/07/owl#>	OWL vocabulary
dc	<http://purl.org/dc/elements/1.1/>	Persistent Uniform Resource Locators
chembl	<http://rdf.farmbio.uu.se/chembl/onto/#>	ChEMBL vocabulary
bo	<http://www.blueobelisk.org/chemistryblogs/>	Chemistry vocabulary

2.4.2 SPARQL

RDF-ized data can be accessed with query languages as comparable to relational databases and a query language like Structures Query Language (SQL). In this study SPARQL Protocol And RDF Query Language (SPARQL)[30] was used to retrieve data. SPARQL is a powerful query language accessing RDF data independently of notation language, which is an advantage SPARQL has over its predecessors like N3 Query Language (N3QL) and RDF Query (RDFQ). The structure of a simple SPARQL query is shown in Figure 4. Figure 4 demonstrates the usage of prefixes as previously explained in Section 2.4.1. The SELECT line declares what objects that are asked for followed by two query lines stated with triples. Each query line is build up with a *subject*, *predicate* and *object* and as can be seen in Figure 4 not all of them are resources (URI's). One or two fragments of the triples contain a question mark in front of them representing unknown variables: *?title* and *?book*. Variables are unidentified resources that during a SPARQL search will be provided with information.

```
PREFIX db: <http://www.dbpedia.org/property>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
      SELECT ?title WHERE{
             ?book db:ISBN “978-0-470-19510-9”.
             ?book dc:title ?title.
}
```

Figure 4. This code exemplifies a simple SPARQL query requesting the title of a book given a certain ISBN. A joint condition with the variable ?book is used. The ?book variable is utilized to collect the title of the book that will be stored in the variable ?title.

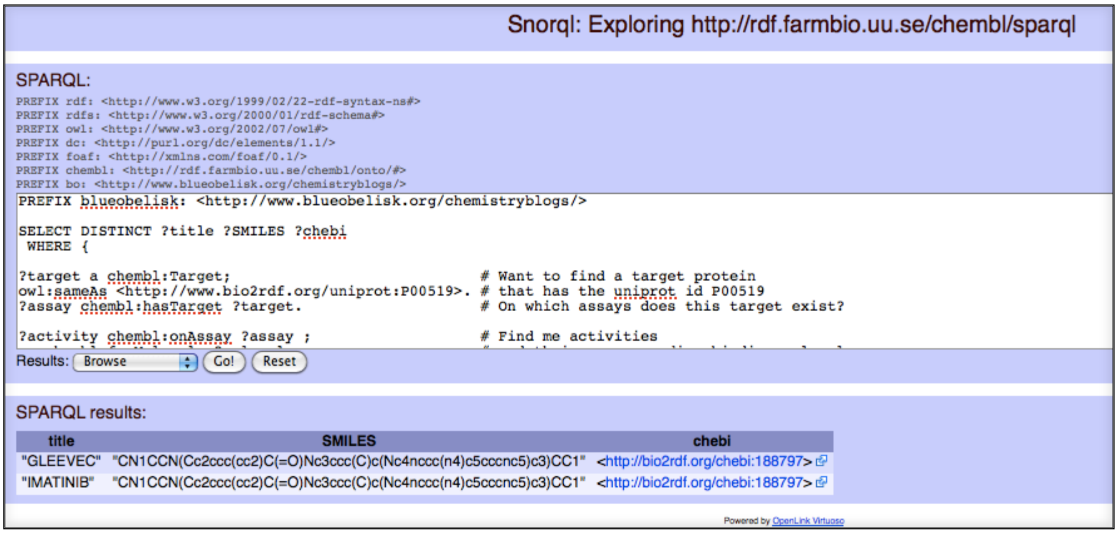
Moreover, the SPARQL engine explore RDF graphs to find accurate data for those variables, which they will store and if asked for also return. Additionally, variables might not have the purpose to be returned but to provide information in other query lines to find the answer for requested variables using so called joint conditions. To access data online interfaces so-called endpoints are used. Figure 5 displays an online SNORQL[31] endpoint where queries are added to a text field and the table beneath shows the results.

2.5 Open Source

This project builds on the principle of using techniques that are open, free and where involvement is straightforward. Besides Open Data and Open Standards that were described in previous sections Open Source projects like Bioclipse, SWT and JFreeChart have also been utilized and are explained in the following section.

2.5.1 Bioclipse

Bioclipse[11][12] is an Open Source life science workbench providing applications for bio- and chemoinformatics. The Bioclipse movement breathes on easy implementation of other Open Source software's consequently creating a platform consisting of multiple functionalities. For instance, editors for chemical compounds, QSAR functions and sequence alignment are only few examples of existing features. Figure 6 displays a screenshot of the Bioclipse workbench. Bioclipse was utilized in this study because the promising platform was suitable for development of drug analyzing tools. Performance of molecular substructure mining and further studies could be carried out with ease in the Bioclipse workbench. With the existing functionalities of Bioclipse both graphical and script components could be developed, which was a perfect way to approach the research problem. Additionally, other Bioclipse applications in the area of life science could be used on the same data creating opportunities to expand a study in any direction.



Snorql: Exploring http://rdf.farmbio.uu.se/chembl/sparql

SPARQL:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX chembl: <http://rdf.farmbio.uu.se/chembl/onto/#>
PREFIX bo: <http://www.blueobelisk.org/chemistryblogs/>
PREFIX blueobelisk: <http://www.blueobelisk.org/chemistryblogs/>

SELECT DISTINCT ?title ?SMILES ?chebi
 WHERE {

?target a chembl:Target;                             # Want to find a target protein
owl:sameAs <http://www.bio2rdf.org/uniprot:P00519>.  # that has the uniprot id P00519
?assay chembl:hasTarget ?target.                     # On which assays does this target exist?

?activity chembl:onAssay ?assay ;                    # Find me activities
```

SPARQL results:

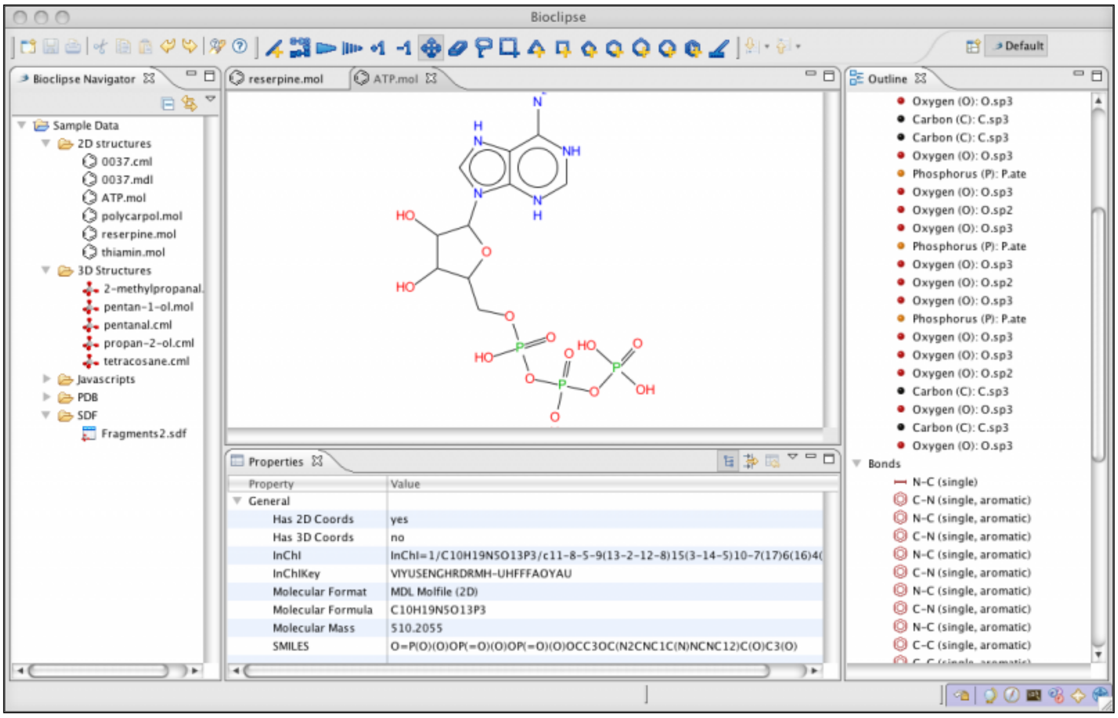
title	SMILES	chebi
"GLEEVEC"	"CN1CCN(Cc2ccc(cc2)C(=O)Nc3ccc(C)c(Nc4nccc(n4)c5cccnc5)c3)CC1"	<http://bio2rdf.org/chebi:188797>
"IMATINIB"	"CN1CCN(Cc2ccc(cc2)C(=O)Nc3ccc(C)c(Nc4nccc(n4)c5cccnc5)c3)CC1"	<http://bio2rdf.org/chebi:188797>

Figure 5. A screenshot of a SNORQL endpoint is shown. SNORQL wraps a default SPARQL endpoint and provides some extra user friendliness, such as predefined prefixes seen at the top of the picture. In the white text field a SPARQL query is written and below the results are shown in a table.

Further, Bioclipse bases its framework on a Rich Client Platform (RCP) provided by the Eclipse foundation. Eclipse[32] is an Open Source Integrated Development Environment, (IDE) with functionality for source code editor, compiler, debugger and building tools. The support for different programming languages in Eclipse is major; Java, C++, Perl and PHP are only few of the language supported. Eclipse has an advanced plug-in architecture and with the extensive RCP functions new software could easily evolve which has been proven to be a concept acceptable by many software developers[33], nonetheless Bioclipse. Moreover, Bioclipse adds an extra layer on top of the RCP influencing the model to adjust the platform in desired ways. Bioclipse contain managers[34] supporting JavaScript and Graphical User Interface environments. The used version of Bioclipse in this study was 2.2.0.v20100127.

2.5.2 SWT and JFreeChart

For people with little computer experience difficulties using JavaScript in Bioclipse is a certainty. This problem was solved with the creation of graphical user interfaces. The Standard Widget ToolKit (SWT)[35][36] is a Java based graphical toolkit that was used to design two wizards. The upholder for SWT is the Eclipse Foundation[32], the same foundation providing the RCP for Bioclipse. SWT provides an API facilitating graphical creation by providing standard layout components including buttons, text fields, layout tools etc. Further, JFreeChart[37] is another Open Source program for Java platforms that was applied in order to extend some graphical work that SWT does not provide. This extensive Java library with tools for formation of charts and plots gave extra graphical techniques to from required histograms. JFreeChart is dependent of yet another Java library JCommon[38] thus both were integrated into the project.



Bioclipse Navigator

- Sample Data
  - 2D structures
    - 0037.cml
    - 0037.mdl
    - ATP.mol
    - polycarpol.mol
    - reserpine.mol
    - thiamin.mol
  - 3D Structures
    - 2-methylpropanal.
    - pentan-1-ol.mol
    - pentanal.cml
    - propan-2-ol.cml
    - tetracosane.cml
  - Javascripts
  - PDB
  - SDF
    - Fragments2.sdf

Properties

Property	Value
General	
Has 2D Coords	yes
Has 3D Coords	no
InChI	InChI=1/C10H19N5O13P3/c11-8-5-9(13-2-12-8)15(3-14-5)10-7(17)6(16)4(
InChIKey	VIYUSENGHRDRMH-UHFFFAOYAU
Molecular Format	MDL Molfile (2D)
Molecular Formula	C10H19N5O13P3
Molecular Mass	510.2055
SMILES	O=P(O)(O)OP(=O)(O)OP(=O)(O)OCC3OC(N2CNC1C(N)NCNC12)C(O)C3(O)

Outline

- ● Oxygen (O): O.sp3
- ● Carbon (C): C.sp3
- ● Carbon (C): C.sp3
- ● Oxygen (O): O.sp3
- ● Phosphorus (P): P.ate
- ● Oxygen (O): O.sp3
- ● Oxygen (O): O.sp2
- ● Oxygen (O): O.sp3
- ● Phosphorus (P): P.ate
- ● Oxygen (O): O.sp3
- ● Oxygen (O): O.sp2
- ● Oxygen (O): O.sp3
- ● Phosphorus (P): P.ate
- ● Oxygen (O): O.sp3
- ● Oxygen (O): O.sp3
- ● Oxygen (O): O.sp2
- ● Carbon (C): C.sp3
- ● Oxygen (O): O.sp3
- ● Carbon (C): C.sp3
- ● Oxygen (O): O.sp3

Bonds

- N-C (single)
- C-N (single, aromatic)
- N-C (single, aromatic)
- C-N (single, aromatic)
- N-C (single, aromatic)
- C-N (single, aromatic)
- N-C (single, aromatic)
- C-N (single, aromatic)
- N-C (single, aromatic)
- C-C (single, aromatic)

Figure 6. The picture displays a screenshot of the Bioclipse workspace. Four active work views can be seen in the figure.

2.6 Human Computer Interaction

Advance data analysis requires well-reasoned applications to control formation of massive sets of information. A well-studied area to perform formations of this kind is the field of human computer interaction. Several principles from this approach have been used to create stable and useful interaction tools.

The *MoSCoW*[39][40] rules are a combination of four rules to guide developers through out the work phase. The *MoSCoW* principle is a technique of the Dynamic System Development method (DSDM)[41] and is highly useful in the design of systems that specially are under time and cost constraints. The *MoSCoW* rules aim to build up systems with priority of requirements by simply follow these easy rules:

- Must have
- Should have
- Could have
- Want to have but will not have time this round

The four rules are quite explanatory by name. ‘Must have’ simply focus on system requirements to make the program function. ‘Should have’ involves things that should be included to make the system as usable as possible, though not crucial for the program. ‘Could have’ includes functions that could be part of the system to add further functionalities, for instance a keyboard shortcut to an already existing button. ‘Want to have but will not have time this round’ are features that are not as important as other and could be added in the next release if time constraints exist.

Research to improve interactive systems grew larger as computers became more involved in our work. Norman (1998) and Nielsen (1993) developed numerous of principles that could and should be noticed in interactive system design. Some of those principles[40] are adapted in this study including: visibility, consistency, familiarity, affordability, navigation, control, feedback, recovery, constraints, flexibility, style and conviviality. Further, the human brain is a very complex and interesting organ that is important to acknowledge during the design phase. The brain interprets all given information and does it differently depending on how the information is represented. Acknowledgement of *cognitive psychology*[42] is thus very helpful in the design stages, especially where complex analysis exist. First thing to mention is the human memory. Our memory contains a long-term and a working (short-term) memory. The long-term memory is the storage for our long-time memories with non-limited space. However, the working memory is only able to hold information for maximum 30 seconds. To maintain information in the working memory continuous repetition is required. Eventually information will be stored in the long-term memory but that is a process that usually takes time. Clearly, when it comes to advance data analysis it is important to visualize information for as long as the end-user requires.

Secondly, the Gestalt laws of perception[40] should be considered when designing the layout of a system. Proximity, continuity, part-whole relationships, similarity and closure are the five main concepts to recognize. If graphical interfaces follow these simple laws infinite information could be displayed simultaneously and humans would still be able to interpret it.

- Proximity describes how objects appear with respect to each other and how humans organize them after appearance.
- Continuity means that the human understanding of smooth continuous patterns is greater than broken and disturbed designs.
- Part-whole relationships explain that it is easier to see an object as whole rather than sub-parts of that object.
- Similarity explains how human group objects that are more similar to each other than none-similar.
- Closure describes that the perceiving of closed figures is easier than unclosed.

Thirdly, system developers always intend to create an environment where users feel comfortable and secure. This is unfortunately not always fulfilled if the software is complex and unfamiliar to the user. In human computer interaction two terms are frequently used: recall and recognition. Recognition is the process where humans remember a specific piece of information when for instance observing an object or listening to a sound. Recall on the other hand is an active search performed to find specific information. It is often easier to recognize than to recall objects. Basing applications on recognition thus generates more comfortable and faster workflows. Applying familiar graphical components including buttons, checkboxes, icons etc is a way to adapt recognition. Also, prior to this study components and styles of the Bioclipse workspace should be followed to follow the recognition of the platform hence SWT (Section 2.5.2) was utilized. Command-line systems are recall-based interfaces that accordingly to Mandler (1980) are hard for inexperienced users to adjust to while graphical interfaces that are recognition-based create frustrations within experienced users. Optimal systems should thereby include both types of interfaces. Moreover, with navigation and restrictions throughout the work leaves small possibilities for error making. This establishes more security for users. If errors would occur well formed messages and guidance are extremely important.

Lastly, to generate a system with great usability it is required to acknowledge human disabilities and different computer experiences. For instance precaution with colors, sizes, advance formations are examples of factors that could be taken in order to create a user-friendly and human-adapted environment. Additionally, experts in an area are of high importance and should be a part of the development continuously. Expert opinions are important in system design when it comes to formation of user-adapted systems. Including specialists during the system development will form the program after the needs that exist in the field creating efficient work situations. Applying given theories of HCI in advance data analysis, which is highly required due to the complex situations that exist in these areas, simplifies and creates good structure in the work.

3 Design and Implementation

Design and implementation of methods and techniques are covered in this section. To design novel and extremely required methods the presented technologies in Section 2 has been combined in order to manage this. This includes among others the important integration of the Semantic Web and the novel ChEMBL plug-in (explained in the following section). The purpose of these methods is to facilitate the work of finding chemical compounds that could be further used in analysis contributing to the research field of drug discovery such as molecular substructure mining. The research question for the paper was –*"How can the Semantic Web and molecular substructure mining contribute to drug discovery research? -As benchmark data kinase protein ligands extracted from the ChEMBL database is used.*

3.1 Extracting and retrieving available data

The Open Standards RDF, Section 2.4.1 and SPARQL, Section 2.4.2 were utilized in all searches against the ChEMBL database. As SPARQL is fully dependent on RDF the Open Data in the database could not be reached because of its none existing support for RDF. The ChEMBL database was thus downloaded via the FTP server[8] of the EBI group and rewritten into a supporting RDF format. The data can freely be reached from the following endpoints [http://rdf.farmbio.uu.se/chembl/sparql/] and [http://rdf.farmbio.uu.se/chembl/snorlq/]. Moreover, knowledge about classes, properties and their relationships (RDF graphs) in the ChEMBL data was retrieved with the use of SPARQL queries, such a query is shown in Figure 7. Queries of this kind were made for all relevant classes and properties in order to complete the understanding of the structure of the RDF-ized ChEMBL database. Queries in purpose of retrieving relevant data in relation to this study were carefully constructed evolving from general to absolute specific. Examples of two queries that collect significant information are seen in Figure 8 and Figure 9. Due to numerous creations of queries no further presentation is done in this paper. The most relevant and kept queries can be studied at the GitHub repository [http://github.com/annzi/bioclipse.chembl].

```
SELECT DISTINCT ?property ?hasValue ?isValueOf WHERE {
       { <http://rdf.farmbio.uu.se/chembl/molecule/m545827>
            ?property   ?hasValue. }
      UNION
      { ?isValueOf      ?property
            <http://rdf.farmbio.uu.se/chembl/molecule/m545827>.}
}
ORDER BY ?property ?hasValue ?isValueOf
```

Figure 7. The code shows a SPARQL query in the search for unknown variables in the RDF graph of ChEMBL. In this case exploration of the class ChEMBL:Compound is performed. A randomly selected molecule *m545827* act as a reference molecule to locate present properties for the given class.

```
SELECT DISTINCT ?target ?description  WHERE {
      ?target a chembl:Target;
            chembl:hasDescription ?description .
      FILTER regex(?description, "Sodium", "i") .
}
```

Figure 8. **An example of a SPARQL query attempting to find targets related to a keyword is shown.** The keyword in this case is "sodium" as can be seen in the filtration function. Both target id and the found description containing the keyword is returned.

```
SELECT DISTINCT ?smiles ?actval WHERE{
      ?target a chembl:Target.
      ?assay chembl:hasTarget ?target .
      ?activity chembl:onAssay ?assay ;
               chembl:type ?actType ;
               chembl:forMolecule ?mol .
      ?mol bo:smiles ?smiles.
      FILTER regex(?fam,"^TK$", "i").
      FILTER regex(?actType, "^IC50 $", ″i").
}
```

Figure 9. An example of a SPARQL query locating compounds that influences a specific kinase protein is exemplified. The presented query seeks compounds that bind to proteins in the TK family with the activity type IC_{50}. An assay has to be preformed in order to determine the activity of the identified molecules.

3.2 Platform integration

Bioclipse stands as a platform for the integration of techniques presented in Section 2 such as the RDF-ized ChEMBL database, SPARQL queries and MoSS. Because of the highly developed Bioclipse project preexisting functionalities could be adopted into this study and are described in the upcoming paragraphs. Implementation of existing technologies formed a new plug-in with new functions corresponding to work around the ChEMBL database and as seen later also around the MoSS application.

3.2.1 Qualities of Bioclipse

Movement from online efforts (Virtuoso endpoint) to the modern life science workbench Bioclipse opened up new possibilities. For example SPARQL queries now had the possibility to be utilized without the necessity to learn the language. Consequently, no evolvement of own-made queries for end-users was essential anymore. Secondly, query modifications is be possible with a simple click in an interface that normally would involve modifications in SPARQL syntax. Thereby, efficient searches to retrieve interesting molecules in drug discovery related researches were enabled for users independent of computer experience.

Further, Bioclipse as mentioned in Section 2.5.1 is an Open Source framework that integrates existing software into its platform. Available applications were taken advantage of in order to expand the Bioclipse module with new integrations. Example of this is the use of *Jena*[43]. Jena is an Open Source Java framework for the Semantic Web. It offers an Application Programming Interface (API) for reading and writing data into RDF graphs with notations like RDF/XML and N-Triples. But more important for this study, Jena provides a SPARQL engine. This enables a SPARQL function inside the Bioclipse workspace, which is highly important for the usability. The *Jena* library was used through the existing rdf plug-in [http://github.com/bioclipse/bioclipse.rdf] to enable querying against the set-up RDF knowledgebase with SPARQL. The *rdf.sparqlRemote(URL, query)* method was used.

- *rdf.sparqlRemote(URL, query)* have two input arguments an URL with the address to a remote RDF database and a string containing a SPARQL query. The method accesses the database bringing the query and later returning the answers where the result is put into a matrix. An example of a way to use the *rdf.sparqlRemote* is given in Figure 10.

3.2.2 The ChEMBL plug-in

A new plug-in in Bioclipse has been developed. The ChEMBL plug-in primarily provides functionality for querying the remote RDF-ized ChEMBL database. Moreover, several Java methods for the mentioned cause and for few other necessities have been developed. A list over developed methods and their function is presented in Appendix 1. The method format is controlled by input arguments from end-users. An input is inserted into a hardcoded query where the pre-made question act as an input argument to the *rdf.sparqlRemote* method, actually in the exact same way as seen in Figure 10 but with different syntax. Additionally, Figure 11 shows a Java method invoking the *rdf.sparqRemote* method with yet another syntax.

The ChEMBL plug-in was developed with Bioclipse standards meaning that it among others contain managers[34] to support JavaScript environment. All methods in the ChEMBL plug-in are thus reachable from a JavaScript console. To create an environment suitable for novice users graphical user interfaces were developed and are presented in Section 4.1 and Section 4.2. Furthermore, to distinguish between different kinds of search perspectives two separate features have been produced: ChEMBL-MoSS and ChEMBL. The ChEMBL-MoSS feature was developed to include support for substructure mining while the ChEMBL feature aim on more general searches against the RDF-ized ChEMBL database.

```
var sparql = "\
SELECT DISTINCT ?target ?key WHERE {\
      ?target a chembl:Target;\
              chembl:hasDescription ?key .\
      FILTER regex(?key , \"Sodium\" ,\"i\") }";

rdf.sparqlRemote("http://rdf.farmbio.uu.se/chembl/sparql",sparql)
```

Figure 10. This code shows the utilization of the method *rdf.sparqRemote*. The method is used in a JavaScript environment in Bioclipse. Notice, the query is the same as given in Figure 8 but written with different syntax.

```
public IStringMatrix getTargetIDWithKeyword(String keyword)
  throws BioclipseException{
      String sparql=
            "SELECT DISTINCT ?target ?key  WHERE {" +
                  " ?target a chembl:Target." +
                  " ?target chembl:hasDescription ?key ." +
                  " FILTER regex(?key,"+"\"("+keyword+")\",\"i\") ."+
                  "}";
IStringMatrix matrix = rdf.sparqlRemote(CHEMBL_SPARQL_ENDPOINT,
sparql);
            cutter(matrix);
            return matrix;}
```

Figure 11. A java method inserting an input argument into a pre-made query and further utilizing *rdf.sparqlRemote* is displayed. The query is the same as presented in Figure 8 and Figure 10 but written in Java. The cutter method removes unwanted URI’s keeping only the desired information.

3.2.3 Molecular Substructure Mining

Molecular substructure mining as described in Section 2.3 was performed on molecular datasets containing ligands binding to proteins from kinase families. Two different studies were carried out: first to find discriminative fragments between active and inactive compounds acting on proteins derived from one kinase family for each of the seven relatives. Secondly, to find patterns between compounds interacting with proteins originating from the largest family namely the TK family in a study where comparisons against the remaining kinase families are done. To begin with, the existing moss plug-in described in Section 2.3 had to be modernized because of drastic changes in Bioclipse over the years. A new plug-in supporting manager environment has thus been developed. Both new and existing Java code was integrated in order to get this plug-in back on its feet. Further, the mining analyses were carried out with the following workflow: molecules from the remote RDF-ized ChEMBL database were retrieved with the use of the novel ChEMBL-MoSS feature and then utilized in the upgraded moss plug-in. Selected datasets can be observed in Table 1. Both the ChEMBL-MoSS wizard (the resulting ChEMBL-MoSS wizard from Section 4.2) displayed in Figure 15 and JavaScript seen in Figure 12 were used to extract desired molecules. To clarify, both environments perform the same kind of tasks but to visualize that this work situation can be performed in more than one way both were applied. From a HCI perspective this broadens the usability for inexperienced as well as experienced users as explained in Section 2.6. Finally, substructure mining on retrieved molecules were performed with the existing MoSS wizard displayed in Figure 1 as well as with the JavaScript shown in Figure 12. Again both procedures were used only to demonstrate the various ways to perform this task.

4 Results

Concepts and standards taken from Section 2 such as the Semantic Web and Human Computer Interaction (HCI) were used during the design and implementation phases described in Section 3. The following section presents the results from this integration including two novel wizards and the outcome from the substructure mining analyses. The aim was to answer the following research question –*"How can the Semantic Web and molecular substructure mining contribute to drug discovery research? -As benchmark data kinase protein ligands extracted from the ChEMBL database is used.* Additionally, the developed ChEMBL plug-in and its methods have already been described in Section 3.2.1 and 3.2.1 and further details are found in Appendix 1.

4.1 The ChEMBL Wizard

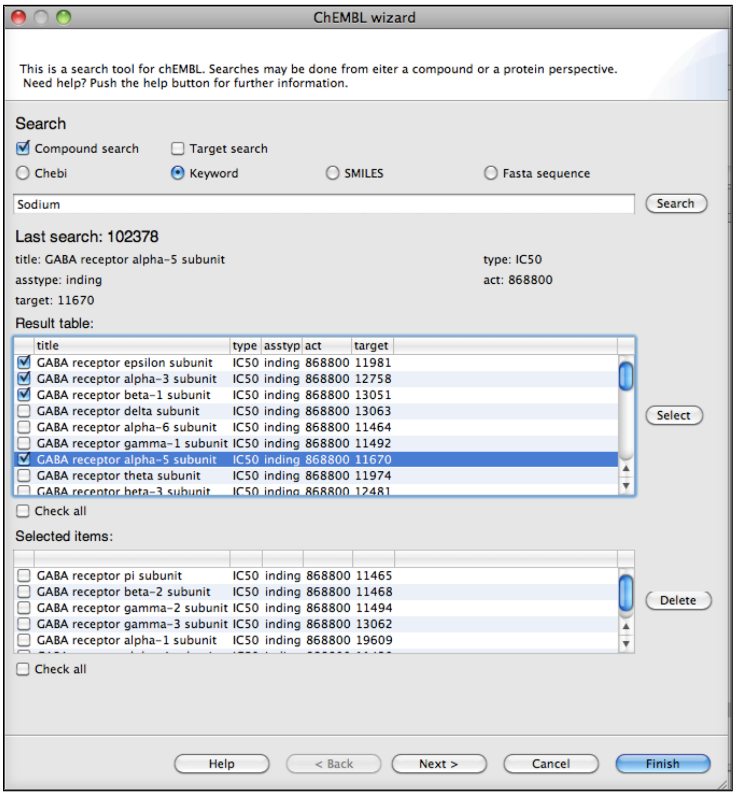
The ambition for the study was among others to improve the discovery of potential pharmaceutical thus two computer tools have been designed. This paragraph presents the first out of two developed applications, namely the ChEMBL wizard. The aim for the ChEMBL wizard is to enable easy search of information involving ligand-target interactions. Therefore a search engine querying the remote RDF-ized ChEMBL database mentioned in Section 3.1 has been developed into the Bioclipse workspace. The ChEMBL wizard is one out of two features in the ChEMBL plug-in and has been developed with the combination of Java, SWT, SPARQL and Bioclipse. The first page in the interface is shown in Figure 13.

```
>  var  v=chembl.mossGetCompoundsFromProteinFamilyWithActivity("AGC",
"IC50 ")
> v.getRowCount()
5528
> chembl.moSSViewHistogram(v) //Histogram shows
> var v:2 =chembl.mossSetActivityBound(v, 1, 1000000)

> var v3=
chembl.mossGetCompoundsFromProteinFamilyWithActivityBound("CAMK",
"IC50 ", 1, 1000000)

> v3.getRowCount()
2565
> chembl.saveToMossFormat("/ChEMBL-MoSS/TEST/AGC-CAMK", v:2, v3)

> var mossObject = moss.createMoSSProperties()

> mossObject
{"path":null,"mode":10273,"seed":"","threshold":0.5,"minimalSupport":
10.0,"maximalSupport":2.0,"exNode":"H","exSeed":"","minRing":0,"maxRi
ng":0,"maxEmbMemory":0,"mbond":31,"mrgbd":31,"matom":127,"mrgat":127,
"ignoreBond":"never","matchChargeOfAtoms":"nomatch","aromatic":"never
","ignoreTypeOfAtoms":"never","ringExtension":"none","kekule":true,"c
arbonChainLength":false,"extPrune":"none","canonic":true,"equiv":
false,"unembedSibling":false,"closed":true,"split":false,"maxEmbed":0
,"minEmbed":1,"pathId":null,"namefile":null,"namefileId":null,"matchA
romaticityAtoms":"nomatch"}

> mossObject.setMinimalSupport(5.0)
> mossObject.setMaximalSupport(1.0)

> mossObject.setIgnoreTypeOfAtoms("always”)

> moss.run("/ChEMBL-MoSS/TEST/AGC-CAMK",("/ChEMBL-MoSS/TEST/AGC-
           CAMKOUT", ”/ChEMBL-MoSS/TEST/AGC-CAMKINDEX", mossObject)
```

Figure 12. **A JavaScript showing the simple workflow of substructure mining studies.** Compounds interacting with proteins in the AGC and CAMK family with the activity type IC_{50} are collected in two different ways. All compounds from the AGC family that are specified with IC_{50} are collected (5528). A histogram is called and displays the activities in a plot. This makes the next step possible where activity boarders are set. For the CAMK family the activity bounds were already known from the beginning due to previous runs and thus the compounds for those parameters could be searched for with only one call. Further, a mossObject is created with default settings where two of them are modified. Adding input, output, outputIndex files together with the mossObject to the moss.run method starts and performs the mining.

The second and last page of the wizard is displayed in Figure 14 but notice that the screenshot is taken from an identical page from the ChEMBL-MoSS wizard see Section 4.2. Further, two search perspectives have been set up on the first page. Compound search is the first perspective including three different search categories: *chebi id*, *keyword* and *SMILES*. These searches return information about molecules and their relation to targets, activity and so on. Target search, however is the second perspective including four categories: *target id*, *keyword*, *ec-number* (classification scheme for enzymes) and *FASTA sequence* (format for peptide sequences). These searches locate molecules interacting with specified targets and also return other appropriate properties like for instance UniProt identification. The two boxes shown at the top of the wizard in Figure 13 represent the two search perspectives while the radio boxes below denote various categories. The category titles transform into accurate labels as the perspective mode is changed. The wizard in Figure 13 shows titles when a compound search mode is enabled. Moreover, adding text to a *text field* and pushing the search button generates a search. A SPARQL query method in the plug-in is awoken. It initializes a pre-made SPARQL query and accesses the remote database as described in Section 3.2.1. Then it returns the results to fill the *upper table*. Occasionally there are searches with hits over 10000. The Virtuoso endpoint however does not return more than 10000 hence some results are unfortunately cut-off. Further, the *table below* on the contrary has the purpose of storing selected items to enable various searches without loosing desired information. There are two ways to select an item or items, which are drag and drop and checking item(s) and pushing the *select button*. Deletion of an item in the *lower table* is done in somewhat the same way; checking undesired item(s) and pushing the *delete button*. The middle part of the wizard contains feedback features where information is shown to fulfill HCI standards. The *last search label* simply displays the last performed search and the field below gives feedback about the selected item in the result table. The field is dynamic thus returns information depending on selections and actions. The *next button* generates a second page with a file browser for name and path selection as similar to the one shown in Figure 14.

Furthermore, error constraints are controlled with concepts taken from HCI as seen in Section 2.6 to generate a simple workflow. Not all error scenarios could be eliminated thus an important part was to develop an information system that informs users about existing errors with messages and restrictions.



ChEMBL wizard

This is a search tool for chEMBL. Searches may be done from eiter a compound or a protein perspective.
Need help? Push the help button for further information.

Search

☑ Compound search □ Target search

○ Chebi ● Keyword ○ SMILES ○ Fasta sequence

Sodium

Last search: 102378

title: GABA receptor alpha-5 subunit
type: IC50
asstype: inding
act: 868800
target: 11670

Result table:

	title	type	asstyp	act	target
☑	GABA receptor epsilon subunit	IC50	inding	868800	11981
☑	GABA receptor alpha-3 subunit	IC50	inding	868800	12758
☑	GABA receptor beta-1 subunit	IC50	inding	868800	13051
□	GABA receptor delta subunit	IC50	inding	868800	13063
□	GABA receptor alpha-6 subunit	IC50	inding	868800	11464
□	GABA receptor gamma-1 subunit	IC50	inding	868800	11492
☑	GABA receptor alpha-5 subunit	IC50	inding	868800	11670
□	GABA receptor theta subunit	IC50	inding	868800	11974
□	GABA receptor beta-3 subunit	IC50	inding	868800	12481

□ Check all

Selected items:

□	GABA receptor pi subunit	IC50	inding	868800	11465
□	GABA receptor beta-2 subunit	IC50	inding	868800	11468
□	GABA receptor gamma-2 subunit	IC50	inding	868800	11494
□	GABA receptor gamma-3 subunit	IC50	inding	868800	13062
□	GABA receptor alpha-1 subunit	IC50	inding	868800	19609

□ Check all

Figure 13. **The ChEMBL wizard.** The picture displays the first page in the wizard where a keyword search has been preformed with the word "sodium". Several results are chosen in the top table while results from previously made searches are seen in the bottom table.

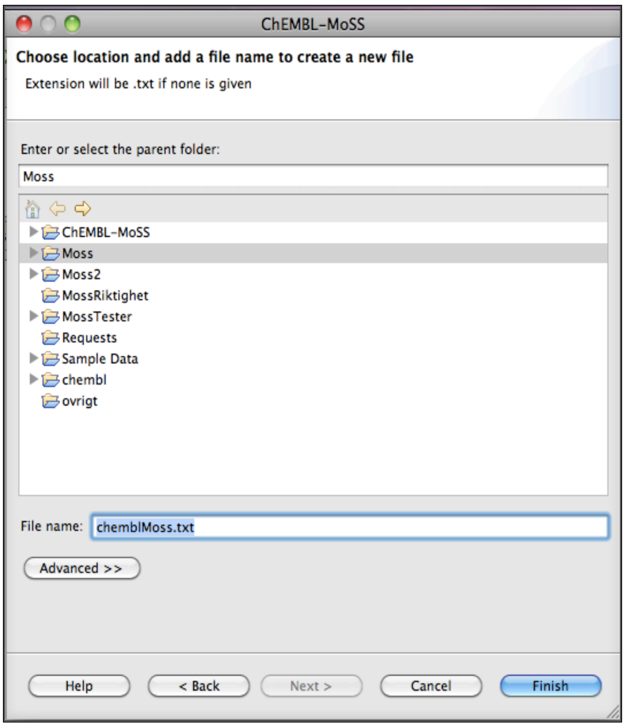
Error messages appear in various cases, for instance when a search has been made with an empty text field, see message (a). If a search has been made with the wrong kind of character for a certain category (b). If a table object has not been selected and the *select button* has been pushed (c). Additionally, messages do not always have to appear because of errors, important messages in purpose of enlightenment also exist as seen in (d) where a search did not get any results despite correct input format.

a. Empty text field.
b. No hits were found for this search. The search was made with a string instead of a chebi id perhaps a keyword search is a more appropriate.
c. No item selected.
d. No hits were found for this search.

An error message simply disappears when the error has been fixed. Messages are shown in the top field of the wizard. As long as the *selected items table* is empty there is no need for the user to move forward in the wizard hence the *next* and *finish buttons* are disabled. When *cancel* is chosen the user is asked for confirmation so that there are no loss of information. Additionally, Appendix 4 contains a step-by-step manual for the ChEMBL wizard.

4.2 The ChEMBL-MoSS Wizard

The second developed application is covered in the following section. The aim for the ChEMBL-MoSS wizard is to facilitate the gathering of molecules for upcoming substructure mining analyses as presented in Section 3.2.3.



Choose location and add a file name to create a new file

Extension will be .txt if none is given

Enter or select the parent folder: Moss

File name: chemblMoss.txt

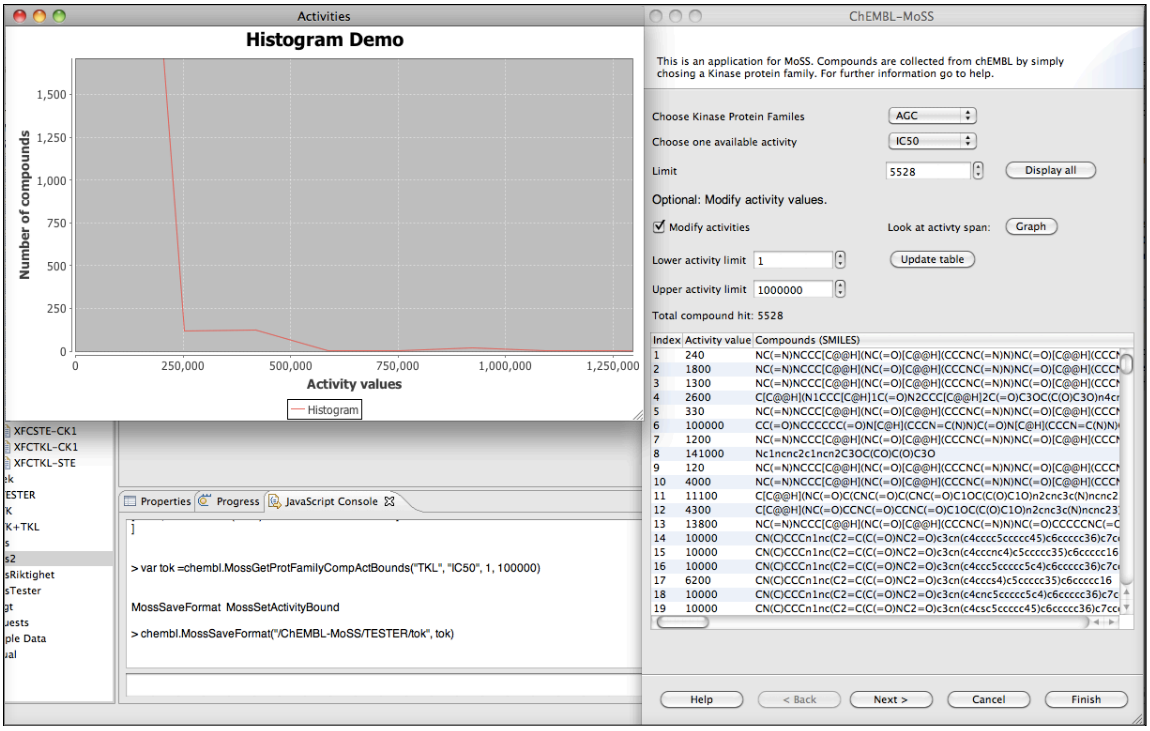
Figure 14. The picture shows the browser page of the ChEMBL-MoSS wizard. To be noticed, the browser function in the ChEMBL wizard has the same appearance and function as the one presented here. Naming of files and directory can be set and if the filename is taken an error is returned thus no overwriting is possible.

The application builds on the same concept as the ChEMBL wizard with focus on enabling fast searches of relevant information. The big difference is that this tool has been developed to actually facilitate a certain form of analysis i.e. substructure mining. The ChEMBL-MoSS wizard has made it possible to retrieve active and inactive compounds binding to a certain category of protein families with a specific activity type. Additionally, such analysis is also manageable between two families where one family is set to be in focus and the other to be the complement part, hence the second provided wizard page. The ChEMBL-MoSS feature is a part of the ChEMBL plug-in explained in Section 3.2.2 and uses Java, SWT, Bioclipse and the Semantic Web.

Moving on, the rightmost window in Figure 15 displays the ChEMBL-MoSS wizard. At the top of the wizard three *combo boxes* are situated but only the first one is accessible when the wizard opens. This *box* generates the workflow for the wizard containing a list of hardcoded kinase families. When a family has been chosen a SPARQL query method from the ChEMBL plug-in is generated accessing the remote database and returning available activity types for that family. It also activates the *combo box* below by brining the results into it. A selected activity type generates yet another query returning compounds involved with proteins from the chosen family and their activity values. The result is then displayed in the *table*. With respect to search time a limit has been set to 50 generating the next and final *combo box*: regulation of the limit. Also, the *display all button* was produced to trigger the same query without any limitation causing the *table* to display all molecules existing in the knowledgebase for that given query.

Further, the next part of the wizard contributes to data modification. By checking *modify activity* a new set of functionalities appear for the user. The *graph button* displays activity values for compounds in form of a *histogram*, generating a visualization overview for this cause. The leftmost window in Figure 15 shows such a *histogram*. Active molecules can be distinguished from inactive ones when knowledge about the activity interval is known. Thereby a histogram plays an important role in this step. The last two combo *boxes* on the page provide features for setting *lower* and *upper activity values* for the interval while the *update table button* updates the table with compounds within that area.

A second, identical page was provided to give the possibility of selecting a second dataset. This allows users to set up a between-datasets for discriminative fragment experiments. If no selection has been made on this page only the selections from the first page will be saved and used. The third and last page shown in Figure 14 allows the user to select a path and name for the MoSS file. Two additional files are automatically saved: one providing the utilized SPARQL query (file.sparql) for learning purpose and the other information about the relations between compounds, activities and targets (fileInformation.txt). Additionally, error handling is minimal for this interface because of the construction of the wizard. The wizard was simply developed to navigate users through the workflow in a more restricted way than the ChEMBL wizard. Steering the user was done with the help of enabling boxes along the workflow; one box simply triggers another.



```
]

> var tok =chembl.MossGetProtFamilyCompActBounds("TKL", "IC50", 1, 100000)

MossSaveFormat  MossSetActivityBound

> chembl.MossSaveFormat("/ChEMBL-MoSS/TESTER/tok", tok)
```

This is an application for MoSS. Compounds are collected from chEMBL by simply chosing a Kinase protein family. For further information go to help.

Choose Kinase Protein Familes: AGC

Choose one available activity: IC50

Limit: 5528

Optional: Modify activity values.

☑ Modify activities

Look at activty span:

Lower activity limit: 1

Upper activity limit: 1000000

Total compound hit: 5528

Index	Activity value	Compounds (SMILES)
1	240	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCN
2	1800	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCN
3	1300	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCN
4	2600	C[C@@H](N1CCC[C@H]1C(=O)N2CCC[C@@H]2C(=O)C3OC(C(O)C3O)n4cr
5	330	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCN
6	100000	CC(=O)NCCCCCC(=O)N[C@H](CCCN=C(N)N)C(=O)N[C@H](CCCN=C(N)N)
7	1200	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCN
8	141000	Nc1ncnc2c1ncn2C3OC(CO)C(O)C3O
9	120	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCN
10	4000	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)[C@@H](CCCN
11	11100	C[C@@H](NC(=O)C(CNC(=O)C(CNC(=O)C1OC(C(O)C1O)n2cnc3c(N)ncnc2
12	4300	C[C@@H](NC(=O)CCNC(=O)CCNC(=O)C1OC(C(O)C1O)n2cnc3c(N)ncnc23
13	13800	NC(=N)NCCC[C@@H](NC(=O)[C@@H](CCCNC(=N)N)NC(=O)CCCCCNC(=C
14	10000	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4cccc5ccccc45)c6ccccc36)c7c
15	10000	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4cccnc4)c5ccccc35)c6ccccc16
16	10000	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4ccc5ccccc5c4)c6ccccc36)c7c
17	6200	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4cccs4)c5ccccc35)c6ccccc16
18	10000	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4cnc5ccccc5c4)c6ccccc36)c7c
19	10000	CN(C)CCCn1nc(C2=C(C(=O)NC2=O)c3cn(c4csc5ccccc45)c6ccccc36)c7cc

Figure 15. **The ChEMBL-MoSS wizard.** The leftmost window displays a histogram of an activity area. The rightmost window shows the actual wizard. The Bioclipse workspace appears in the background with the JavaScript console in display.

As long as there are no compounds found the wizard cannot proceed to the next page or finish. Occasionally as in the ChEMBL wizard there are searches that reaches over 10000 hits. However, the same problem for the Virtuoso endpoint exists here where it cannot return more than 10000 results. Lastly, Appendix 3 shows a step-by-step manual for this wizard.

4.3 Molecular Substructure Mining

The third and last approach was performed with two separate molecular substructure mining analyses to locate regularities within compounds studied in relation to kinase families. The primary goal involved locating discriminative fragments between active and inactive compounds for every kinase family. A total of seven different searches were made. The used families are shown in Table 1 where corresponding activity properties are displayed which includes among others the size of the active and inactive molecule sets. The study was performed with the developed ChEMBL-MoSS feature that can be seen in Section 3.2.3. The used parameters for MoSS can be further reviewed in Appendix 2.

Furthermore, Figure 16 shows seven independent plots i.e. one for each inspection. The different plots show the distribution frequencies of the found substructures. The x-axis in the plots represents the percentage of a found substructure in molecules set classified as active (focus part). The y-axis on the other hand corresponds to the frequency of the found substructure within molecules set as inactive (complement part). Overall, the plots show the relationship between active and inactive frequencies of found substructures. Interesting fragments should be found in the right bottom area where the support is high in the active subset and low for the inactive subset. The data is unfortunately not optimal hence interesting fragments is quite spread around in the plots i.e. the chosen support is tolerable for all values within the plot area. Each study used individual minimal and maximal settings for support in focus and complement part. In percent (%): 5/2 (TK), 5/2 (CMGC), 5/2 (CAMK), 10/2 (TKL), 10/2 (STE), 5/2 (AGC), 15/2 (CK1). Some of the outstanding fragments, shown in SMILES, are provided in Table 3.

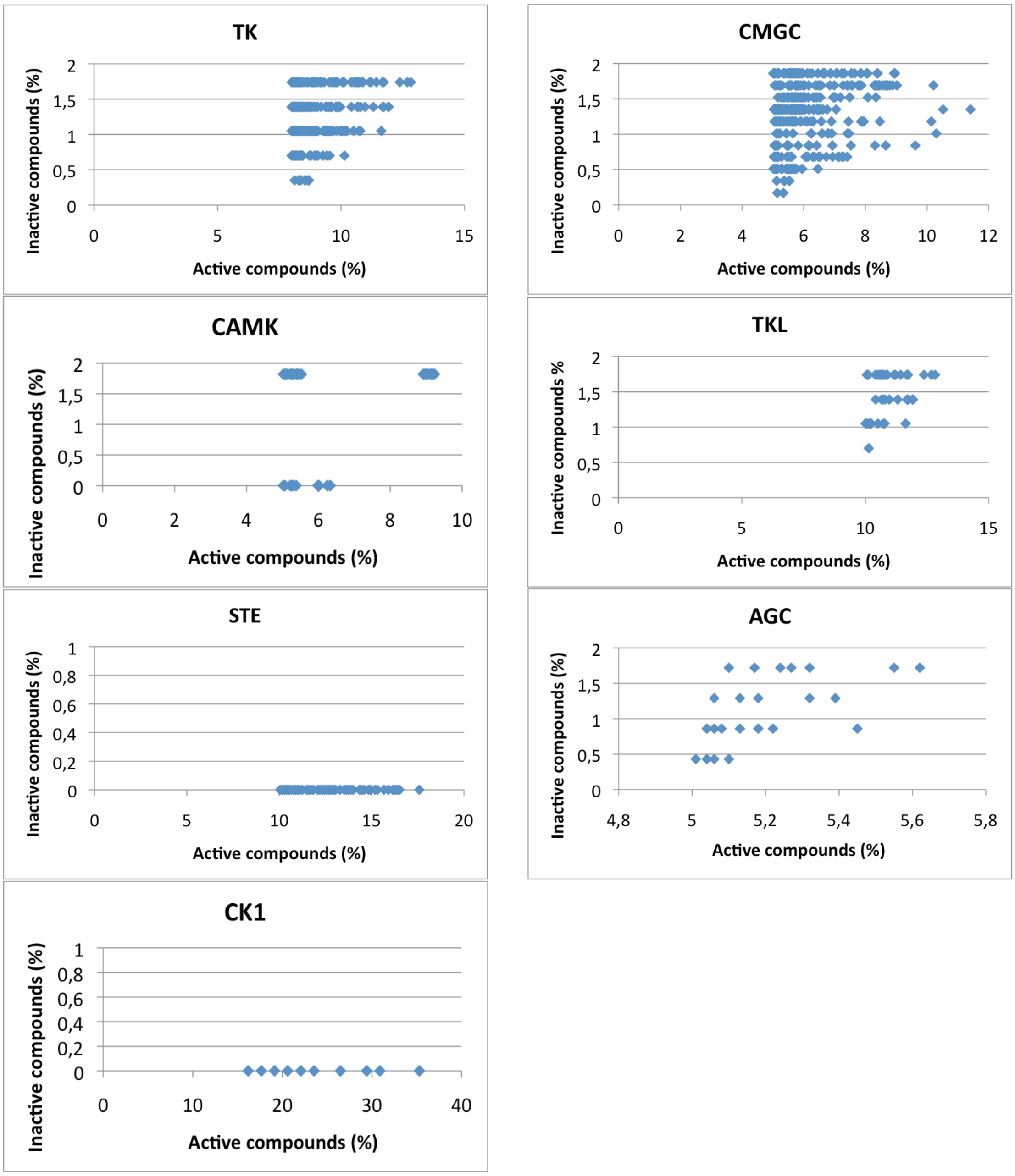


Figure 16. The relation between substructure frequencies in focus and complement sets for active and inactive molecules are displayed. One subplot represents the result from a performed study and there is a graph for each kinase family investigated. The plots display the frequency relationship between substructures found in active (x-axis) and inactive (y-axis) compounds. The minimal support for the active compounds were set to 5%, 10% or 15% while maximum support for the inactive compounds were set to either 1% or 2%. It is seen that there exist active substructures with high support and inactive molecules with quite low support indicating that there are fragments that could be acting as important role players in these ligand-target interactions.

The second aim in the substructure mining studies attempted to discover patterns in compounds binding to proteins of the Tyrosine kinase family (TK), the biggest and most known kinase family. These molecules were set as focus part and compounds derived from the remaining families set as complement part: one family for each study. A total of six studies were completed. The relation between the frequencies of the TK family (focus) and the other families (complement) are given in Figure 17. All plots in Figure 17 represents every performed analysis. The x-axis represent the frequency of substructures in relation to the TK family while the y-axis shows the rate of those structures in the comparing family. The graphs can be reviewed the same way as the plots in Figure 16. Individual minimal and maximal support in focus and complement were set. In percent (%): 5/1 (TK-CMGC), 5/1 (TK-CAMK), 5/1 (TK-TKL), 5/1 (TK-STE), 5/1 (TK-AGC), 10/1 (TK-CK1). Furthermore, frequently occurring molecules for every study, given in SMILES, can bee seen in Table 4.

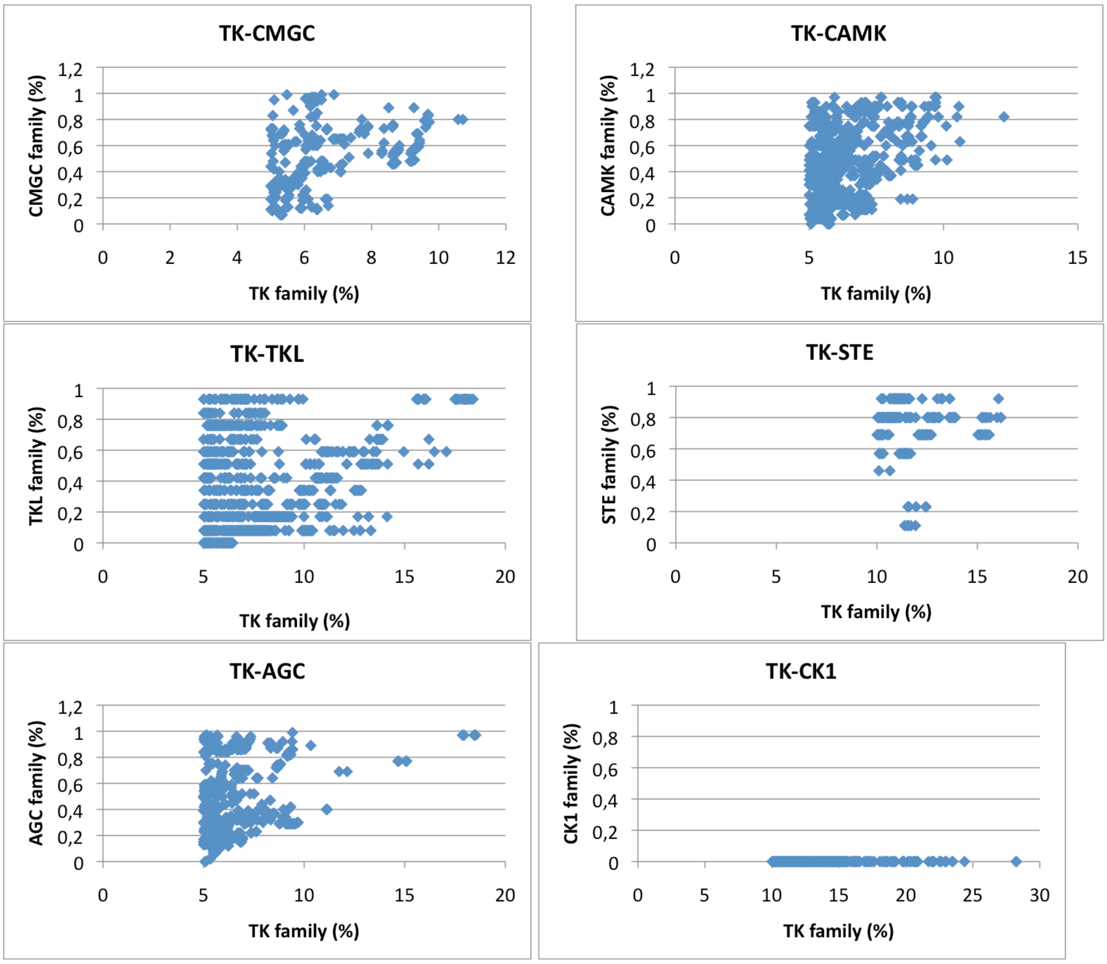


Figure 17. The relationship between substructure frequencies for TK (focus) and the other families (complement) are shown. The plots display the relation of the frequencies between substructures found for the Tyrosine kinase family against structures found for the remaining kinase families. The minimal support for the TK family was set to either 5% or 10% and maximal support for the remaining families was set to 1%. The plots show that there are interesting groups of substructures where the support is high for compounds corresponding to the TK family and where the support is low support for the complement sets. This implies that there are numerous of important substructures in compounds acting on TK proteins while they are not as significant for other families. The CK1 study indicates that a to small subset of molecules exist for that family since no spread is shown.

Table 3. Support for Active (A) and Inactive (I) compounds. Outstanding and frequently occurring fragments between active and inactive compounds are shown for each kinase protein family. The structures are reported in SMILES.

A (%)	I (%)	Structure (SMILES)	A (%)	I (%)	Structure (SMILES)
TK			**CMGC**		
12.83	1.74	N(-C)-c:c:c:c:c:c:c:c	11.41	1.35	O=C-N-c(:n):c
12.69	1.74	O(-C)-c:c:c:c:c:n:c:c	10.3	1.01	N(-c:c:c:c:c:c:c:c)-C-C
10.14	0.7	n(:c:c:c:c:c:c):c:n:c:c	7.86	1.86	F-c(:c):c:c:c:c:c(:n):c:c:c:n
8.36	0.35	Cl-c1:c:c:c:c(-Cl):c:1	5.75	0.51	s(:c):c-N
			5.71	1.86	Cl-c:c-N
CAMK			**TKL**		
9.23	1.82	Cl-c:c:c(-N):c	12.83	1.74	N(-C)-c:c:c:c:c:c:c:c
6.33	0	s:c-N-C	12.69	1.74	O(-C)-c:c:c:c:c:n:c:c
6.25	0	s:c:c-C	10.14	0.7	n(:c:c:c:c:c:c):c:n:c:c
			8.36	0.35	Cl-c1:c:c:c:c(-Cl):c:1
STE			**AGC**		
17.58	0	O-c:c(-C):c	5.62	1.72	s:c:c:c:c-C
16.51	0	O-c:c:c:c(-N-c(:c:c):c:c:c):c:c	5.55	1.72	F-c:c:c:c:c:c:c
15.91	0	n:c:c:c:c:c(-C):c			
15.31	0	O-C-C-N-C-C			
13.88	0	s1:n:c(-O):c(-C):c:1			
CK1					
35.29	0	n1:c:c:c:c:c:1			
30.88	0	C(-C)-c1:c:c:c:c:c:1			
29.41	0	N(-c:c:c)-C-C			
20.59	0	O=C(-C)-C			
16.18	0	O(-C)-c1:c:c:c:c:c:1			

Table 4. Discriminative substructures found for the TK family and their support are displayed. Interesting structures for the TK family are found when other families were set as complement. The structures are reported in SMILES.

TK (%)	CMGC (%)	Structure (SMILES)	TK (%)	CAMK (%)	Structure (SMILES)
10.72	0.8	O(-C)-c1:c:c:c(:n):c(:c):c:1	12.25	0.82	N(-C)-c:n:c:c:c:n:c
9.72	0.78	O(-C)-c(:c:c:c):c(-O):c:c	10.13	0.49	O-c:c:c:c:c(:c):c-N
6.71	0.14	P=O	6.35	0.15	Cl-c(:c):c-N-c:c
6.33	0.97	Cl-c:c-N-c:c:c:n:c	6.23	0.07	P(-O)=O
6.3	0.82	Cl-c:c:c:c:c:c-O	5.77	0	O-c1:c:c:c:c(:c-N-c2:c:c:c:c:c:2):c:1
5.05	0.12	P(-O)(-O)(-O-C)=O	5.07	0	N(-c:c:c)-c(:n):c:c(:n):n:c
TK (%)	**TKL (%)**	**Structure (SMILES)**	**TK (%)**	**STE (%)**	**Structure (SMILES)**
18.4	0.93	N(-c:c:c:c)-c(:n):c:c:n:c	16.18	0.8	N(-c:c)-c:n:c:c:c
9.12	0.25	N(-c1:c:c:c:c:c:1)-c:n:c:n:c(:c):c:c:c	11.92	0.11	N(-c:c:c:c:c)-c:n:c:c:c:c
9.14	0.42	O=C-C-C-c:c	10.22	0.92	O=C(-N-c:c:c)-C-C
6.35	0.67	Cl-c(:c):c-N-c:c	9.64	0	N-c(:n):c(:c):c(:c):c:c:c
6.23	0	P(-O)=O	6.33	0.69	Cl-c:c-N-c:c:c:n:c
5.97	0	O-c:c:c:c:c:c:c(-N):n	6.28	0	P(-O)=O
TK (%)	**AGC (%)**	**Structure (SMILES)**	**TK (%)**	**CK1 (%)**	**Structure (SMILES)**
18.53	0.97	O(-C)-c:c:c:c-N	28.26	0	N-c:c(:c):c:c:c
9.57	0.29	Cl-c:c-N	23.45	0	n1:c:c(:c):c:n:c:1
6.71	0.17	P=O	20.3	0	O-c(:c):c:c:c:c-N
6.21	0.17	P(-O)(-O)=O	8.13	0	Cl-c1:c:c:c:c(-Cl):c:1
5.1	0	O-c1:c:c:c(-C-C-C(-N-C(-C)-C)=O):c:c:1	6.71	0	P=O

5 Discussion

The aim of the study was to contribute to pharmaceutical related studies with development of computer tools in order to facilitate discovery of reported compounds in the ChEMBL database. The ambition was also to find frequently occurring molecular substructures within such compounds binding to various kinases. A new plug-in in Bioclipse was created with features aiming to simplify the retrieving of drug related knowledge. Additionally, the performed molecular substructure studies gave a first clue of what type of patterns that could be of importance in ligand-target interactions of kinase receptors. The research question for the paper was *–"How can the Semantic Web and molecular substructure mining contribute to drug discovery research? -As benchmark data kinase protein ligands extracted from the ChEMBL database is used.*

5.1 Semantic Web

Data interoperability is one of many important features of the Semantic Web where scientists easily can connect their data with each other, contributing to great knowledge distribution. Compared to relational databases where schemas are highly required to access data SPARQL has the benefit of being independent of database structure due to the use of ontologies that hides such details. RDF is also very flexible: changes and extensions are easily made making the future for the RDF-ized ChEMBL database bright.

Machines are able to understand syntax and communicate with each other in order to find answers. This not only contributes to faster disposition of data but also more active help from computers. It is reflected in for example the possibility to query with absolute precision generating specific results that manage to avoid human workflows of long chains of answers triggering new questions. Likewise, in this study it is possible to ask precise questions to for instance retrieve compounds that are ligands to a certain type of protein, that are active in a given interval and binds to a protein with the activity IC_{50}. Such querying is possible because of the well-structured metadata provided by RDF and the extended vocabularies from RDFS and OWL. Moreover, the Semantic Web also contributes to richer content because of its support of Open Data and more well formed collaborations between scientists. An important thing to notice is that the mark-up used in the RDF-ized ChEMBL database is not only built on known standards due to novel required mark-ups. Instead some identifiers are created locally for this project including molecule id, activity id and assay id. Not until there exist standards that cover every model troubles could appear when linking data with each other. Conflicts between resources could occur due to different markings in respective model, but since no external linking was made no similar conflict was encountered.

The next movement involving the Semantic Web would be to start an interaction between ChEMBL data and other projects including ChEBI[6], HCLS[21], Bio2RDF[19] and chem2Bio2RDF[20]. The connection between databases is established with unique identifiers. At the moment molecules in the ChEMBL database only have a ChEBI id that could be used for external linking. This id does unfortunately not match the ChEBI id found in the Bio2RDF knowledgebase due to different installed versions. Therefore it was not possible to begin these forms of expansions. Once it is possible to connect these databases more detailed information about molecules could be retrieved such as InChI's, molecular weight, charge etc. The interaction with databases could for instance give knowledge about specific genes that are common for certain ligand types generating further analysis like expression studies. Additionally, awareness about what kind of diseases a gene or its family is responsible for and specification of metabolic pathways could also be examples of knowledge that can be easy located with the help from different database interactions. Moving even further into the future I believe that Semantic Web projects involving pharmaceutical drug discovery will include patient archives improving pharmacogenomic findings. Actually studies involving semantic technologies and pharmacogenomic findings have already begun[44]. Unfortunately, factors such as patient confidentiality, ethics and gene patent complicate analysis.

5.2 System interaction

Implementation of ChEMBL data into Bioclipse has contributed with two new features enhancing research for finding potential pharmaceuticals. Unfortunately, no experts participated during the design since there were not any skilled users available that could have made such an influence. Development of systems should always include experts that can influence the usability for the better in order to make a system as area adapted as possible. With the help from the project blog http://annziproject.blogspot.com and through other system developers some feedback was actually given. Additionally, the resulting wizards were tested on random people to investigate understandability of the applications. Their opinions and thoughts were applied to make improvements. Most of the mentioned principles described in Section 2.6 have been applied during the design. Though, cases where they are violated do exist. For instance the working memory is not fully supported. When the wizard opens the workbench behind it locks down and other documents are thus unreachable. Also, settings from the first wizard page could be shown on the second page and vice versa to reduce the moving back and forward to seek desired information. These problems are somewhat minor and the work situation is still friendly and usable. Although, the working memory is highly supported by the distribution of components in the wizards: each component is classified as a 'chunk' in the brain, which supports the way the brain handle and interpret data.

The two last rules in the MoSCoW theory "Could have" and "Want to have but will not have time" is from where future development should take place. Both rules contain unimplemented functions for ChEMBL-MoSS they include: progress bar, selecting more than one activity span, viewing of selections on other pages and better visualization of results. Further required features for the ChEMBL application should be: progress bar, more search functions, multiple item drag and drop. Both also need to invoke experts to be fully developed, which I believe will happen as users come along and of course if the project is taken over by someone else in the Bioclipse group. Moreover, with these features users do not have to learn SPARQL or RDF, which from a user interaction perspective is an advantage. Opportunity to learn and adjust a used SPARQL query is however possible thanks to the .sparql file that is generated from the ChEMBL-MoSS feature, see Section 4.2. By invoking *rdf.sparqlRemote* in the JavaScript console modifications in such a query could be made indicating that further and more detailed studies can be made. This is a strong reason why Bioclipse was chosen as platform for this study. Another example of adapted system interaction is that the features apply both text-based and graphical interfaces that according to Mandler (1980) generate good work situations for both novice and experienced users.

5.3 Pattern recognition

Electronegative groups are frequently occurring in compounds that are classified as active. Fluorine, chlorine, sulfur, oxygen and nitrogen atoms have been detected. These atoms are present in functional groups like amides, alcohols, ketones and amines. This indicates that polar molecules probably are desirable candidates in the interaction with kinase proteins. Aromaticity is also property that constantly appears for the found substructures implying that stable structures are required, perhaps in the binding sites where the bonds between ligands and targets must be strong. Also, it is not very surprisingly that the ligands show resemblance to each other since the kinase families are closely related. Unexpected data is seen in Table 3 where TK and TKL have the same substructures and support. Perhaps the same kind of molecules act for both families or a possible error exist. To exclude the possibility of faulty made regular expressions filtration this was tested and did not show any indications of errors. It is rather strange that there are identical values and fragments for these families and the reason should be more closely looked into. Moreover, interesting patterns are evidently shown in the in-between datasets as well. Not only do they also have tendencies of electronegative groups and aromaticity but also phosphor fragments are located. Examples of found substructure are P(-O)(-O)=O and chlorine and nitrogen (71% of the structures) fragments such as Cl-c(:c):c-N-c:c. Ring structures occur at random support for the TK family. Occasionally the support is higher as between TK-CMGC and TK-TKL and not as high between TK-AGC but the fact is that there are ring structures that are frequent for TK and not for the other families.

Substructures including phosphor fragments are found in the in-between family study for TK but disturbingly not in the active-inactive classification analysis. An additional mining analysis was performed where the active compounds were set as complement and inactive molecules as focus part. But this did not result in any phosphor-including substructures either. Clearly, since they do not appear in the active-inactive studies these structures do not fulfill the given supports. This indicates that phosphor-including fragments are occurring in both active and inactive compounds with approximately the same proportion. However, there is no knowledge about the activity for these compounds. Perhaps they are close to the cut-off limit and should be further investigated to conclude that phosphor-including molecules are not playing an important role for active compounds that interact with TK proteins. What actually can be concluded is that compounds acting on Tyrosine kinase do contain phosphor fragments in a much larger extent than for any of the other families. A further investigation is suggested. An example would be to move the boarders that distinguish active from inactive compounds in order to see if these molecules appear close to the existing limit. When comparing these phosphor fragments to other found substructures in the analysis I believe that they do have a certain importance since the other discoveries mostly include common organic sequences.

The research around compounds acting on kinases can be further explored with analyses concentrating on locating regularities in inactive molecules as well i.e. letting the inactive compounds be in the focus group and active set to be in the complementary group. This study would help locating substructures that are frequently occurring in inactive molecules to find an explanation of why they are not as desirable candidates. Steric hindrance, acid/basic groups, polarity and weak bonds might be factors that are present that disturb the fitting of ligands in targets. If such regularities are found a better descriptive model could be created. Of course more studies involving the other kinase families could and should be performed to get more understandings on how those proteins work and what drugs that are suitable for kinases. Since the scientific part of the Semantic Web projects such as Bio2RDF is growing perhaps more data could be added to the model to see if the model is supported with other datasets. Other possibilities would be to examine already existing pharmaceuticals interacting with kinase receptors to study their structure and behavior. This paper could actually be seen as the beginning of these types of analyses where there now exist more questions than real answers. This is exciting since there even now exist potentials for further studies.

A limitation from the Virtuoso endpoint as described in Section 4.1 and 4.2 is the cut-off that appears when a search reaches a result greater than 10000 hits. This affects data where large datasets are used since all the existing compounds in the database are not returned. There are two families affected by this: TK and CMGC. Even though this was known their data were taken into this study anyhow since pattern recognition should be possible anyway. ChEMBL do not contain all potential compounds for these families as research might be closed or not yet reported in to ChEMBL. This means that the compounds in this study is only a fraction of all possible molecules thus the search should not be affected by a cut-off. If patterns do exist they should appear even though a cut-off has been made. On the other hand, this is something that needs to be adjusted since other projects might be more sensitive to dataset size. Moreover, occasionally there are molecules found in the database interacting with a target more than once with the same type of activity but with different values. Reporting of different assays for the same target in the original ChEMBL database could have caused this but which one is then the accurate answer? Because this question could not be answered all compounds were brought in the mining studies. The downside to this is that a certain fragment might have appeared more frequently than it should. As the understanding of this appeared late in the study no measures to solve this were made. A proposal is to scan files before mining them to remove duplicates. This proposal should also be made towards those compounds intervening with more than one target. Removing compounds that binds to more than one target was not an option either because that is an extremely interesting fact. The active structure actually binds to more than one target for some reason and it is interesting to find out why.

6 Conclusions

"How can the Semantic Web and molecular substructure mining contribute to drug discovery research? - As benchmark data kinase protein ligands extracted from the ChEMBL database is used.

This research question summarizes the aim of the study, which was to improve pharmaceutical related research with the use of molecular substructure mining that were applied on ligands binding to proteins derived from kinase families. The aim was also to develop applications adjusted after the needs that these kinds of analysis have. Improvements in retrieving promising drug candidates and related information from the ChEMBL database with Semantic Web technologies have been made with the creation of two applications in the Bioclipse platform. We used one of these features to retrieve molecules binding to kinase proteins with the activity IC_{50} to further explore them in substructure mining analysis with MoSS. Over thousands of substructures and six very characteristic groups were found such as: electronegative groups, amines, amides, phosphors, phenyls and aromatic groups. This shows that the introduced methods in this paper make it possible to chemically study patterns in the ChEMBL database. Additionally, the substructure mining studies show that patterns in ligands interacting with proteins in the Tyrosine kinase family contain phosphor fragments in a much higher extent than ligands correlated to other kinase families. In addition, phenyl groups are also often detected in Tyrosine kinase binding molecules. Over all it is shown that polar and aromatic compounds are desirable for kinase proteins since they appear regularly in molecules classified as active and not as frequently in inactive classified compounds.

Moreover, extended research is suggested for the in-between datasets. More full-scaled results for all the ligands-target interactions could be retrieved if comparisons between all possible family relations were performed. Also, locating undesirable substructures in molecules classified as inactive would generate a more detailed descriptive model. Unfortunately, only limited conclusions can be derived from this study due to small amounts of data. The Semantic Web is part of a growing network where lots of knowledge from studies are freely available, for instance through the chem2bio2rdf project. Recommended future research is to utilize the power of the Semantic Web to collect more data, which could help examine if the model holds for other dataset. This could also uncover more information regarding the substructure mining analysis to observe whether or not other research verdicts support or contradict these results. Looking at for instance drugs that already bind to kinase targets could be one suggestion of verifying the result. If FDA approved drugs actually appear to have the same pattern regularities this paper shows this could in fact validate these results. Alternatively, laboratory expertise is also an alternative to confirm these discoveries. Testing molecules with qualities that this study indicates are of importance against kinase targets reveals information if such properties are desirable through for instance affinity and toxicity and other important parameters.

7 Acknowledgements

I would like to thank to my supervisor Dr Egon Willighagen for this opportunity. I also would like to thank Prof. Jarl Wikberg for being my scientific reviewer and once again letting me perform a project at the Department of Pharmaceutical Bioscience. Thanks also to the rest of the department for your opinions, sharing of knowledge and of course always lending me a key. I'm very grateful to Andreas Olsson for proofreading and giving valuable inputs. Moreover, I would like to thank Margareta Krabbe at Biology Education Centre at Uppsala University for her kind support. Finally, I would like to thank my opponents Sebastian DiLorenzo and Anders Sundström.

8 References

[1] R. Ng,”*Drugs: From Discovery to Approval*”, Wiley-Blackwell, 2008.

[2] P. Krogsgaard-Larsen, U. Madsen, and K. Stromgaard, ”*Textbook of Drug Design and Discovery, Fourth Edition*”, CRC Press, 2009.

[3] B.L. Ligon, ”Penicillin: its discovery and early development”, *Seminars in Pediatric Infectious Diseases*, vol. 15, Jan. 2004, pp. 52-57.

[4] J. Overington, ”ChEMBL. An interview with John Overington, team leader”, Chemogenomics at the European Bioinformatics Institute Outstation of the European Molecular Biology Laboratory (EMBL-EBI). Interview by Wendy A. Warr, *Journal of Computer-Aided Molecular Design*, vol. 23, Apr. 2009, pp. 195-198.

[5] ChEMBL Team Home Page | EBI, [http://www.ebi.ac.uk/chembl/], January-July 2010.

[6] Chemical Entities of Biological Interest (ChEBI), [http://www.ebi.ac.uk/chebi/], August 2010.

[7] Kinase SARfari, [http://www.sarfari.org/kinasesarfari/], January-July 2010.

[8] FTP EBI, [ftp://ftp.ebi.ac.uk/pub/databases/chembl/], January 2010.

[9] C. Borgelt, T. Meinl, and M. Berthold, “MoSS: a program for molecular substructure mining,” *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, Chicago, Illinois: ACM, 2005, pp. 6-15.

[10] C. Borgelt's Webpage, [http://www.borgelt.net/moss.html], January-July 2010..

[11] O. Spjuth, T. Helmus, E. Willighagen, S. Kuhn, M. Eklund, J. Wagener, P. Murray-Rust, C. Steinbeck, and J. Wikberg, “Bioclipse: an open source workbench for chemo- and bioinformatics,” *BMC Bioinformatics*, vol. 8, 2007, p. 59.

[12] Bioclipse, [http://bioclipse.net/], January-July 2010.

[13] C. Borgelt, M.R. Berthold, and D.E. Patterson, “Molecular Fragment Mining for Drug Discovery.”

[14] M.J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, *New Algorithms for Fast Discovery of Association Rules*, University of Rochester, 1997.

[15] T. Berners-Lee, J. Hendler, O. Lassila, and others, “The Semantic Web,” *Scientific american*, vol. 284, 2001, pp. 28–37.

[16] Semanticweb.org, [http://semanticweb.org/wiki/Main_Page], January-July 2010.

[17] World Wide Web Consortium (W3C), [http://www.w3.org/], July 2010.

[18] Wiki.dbpedia.org , [http://dbpedia.org/About], July 2010.

[19] Quebec | Bio2RDF.org, [http://bio2rdf.org/], July 2010.

[20] chem2bio2rdf, [http://cheminfov.informatics.indiana.edu:8080/], July 2010.

[21] Semantic Web Health Care and Life Sciences (HCLS) Interest Group, [http://www.w3.org/2001/sw/hcls/], July 2010.

[22] RDF - Semantic Web Standards, [http://www.w3.org/RDF/], July 2010.

[23] F. Manola and E. Miller, “RDF Primer, [http://www.w3.org/TR/rdf-primer/], July 2010,” Jul. 2010.

[24] D. Beckett, “RDF/XML Syntax Specification", [http://www.w3.org/TR/REC-rdf-syntax/], July 2010.

[25] D. Beckett and T. Berners-Lee, “Turtle - Terse RDF Triple Language", [http://www.w3.org/TeamSubmission/turtle/], July 2010.

[26] T. Berners-Lee and D. Connolly, “Notation3 (N3): A readable RDF syntax", [http://www.w3.org/TeamSubmission/n3/], July 2010.

[27] T. Berners-Lee and D. Connoly, “N-Triples", [http://www.w3.org/2001/sw/RDFCore/ntriples/], July 2010.

[28] D. Brickley and R. Guha, “RDF Vocabulary Description Language 1.0: RDF Schema", [http://www.w3.org/TR/rdf-schema/], July 2010.

[29] W3C OWL Working Group, “OWL Web Ontology Language Overview", [http://www.w3.org/TR/owl2-overview/], July 2010.

[30] E. Prud'hommeaux and A. Seaborne, *SPARQL Query Language for RDF, [http://www.w3.org/TR/rdf-sparql-query/], January-July 2010*, 2006.

[31] GitHub kurtjx's SNORQL, [http://github.com/kurtjx/SNORQL], January 2010.

[32] Eclipse.org home, [http://www.eclipse.org/], January-Juli 2010.

[33] Open Source Rich client platform (RCP) applications, [http://www.eclipse.org/community/rcpos.php], July 2010..

[34] O. Spjuth, J. Alvarsson, A. Berg, M. Eklund, S. Kuhn, C. Masak, G. Torrance, J. Wagener, E. Willighagen, C. Steinbeck, and J. Wikberg, “Bioclipse 2: A scriptable integration platform for the life sciences,”*BMC Bioinformatics*, vol. 10, 2009, p. 397.

[35] SWT: The Standard Widget Toolkit, [http://www.eclipse.org/swt/], March-July 2010.

[36] R. Harris, *The Definitive Guide to SWT and Jface*, APress,US, 2007.

[37] JFree.org, [http://www.jfree.org/index.html], June 2010.

[38] JCommon, [http://www.jfree.org/index.html], May 2010.

[39] S. Ash, Moscow_prioritisation_briefing_paper.exe, [http://www.dsdm.org/knowledgebase/details/165/moscow-prioritisation-briefing-paper.html], Apr. 2007.

[40] D. Benyon, P. Turner, and S. Turner, *Designing Interactive Systems: People, Activities, Contexts, Technologies*, Addison Wesley, 2005.

[41] DSDM Consortium - Enabling Business Agility, [http://www.dsdm.org/], June 2010.

[42] M. Passer and R. Smith, *Psychology: The Science of Mind and Behavior*, McGraw-Hill Humanities/Social Sciences/Languages, 2007.

[43] Jena Semantic Web Framework, [http://openjena.org/], July 2010.

[44] M. Dumontier and N. Villanueva-Rosales, “Towards pharmacogenomics knowledge discovery with the semantic web,” *Briefings in Bioinformatics*, vol. 10, Mar. 2009, pp. 153-163.

Appendix

Appendix 1 – Methods of the ChEMBL plug-in

Table 5 contains methods that build up the ChEMBL plug-in, presented in Section 3.2.2. The methods are available from the JavaScript console in Bioclipse via the ChEMBL manager. They make it possible to search and retrieve information about ligands and targets and further it is even possible to perform substructure mining. Moreover, the code is available at [http://github.com/bioclipse/bioclipse.chembl].

Table 5. Explanations of the methods for the ChEMBL plug-in are shown.

cutter(*IStringMatrix matrix*)	Removes URI, keeps relevant information.
getActivities(*Integer targetID*)	Finds existing activities for a target with target identifier.
getChEBIId(*Integer molID*)	Returns ChEBI id of a molecule.
getCompoundInfo(*Integer chebiID*)	Get activity type, assay type, conf score, target id and description from ChEBI id.
getCompoundInfoWithKeyword(*String keyword*)	Finds SMILES, ChEBI id, target id and description from a keyword.
getCompoundInfoWithSmiles(*String SMILES*)	Finds ChEBI id and title (slow search).
getPCM(*String activity, String class6, String class3*)	Finds classifications, target id, pubmed, protein seq and activity value where activity type is filtered.
getProperties(*Integer targetID*)	Finds organism, protein type and title from a target id.
getProteinData(*Integer targetID*)	Finds activity type, various classifications, ChEBI ids, organism, UNIPROT, EC number from a target id.
getQSARData(*Integer targetID, String activity*)	Finds SMILES and activity value from target id and activity type.
getQSARDataExtended(*Integer targetID, String activity*)	getQSARData + conf score and activity unit.
getTargetIDWithEC(*Integer ec-number*)	Finds target ids and its description that have a specific EC number.
getTargetIDWithKeyword(*String keyword*)	Finds target id and description from a keyword.
getTargetIDWithProteinSeq(*String protseq*)	Finds target id from a protein seq (slow search).
getTargetSequence(*Integer targetID*)	Returns a protein seq from a target id.
saveCSV, saveFile, saveToMossFormat	Save functions.
mossGetCompoundsFromProteinFamily (S*tring family, String acttype, Integer limit*)	Collect compounds from a given family and activity type.
mossGetCompoundsFromProteinFamilyWithActivity (String family, String acttype)	Collect compounds and activity value given family and activity type.
mossGetCompoundsFromProteinFamilyWithActivity SPARQL(*String family, String activity*)	Contains the hardcoded query for the method with the same name.
mossGetCompoundsFromProteinFamilyWithActivity Target(*String family, String acttype, Integer limit*)	Collects compounds, activity value and involved target given family and activity type.
mossGetCompoundsFromProteinFamilyWithActivity Bound(*String family, String acttype, Integer lower, Integer upper)*	Collect compounds involved with a family with specific activity type from a given interval.
mossAvailableActivities(*String family*)	Finds existing activities for a family.
mossSetActivityBound(*IStringMatrix matrix, Integer lower, Integer upper)*	Updates a dataset (matrix) with an activity interval.
mossViewHistogram(*IStringMatrix matrix*)	Displays a histogram for activities given a matrix.

Appendix 2 – Default parameter settings for MoSS

Molecular SubStructure miner (MoSS) as described in Section 2.3 was used to perform substructure mining analysis on ligands binding to kinase proteins. This made it possible to find discriminative fragments and pattern regularities, which was explained in Section 3.2.3. The default parameters that were used during those analyses are shown in Table 6.

Table 6. Default parameter settings for MoSS.

Parameter	Values
Threshold	0.5
Invert split	no
Minimal substructure size	1
Maximal substructure size	0 (no limit)
Node types to exclude	H
Seed types to exclude	“ ”
Minimal support	5%
Maximal support	2%
Absolute support	No
Unembed sibling nodes	No
Substructure support type	Number of containing graphs/molecules
Parameter	Values
Report only closed substructures	No
Aromatic bond	Extra type
Ignore type of bonds	Never
Ignore type of atoms	Never
Match charge of atoms	No
Match aromaticity of atoms	No
Convert Kekulé representation	Yes
Distinguishing ring bonds	No
Variable length carbon chain	No
Maximal embeddings	0

Appendix 3 – Manual for the ChEMBL-MoSS wizard

The ChEMBL-MoSS wizard presented in Section 4.2 is the resulting application for the interaction between ChEMBL (Section 2.2) and MoSS (Section 2.3). Below a step-by-step manual for the application is presented where Figure 18 shows the corresponding numbers that are in parentheses in the guide.

1. First select a protein family from the top combo box (1).
2. If you are online the available activities will be displayed in the box beneath. (2)
3. Choose one.
4. Now the table beneath should contain compounds from chosen family with a specified activity. (4)

If the total hit is fewer than 50 compounds move to step 7.

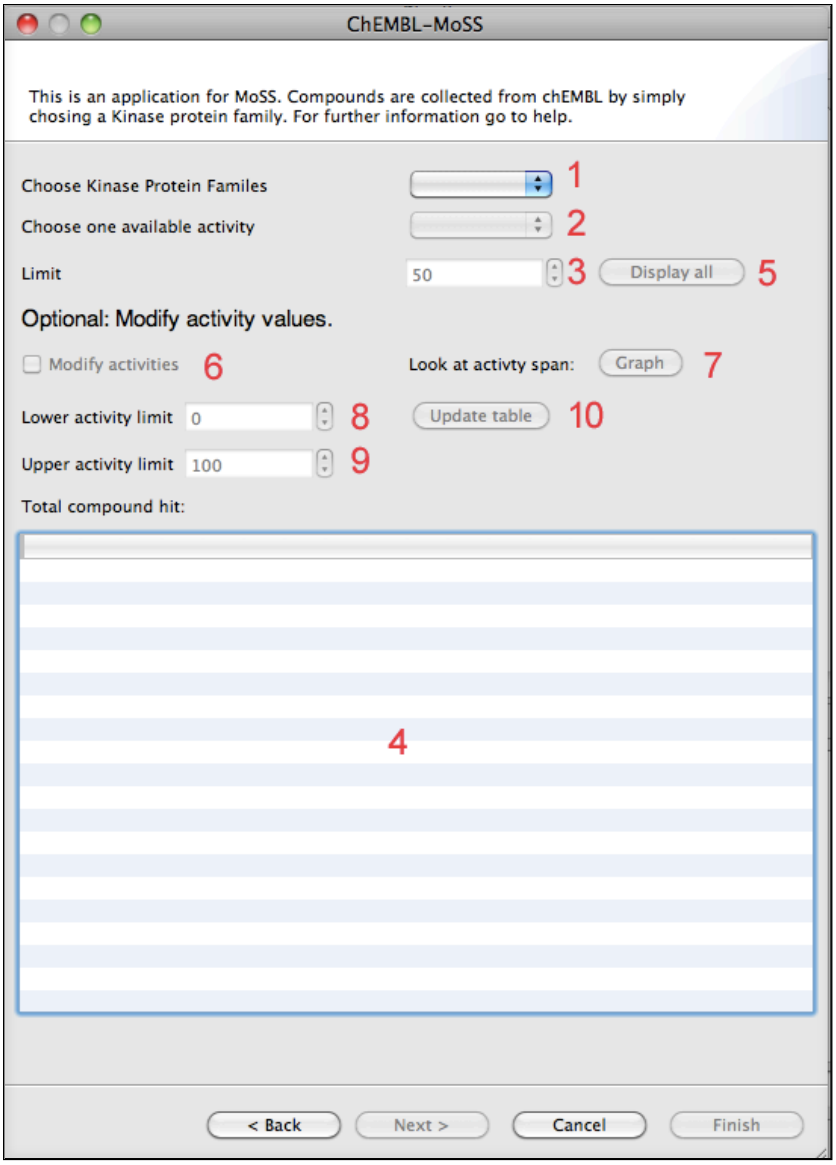
5. The search is limited to a set of 50 compounds. If the set contain more than 50 compounds and you would like to work with all: simply press the button "Display all"(5).

6. You are also able to modify your limit in the limit box (3). If you use the arrows the limit will increase/decrease by one but you could also simply type in a number.
7. Simply click around in the boxes until you find what you are looking for.
8. Modify data by checking the Modify data checkbox (6).
9. To look at the activity interval press the Graph button (7).
10. Update table by adding lower (8) and upper (9) interval limits and push Update table (10).
11. When finish press the Next button.
12. If a secondary dataset is desired start from 1 on the new page. If not press Next again.
13. Chose where to put your file and name it. Three files are saved under that name.

Appendix 4 – Manual for the ChEMBL wizard

The ChEMBL wizard seen in Section 4.1 is a general search engine where information about ligand-target interaction can be retrieved. Various searches can be performed and below a step-by-step manual is given. Figure 19 displays the corresponding numbers in parenthesis given in the manual.

1. Chose search perspective 'Compound' or 'Target' (1).
2. Chose what category you would to search with by clicking into a box (2).
3. Add text or number to the field (3) depending on choice in step 2.
4. Push the Search button (4).
5. Results are shown in the upper table (5)
6. To select an item that you would like to save add it to the lower table (6). This is done by drag and drop or by checking the boxes and pushing the Select button (7).
7. To perform a new search, start over at 1.
8. To delete item(s) from the lower table (6) simply check their boxes and push the Delete button (8).
9. When done click Next to chose where to save file and to name it.



ChEMBL–MoSS

This is an application for MoSS. Compounds are collected from chEMBL by simply chosing a Kinase protein family. For further information go to help.

Choose Kinase Protein Familes

Choose one available activity

Limit 50

Optional: Modify activity values.

□ Modify activities

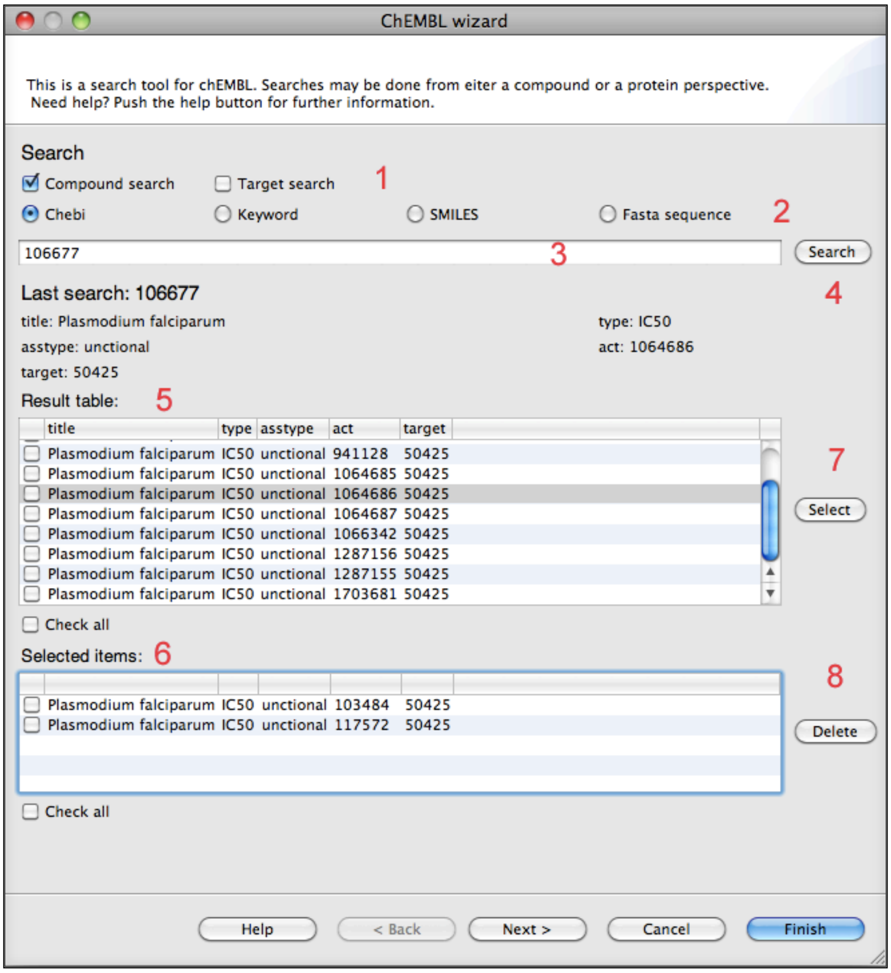
Look at activty span:

Lower activity limit 0

Upper activity limit 100

Total compound hit:

Figure 18. The ChEMBL-MoSS wizard with numbers for guiding purpose, see manual.



ChEMBL wizard

This is a search tool for chEMBL. Searches may be done from eiter a compound or a protein perspective.
Need help? Push the help button for further information.

Search

☑ Compound search ☐ Target search

● Chebi ○ Keyword ○ SMILES ○ Fasta sequence

106677

Last search: 106677

title: Plasmodium falciparum

type: IC50

asstype: unctional

act: 1064686

target: 50425

Result table:

	title	type	asstype	act	target
☐	Plasmodium falciparum	IC50	unctional	941128	50425
☐	Plasmodium falciparum	IC50	unctional	1064685	50425
☐	Plasmodium falciparum	IC50	unctional	1064686	50425
☐	Plasmodium falciparum	IC50	unctional	1064687	50425
☐	Plasmodium falciparum	IC50	unctional	1066342	50425
☐	Plasmodium falciparum	IC50	unctional	1287156	50425
☐	Plasmodium falciparum	IC50	unctional	1287155	50425
☐	Plasmodium falciparum	IC50	unctional	1703681	50425

☐ Check all

Selected items:

☐	Plasmodium falciparum	IC50	unctional	103484	50425
☐	Plasmodium falciparum	IC50	unctional	117572	50425

☐ Check all

Figure 19. The ChEMBL wizard with numbers for guidance, see manual.