Discovery and characterization of polyamine analogues as inhibitors of the *Plasmodium falciparum* polyamine pathway using cheminformatics

by

Jurgens Jacobus de Bruin

Submitted in partial fulfillment of the requirements for the degree Magister Scientiae in the Faculty of Natural & Agricultural Science

Bioinformatics and Computational Biology Unit

Department of Biochemistry

School of Biological Science

Natural & Agricultural Science

University of Pretoria

Pretoria

South Africa

August 2008

Declaration

I, ... declare that the thesis/dissertation that I hereby submit for the degree in ..... at the University of Pretoria has not previously been submitted by me for degree purposes at any other university and I take note that, if the thesis/dissertation is approved, I have to submit the additional copies, as stipulated by the relevant regulations, at least six weeks before the following graduation ceremony takes place and that if I do not comply with the stipulations, the degree will not be conferred upon me.

SIGNATURE..... DATE.....

Acknowledgements

To my Lord, I would like to extend a hand of thanks for it is by Your grace alone I have completed this project.

Thanks to my family for their everlasting support and to Prof Fourie Joubert for the endless advice and guidance throughout this project.

My fellow colleagues thanks to all of you for advice and help provided when needed.

I would also like to thank the National Bioinformatics Network for generously funding and supporting this project.

Table of Contents

List of Abbreviations

1D	1-Dimensional
2D	2-Dimensional
3D	3-Dimensional
4MCHA	Cis-4-Methylcyclohexylamine
ADME-TOX	Absorption, Distribution, Metabolism, Excretion and Toxicity
AdoDATO	S-adenosyl-1,8-diamino-3-thiooctane
AdoMetDC	S-Adenosylmethionine decarboxylase
BBB	Blood Brain Barrier
CHEBI	Chemical Entities of Biological Interest
CNS	Central Nervous System
CoMFA	Comparative Molecular Field Analysis
CSS	Common Substructure
dcAdoMet	Decarboxylated S-adenosyl-L-methionine
FunGIMS	Functional Genomics Information Management System
FP	Finger Print
GPCR	G-Protein-Coupled Receptors
HEP	Human Effective Permeability
HTS	High-Throughput Screening
IDE	Integrated Development Environment
IT	Information Technology
MCS	Maximum Common Structure
MQS	Molecular Quantum Similarity
MR	Molecular Refractivity
MW	Molecular Weight
ODC	Ornithine decarboxylase
PDB	Protein Data Bank
PfSpdSyn	Plasmodium falciparum Spermidine Synthase
Spd	Spermidine
PSA	Polar Surface Area
Put	Putrescine
QSAR	Quantitative structure-activity relationship
QSCD	Quantized Surface Complimentarily Diversity
ROF	Rule-of-Five
RPC-server	Remote procedure calls servers
RTI	Record Type Indicators
SDF	Structure Data File
Sid	System Identifier
SMILES	Simplified Molecular Input Line Entry System
Spm	Spermine
SpmSyn	Spermine Synthase
Tc	Tanimoto coefficient
URI	Uniform Resource Identifier

List of Figures

List of Tables

Summary

It is well known that the costs associated with drug discovery are extremely high due the use of expensive *in vitro* methods as well as the high failure rate of drugs during clinical testing. In order to effectively fight the war against diseases such as malaria a far less expensive approach is required. Increasing the amount of *in silico* work would decrease the amount of experiments that have to be done and so doing reduce the costs. *In silico* methods have the ability to predict major limiting factors in drug discovery resulting in only high quality drug molecules being subjected to clinical trials, thus increasing the probability of success.

In this study a web-based chemoinformatics workspace aimed at research biologists was developed inside the FunGIMS application framework. This particular web-based chemoinformatics application was developed in the context of the FunGIMS suite of tools, specifically providing biological users with some of the most common chemoinformatics functions such as (1) the representation of chemical compounds, (2) chemical data, (3) databases and data sources, (4) structure search methods, (5) methods for calculating physical and chemical data (6) calculation of structure descriptors. The chemoinformatics module is supported by a modified version of the CHEBI database and the integration of OpenBabel and Frowns made it possible to perform analysis on molecules by means of SMILES-structures.

As part of the validation of the chemoinformatics module of FunGIMS, the chemical space of the polyamine pathways in *Plasmodium* was explored in order to obtain a library of molecules based on similarities that could be possible lead-like compounds. This was used in a study of compounds against spermidine synthase from *Plasmodium falciparum*. By means of similarity and substructure searches the chemoinformatics module was able to produce molecules that are structurally similar. Additional filtering enabled a library of related molecules to be obtained, and used in a docking study. Compounds were found with high docking scores, thereby validating the effectiveness and usefulness of the chemoinformatics module of FunGIMS.

Chapter 1

Introduction

The extremely high costs of drug discovery are in part due to the large amount of experimental work that has to be performed in the drug discovery process, as well as the numerous compounds that fail during clinical trials. By increasing the amount of *in silico* screening, the cost associated with drug discovery can be dramatically decreased, as the amount of experiments are decreased and higher quality compounds are sent to clinical trials.

This literature review will discuss the influence of chemoinformatics in drug discovery, focusing on techniques and methodologies such as library design, molecular descriptors, ADME-Tox, QSAR and the concept of similarity and dissimilarity.

1. The Science of Chemoinformatics

It can be stated that chemoinformatics has existed since it was first required to store chemical data. The first computational chemists applied mathematical graph theories to two-dimensional structure-searches of chemical databases. These where complemented by indexing and search tools (Hrib and Peet, 2000; Smith, 2002; Marshall, 2004).

Recently, Frank Brown stated that chemoinformatics is the "mixing of information resources to transform data into information and information into knowledge, for the intended purpose of making better decisions faster in the arena of drug lead identification and optimisation" (Brown, 1998). This concept of data transformation is illustrated in Figure 1.1. Stated differently, chemoinformatics is the combination of chemical synthesis, biological screening and data analysis, all aiding in drug discovery and development (Blake, 2004).

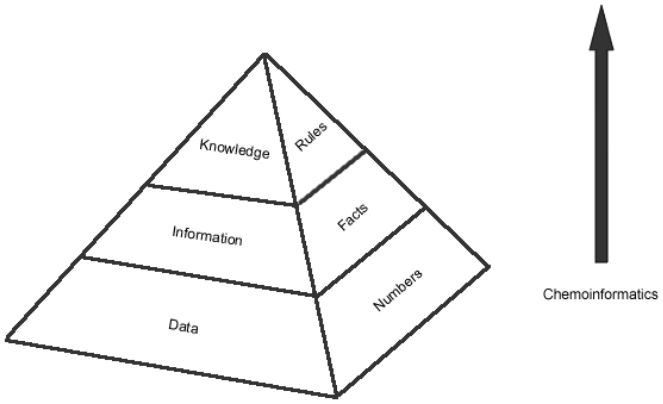


Figure 1.1 : The illustration of the transformation of data to knowledge in chemoinformatics. It is required that data be transformed into information and this information can provide knowledge on the particular system being studied (Hann and Green, 1999).

A short overview of the major tasks in chemoinformatics can be represented by the following list:

1. The representation of chemical compounds.
2. The representation of chemical reactions.
3. Developing data sources and databases.
4. Designing structure search methods.
5. Building methods for calculating physical and chemical data, such as Absorption Distribution Metabolism Excretion and Toxicity (ADME-Tox) or Quantitative Structure-Activity Relationship (QSAR).
6. Calculation of structure descriptors.
7. Development of data analysis methods.

These all represent activities that have been used in drug design for many years (Gasteiger, 2006).

Chemoinformaticists have developed many applications to aid in the relevant areas of drug design, including:

1. The identification of new lead structures.
2. The optimization of lead structures.
3. Establishing QSAR.
4. Construction of virtual libraries containing a specific subset of molecules.
5. Analysing high-throughput screening data.
6. Docking studies.
7. Modelling ADME-Tox data.
8. The prediction of metabolism of xenobiotics.

Figure 1.2 illustrates the involvement of chemoinformatics in drug design.

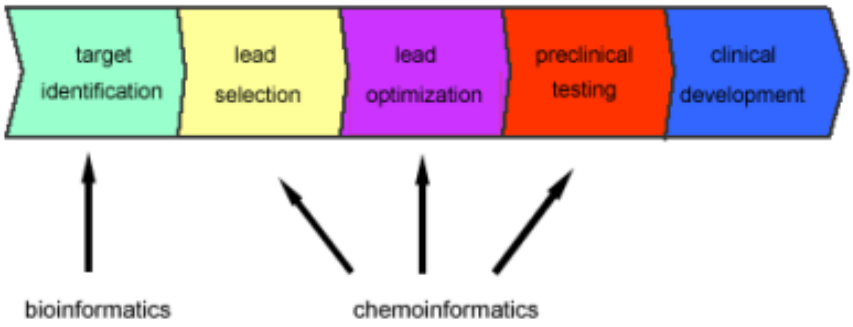


Figure 1.2 : A simplified outline of the drug design process and the influence of chemoinformatics. It is clear that the role of chemoinformatics cannot be ignored (Gasteiger, 2006).

The advances in chemoinformatics over the past years are largely due to the need to handle the exponential growth in the amount of data being generated by techniques such as high-throughput screening (HTS) and library synthesis. Another important factor responsible for the growth of chemoinformatics is the major economic pressure to reduce the cost of developing new drugs, as well as a reduction in time spent on the development (Langer and Hoffmann, 2001; Stahura and Bajorath, 2002). The realization that the biological activity of chemical compounds cannot yet be predicted from first principles opens a door to the development of methods that can learn from

available data. The ever increasing available computational power has given researchers the ability to handle large volumes of data, increasing the speed at which data can be transformed into knowledge. This includes tools to help with the analysis of experimental results and prediction of molecular properties (Cramer, Patterson *et al*., 1998; Xue and Bajorath, 2000; Ritchie, 2001).

2. Chemoinformatics and Drug Discovery

2.1 Drug Discovery

Drug discovery is the process by which drugs are discovered and/or designed. In the past, most drugs have been discovered by identifying the active ingredient from traditional remedies. A new approach to drug discovery is based on extensive background knowledge of the target protein-receptor obtained beforehand, thus understanding the molecular and physiological characteristics and control mechanisms of the target specific entities (Marshall, 2004).

The process of drug discovery starts with the identification of possible candidates, followed by the synthesis of these compounds and characterizing, screening, and assaying for the therapeutic efficiency of the synthesized compounds. Once a compound has shown its value in these tests, it will enter the process of drug development prior to clinical trials (Brown, 1998).

The majority of targets currently selected for drug discovery efforts are proteins. The process of finding a new drug against a chosen target involves high-throughput screening (HTS), where large libraries of chemicals are tested for their capacity to transform the target. For example, if the target is a novel G protein-coupled receptor (GPCR), compounds will be screened for their capability to inhibit or stimulate that receptor. The selectivity of the compounds for the chosen target can also be determined from HTS, the purpose of this is to find compounds that will interfere with only the chosen target, but not other related targets (Brown, 1998; Marshall, 2004).

It is implausible that a perfect drug candidate will emerge from early screening runs. It is more frequently observed that several compounds are found to have some degree of activity. Consequently a library of possible candidate molecules is produced.

2.2 Library Design

Recent advances in combinatorial chemistry have made it possible, even plausible, for chemists to synthesize large numbers of chemical compounds. HTS methods decrease the time required to discover compounds that exhibit biological activity. Nevertheless, the quantity of chemical compounds that can be synthesized has remained a small percentile of that which has been suggested theoretically to have therapeutic effects. Consequently, computer-aided drug design and chemoinformatics have developed promising tools to aid in this task (Duchowicz, Talevi *et al.*, 2007).

The initial procedure in virtual library generation entails the construction of a set of molecules. Two approaches can be followed. The first is the analog building approach, this consists of the systematic substitution of core structures in different positions with a collection of active groups. The second approach involves the construction of molecules based upon reaction driven combinations of fragments (Abraham, Takacs-Novak *et al.*, 1997; Duchowicz, Talevi *et al.*, 2007).

3. Molecular Descriptors

Molecular descriptors can be defined as conceptual representations of molecular structures and properties.

Over the past few years hundreds of descriptors have been reported and the majority of these descriptors can be classified according to their dimensionality.

In accordance to this, one-dimensional (1D) descriptors may represent the majority of properties such as molecular mass, two-dimensional (2D) descriptors would portray properties that can be calculated from the two-dimensional structure of a molecule and three-dimensional (3D) descriptors are more complex as they rely on the 3D conformation of a molecule. Table 1 illustrates some examples of different descriptors (Abraham, Takacs-Novak *et al.*, 1997; Xue and Bajorath, 2000).

Table 1.1 : Molecular descriptors can be classified according to the structural dimension they are deduced from.

Descriptor	Definition
1D Descriptors	
weight	Molecular weight
mr	Molar refractivity
logP(o/w)	Log of the octanol/water partition coefficient
2D Descriptors	
a_nN	Number of nitrogen atoms in the molecule
b_double	Number of double non-aromatic bonds
vsa_acc	Approximate sum of Van der Waals surface area of H-bond acceptors
3D Descriptors	
ASA+	Solvent accessible surface area of all atoms with positive partial charge
pmi	Principal moment of inertia
vol	Van der Waals volume

3.1 Fingerprints

A particular complex form of molecular descriptors that is worth mentioning is fingerprints. These are usually encoded as binary bit strings. The settings of a particular binary string produce a bit "pattern" characteristic of a molecule, Figure 1.3 shows an illustrated design of a fingerprint (Xue and Bajorath, 2000).

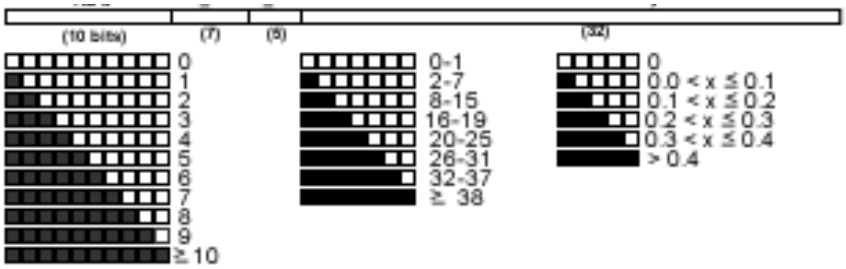


Figure 1.3 : Schematic representation of a binary molecular fingerprint. In this case, bit settings for three numerical descriptors are combined with 32bits, each of which accounts for the presence (1) or absence (0) of a defined structural fragment. Gray shading means bit position is set on (1). For example the value range of HB-a is encoded using segments consisting of 10 bits. If all 10 bits are set to (0) , no hydrogen bond acceptors are present. If the first five are set to (1) then the molecule has five hydrogen bond acceptors. In a "keyed" fingerprint, as shown here, each bit position is associated with a specific descriptors and value. In "hashed" fingerprints, this no longer is case as different features and values are mapped to overlapping bit segments (Xue and Bajorath, 2000).

Bit settings account for a specific range, if numerical descriptors are encoded within the binary format. This entails that for each descriptor a significant value range should be determined preceding the design of the fingerprint (Xue and Bajorath, 2000).

In contrast to other fingerprints Daylight fingerprints consist of 2048 bits, these complex fingerprints are often "hashed" or "folded" (http://www.daylight.com). This means that properties or structural patterns are mapped to overlapping bit segments, consequently resulting in very unique bit patterns. As a result of this a single bit position can no longer be coupled with a specific feature (Xue and Bajorath, 2000). Fingerprints are often used to determine the similarity of other molecules to a specific compound in question. In the case of pair wise comparison, the overlap of fingerprints is represented as a measure of similarity and usually calculated by means of the Tanimoto coefficient (Tc) see section 5.3 (Flower, 1998; Xue and Bajorath, 2000).

3.2 Selection of Descriptors

It is of importance that descriptors represent the chemical reality and situation of a system as best possible in order to maintain the true representation of chemical space of the particular system (Xue and Bajorath, 2000).

Over the years many different molecular descriptors have been used and this brings up the topics of selecting the best possible descriptors. This topic is highly complex and is beyond the scope of this text.

3.2.1 Property Based Selection

The pursuit of superior oral bioavailability for an effective compound may be the most variable, as well as time consuming and labor intensive aspect of drug discovery.

It was stated by Lipinski (Duchowicz, Talevi *et al*., 2007) that these difficulties can be taken back to the pursuit of a compound that exhibits poor physiochemical properties. From this Lipinski's Rule-of-five was born. This is a set of criteria for predicting the oral bioavailability of a compound on the basis of simple molecular features (molecular weight, logP, numbers of hydrogen-bond donors and acceptors). This is often used to profile a library or virtual library with respect to the proportion of drug-like members which it contains (Clark and Pickett, 2000).

LogP, Polar Surface Area and Molar Refractivity.

LogP

A partition (P) or distribution coefficient (D) is the ratio of concentrations of a compound in the two phases of a mixture of two immiscible solvents at equilibrium. These coefficients are thus a measure of differential solubility of the compound between two immiscible solvents. In general one of the solvents chosen is water while the second is hydrophobic such as octanol. Both the partition and distribution coefficient are measures of how hydrophilic or hydrophobic a chemical substance is. Partition coefficients are used in estimating the distribution of drugs within the body. Hydrophobic drugs with high partition coefficients are preferentially distributed to hydrophobic compartments such as lipid bilayers of cells, while hydrophilic drugs with low partition coefficients are preferentially found in hydrophilic compartments such as blood serum (Abraham, Takacs-Novak *et al*., 1997; Blake, 2000).

The logarithm of the ratio of the concentrations of the non-ionized solute in the solvents is called log *P*, given by Equation 1.1 (Abraham, Takacs-Novak *et al.*, 1997) .

$$\log P = \log\left(\frac{[solute]_{oct}}{[solute]_{wat}}\right)$$

Equation 1.1 : LogP, [solute]oct is the concentration of solute in octanol phase and [solute]wat represents the concentration in the water phase

Polar Surface Area.

PSA of a particular molecule is defined as the area of its van der Waals surface that arises from oxygen and nitrogen atoms or hydrogen atoms attached to oxygen and nitrogen atoms as demonstrated in Figure 1.4. The PSA of a molecule can thus be related to the ability of the molecule to form hydrogen bonds (Clark, 1999; Ertl, Rohde *et al.*, 2000).

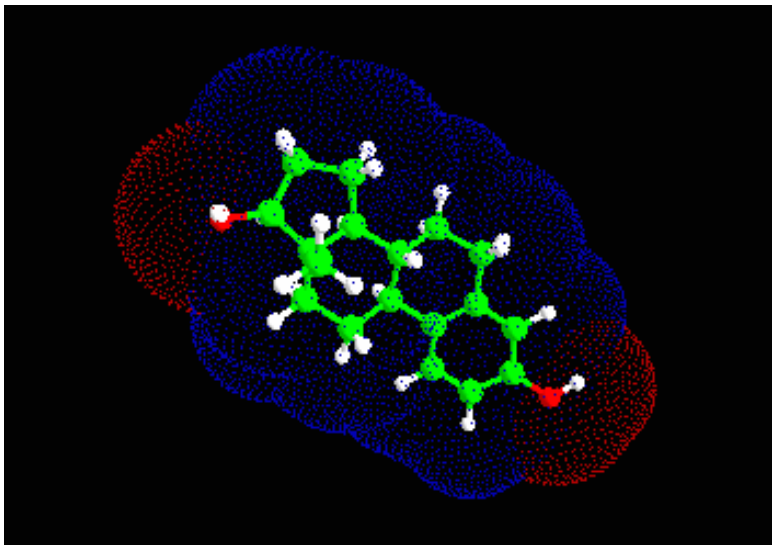


Figure 1.4 : Polar Surface Area of a Simple Molecule. This is defined as the area of its van der Waals surface that arises from oxygen and nitrogen atoms or hydrogen atoms attached to oxygen and nitrogen atoms. Blue surface dots refer to hydrogen atoms bound to carbon and the red surface dots refer to hydrogen bound to oxygen atoms. (http://www.ddl.unimi.it/papers/02/property_calculation.htm).

Molar Refractivity

Molar refractivity is a representation of the real volume of a molecule where radiation is of infinite wavelength. Molar refractivity not only relates to the volume of a molecule, but also to the London dispersive forces that act in the drug-receptor interaction (Padron, Carrasco *et al*., 2002) .

One form of the Lorentz-Lorenz formula gives the molar refractivity of a dilute gas as:

$$A = \frac{RT(n^2 - 1)}{3p}$$

Equation 1.2 : Clausius-Mossotti equation, where R is the universal gas constant, T is the temperature, n is the index of refraction and p is the pressure

4. ADME-Tox and QSAR

4.1 ADME-Tox

Absorption, Distribution, Metabolization, Excretion and Toxicity (ADME-Tox) describes the disposition of a pharmaceutical compound within an organism. These five criteria influence the drug levels and kinetics of drugs exposure to the tissues and hence influence the performance and pharmacological activity of the compound as a drug (Tetko, Bruneau *et al*., 2006).

Consideration of ADME-Tox properties early in the drug discovery process has reduced the time, effort and even prevented late stage failures in drug development. Thus, the need to predict the ADME-Tox early in drug discovery has lead to various high-throughput *in vitro* screening methods. As these *in vitro* HTS methods provide data for compounds that have been synthesized, there is a necessity for dependable and easy applicable *in silico* models that would aid in the process of drug discovery (DeWitte, 2006; Gunturi, Narayanan *et al*., 2006).

4.2 QSAR

QSAR (Quantitive Structure Activity Relationships) is a process for predicting the biological activity of a compound or class of compounds based on its structure and physico-chemical attributes. QSAR provides a method for evaluation of untested or hypothetical compounds for a specific biochemical

function (Smith, 2002). It is based on numerical descriptions of a given set of structures, geometries, energies, electronic and spectroscopic properties and volume, also taking into consideration known biological activity or chemical activity. QSAR uses this information to build up rules and predict possible biological activity. QSAR is an integral part of drug discovery as it eliminates structurally qualified but biologically failed compounds (Borghini, Pietra *et al.*, 2005; Jaworska, Nikolova-Jeliazkova *et al.*, 2005; Leonard and Roy, 2006; Tetko, Bruneau *et al.*, 2006).

4.3 Caco-2, HEP, LogBB

4.3.1 Caco-2

Caco-2 cells are human colonic adenocarcinoma cells that are able to express differentiation features characteristic of mature intestinal cells, such as enterocytes or mucus cells. These cells are valuable *in vitro* tools for studies related to intestinal cell function and differentiation. Caco-2 permeability refers to the passage of drugs through a Caco-2 monolayer.

The following QSAR model (Equation 1.3) describes the Caco-2 permeability:

$$\log P_{app} = 0.008 \times MW - 0.043 \times PSA - 5.165$$

Equation 1.3: Caco-2 permeability

Where $\log P_{app}$ is the logarithm of the apparent permeability (cms^{-1}) through the monolayer and PSA signifies the polar surface area of the molecule. From the model it can be concluded that permeability is directly proportional to the molecular weight (increase in MW results in an increase in permeability) and inversely proportional to PSA (decrease in PSA results in an increase in permeability) (Clark and Pickett, 2000).

4.3.2 Human Effective Permeability

The rate and extent of intestinal drug absorption can be described by means of the effective intestinal permeability (P_{eff}).

Partial Least Square (PLS) analysis on calculated molecular descriptors resulted in a model to describe effective permeability (P_{eff}) as written in Equation 1.4. The model includes PSA values

together with the number of hydrogen-bond donors (HBD) (Lennernas, 1997; Lennernas, 1998; Winiwarter, Bonham *et al.*, 1998; Clark, 1999; Clark and Pickett, 2000).

$$\log P_{eff} = -2.546 - 0.011 \times PSA - 0.278 \times HBD$$

Equation 1.4 : Human Effective Permeability

As with equation for $\log P_{app}$, this model shows that diminishing the number of polar functional groups (hydrogen bonding) in a molecule would favor passive absorption.

4.3.3 Blood-Brain-Barrier (BBB)

The homeostasis of the central nervous system (CNS) is maintained by separating the brain from systemic blood circulation. This is accomplished by a complex cellular system called the blood-brain-barrier (BBB). If a particular drug is targeted at the CNS BBB-penetration is essential, on the other hand drugs aimed at other sites of action could have unwanted side-effects if passed through the BBB. Hence, determining BBB-penetration with in drug discovery research is of great importance (Abraham, Takacs-Novak *et al.*, 1997; Blake, 2000; Clark and Pickett, 2000).

LogBB can be defined as the ratio of the steady-state concentrations of a drug molecule between the brain and blood. Values published for logBB fall in the range -2.00 to + 1.00.

Molecules with logBB > 0.3 are found to cross the BBB spontaneously, while those with loggBB < -1.00 are poorly distributed to the brain.

A straightforward two-variable equation (Equation 1.5) that was obtained by PLS to describe logBB has been reported(Clark and Pickett, 2000):

$$\log BB = -0.0148 \times PSA + 0.152 \times ClogP + 0.139$$

Equation 1.3 : LogBB

As in the equations for $\log P_{app}$ and $\log P_{eff}$, LogBB can also be related to PSA and ClogP, ClogP being the calculated octanol-water partition coefficient for the particular molecule.

4.4 Methods of Prediction

Jaworska *et al* (2005) proposed a method based on similarity of molecules in descriptor space. This method of approach can be qualified in to several classes. These are range-based methods, geometric methods, distance-based and probability density distribution range methods

Range-based methods are based on the assumption that similar molecules have similar properties. Geometric methods involve the determination of a convex hull or convex envelope. This would be the smallest convex region taking into consideration all points from the training set. Distance based methods use the approach of calculating the distance between the test set of compounds and the training set. Probability density distribution range methods are more complex and computationally expensive than what has been described above. This method is used to distinguish dense and low populated regions of structural space (Tetko, Bruneau *et al*., 2006).

The accuracy of which QSAR can be predicted depends upon the presence of all fragments required. Thus, the accuracy decreases when certain of these fragments are not present in the training set or even when their frequencies are very low. It is thus not possible to calculate a statistically significant coefficient that would represent these fragments.

5. Structural Similarity and Dissimilarity and Tanimoto Coefficient

5.1 Structural Similarity and Dissimilarity

The concept of molecular diversity is not a new concept as Richon and Rouvray have shown that molecular diversity ties in with chemical and physical science throughout the ages. In 1842 Kopp reported the relationship between structure and physiochemical properties of molecules and in 1864 Richardson produced similar work for organic molecules. It wasn't until 1947 that it was possible to show a relationship between physiochemical properties and structures by using different descriptors (Maldonado, Doucet *et al*., 2006).

Although the term molecular similarity has had many definitions in science, the use of similarity concepts has become well established in the chemical and pharmaceutical industry. Similar molecules tend to have similar properties. The molecular similarity between molecules can be calculated in numerous ways. Measuring similarity involves three focal components. Firstly structural descriptors, these describe the characteristics of a molecule. Secondly the weighting schema, this is used to distinguish between the more import characteristics and the less import characteristics and thirdly a method of quantifying the degree of similarity between pairs of molecules, these are called similarity coefficients. In the last 50 years vast arrays of methods have

been developed to calculate molecular similarity. These methods differ from each other in their definition of diversity space and the use of molecular descriptors. The most widely used similarity coefficients are the Tanimoto coefficients and the Euclidean distance which are unweighted, but may have consistent descriptors (Willett, 1998; Willett, 2000). Further details on similarity will be discussed later.

5.2 Similarity Approaches in Chemoinformatics

An in depth discussion of approaches on similarity is beyond the scope of this review thus Table 1.2 provides a short summary. For a more detailed discussion see the work done by Maldonado *et al* (Maldonado, Doucet *et al*., 2006).

Table 1.2 : Similarity approaches in chemoinformatics. Similarity approaches as with molecular descriptors can be classified according to dimensions.

1D-representation	2D-representation	3D-representation	Merging-representation
Physiochemical Properties: Associates a molecule to a single value (LogP, MW, dipole, etc.)	**Connection Tables 2D-Structure:** Representation by means connection tables.	**Molecular Shape Approach:** Involves the superposition of molecular shapes.	**Using Multidimensional Descriptors:** Models that use n-dimensional chemical space.
Molecular Code: Representation of chemical information by means of 1D molecular code or line notation.	**(Electro)topological:** Representation of the electrical properties of atoms.	**CoMFA (Comparative Molecular field analysis)**: Associated the information from molecular field analysis with molecular properties.	**Merging**: Pharmacophores and Fingerprints.
	Graphs, Sub-graphs and Reduced Graphs: Representation of a molecule as a function of their structure.	**Molecular Surface Characteristics**: Molecular features by some representation or projections of 3D surface.	
	Molecular Environment: Representation of a molecule as a function of their ordered atomic/fragments environment.	**QSCD (Quantized Surface Complimentarily Diversity)**: Measures molecular diversity by measuring molecular complementarities to a full enumerated set of theoretical target surfaces.	
	Digital Image Processing: Digital Image processing is used in order to determine similarity.	**MQS (Molecular Quantum Similarity):**By use of the density function of a chemical system, constructed in a precise internal state.	

5.3 Coefficients

Similarity coefficients are functions that transform a pair of compatible molecules into real numbers providing a quantitative measure of similarity (Maldonado, Doucet *et al.*, 2006). Examples of similarity coefficients often used in chemoinformatics can be seen in Table 1.3.

Table 1.3 : Examples of similarity and diversity coefficients. For evaluating the similarity between two molecules with the formula listed, (a) represents the properties of the first molecule and (b) the second, (n) is the total number of properties, (c) is the number of common properties and (d) the number of uncommon one between the two molecules (Maldonado, Doucet et al., 2006).

Index/Coefficient/Distance	Expression
Tanimoto	$S_T = \frac{c}{a+b-c}$
Cosine	$S_C = \frac{c}{\sqrt{ab}}$
Squared Euclidean	$S_E = \frac{a+b-2c}{n}$
Russell-Rao	$S_R = \frac{c}{n}$
Forbes	$S_F = \frac{cn}{ab}$
Simpson	$S_F = \frac{c}{\min(a,b)}$
Yule	$S_Y = \frac{nc-ab}{cd+(a-c)(b-c)}$
Pearson	$S_P = \frac{ac-ab}{\sqrt{nab(a-b)(n-a)}}$
Dennis	$S_D = \frac{nc-ab}{\sqrt{nab}}$

*a represents the properties the first molecule and b the second molecule, n is the total number of properties, c is the number of common properties and d the number of uncommon between the two molecules.

The following section will explain some basic mathematical principles of similarity coefficients.

Object A can be described by a vector X_A of n attributes such as Equation 1.6:

$$X_A = (x_{1A}, x_{2A}, x_{3A}, \ldots, x_{jA}, \ldots, x_{nA})$$

Equation 1.6 : Vector, where X_A attribute vector describing object A, X_{jA} value of the jth attribute in object A

Where x_{jA} is the *j*th attribute of object A. The value of the attribute may be a real number over any range and may also involve a certain weighting factor. It can also be defined in terms of dichotomous values such as binary where 0 would represent absence and 1 presence. In the case of molecular objects, attributes may be a set of *n* topological indexes, calculated physiochemical properties, or even the on/off state of each of the *n* bits in a fingerprint.

5.3.1 Tanimoto Coefficient

The use of the Tanimoto coefficient has become popular as a coefficient of similarity in chemoinformatics.

The Tanimoto coefficient is an extended Jaccard coefficient (Equation 1.7). The Jaccard similarity coefficient is a statistic used for comparing the similarity and diversity of sample sets. The Jaccard coefficient is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Equation 1.7 : Jaccard Coefficient.

The Jaccard distance, which measures dissimilarity between sample sets, is obtained by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union, or simpler, by subtracting the Jaccard coefficient from 1 as done in Equation 1.8:

$$J\delta(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

Equation 1.8: Jaccard Distance

For two objects, A and B, each with n binary attributes, the Jaccard coefficient is a useful measure of the overlap that A and B share with their attributes. Each attribute of A and B can either be 0 or 1. The total number of each combination of attributes for both A and B are specified as follows:

The Jaccard similarity coefficient, J, is given by Equation 1.9:

$$J = \frac{M11}{M01 + M10 + M11}$$

Equation 1.9 : Jaccard Similarity Coefficient

The Jaccard distance, J', is given in Equation 1.10:

$$J' = \frac{M01 + M10}{M01 + M10 + M11}$$

Equation 1.10 : J'

Table 1.4 : Variable definitions

Variable	Description
M_{11}	Represents the total number of attributes where A and B both have a value of 1.
M_{01}	Represents the total number of attributes where the attribute of A is 0 and the attribute of B is 1.
M_{10}	Represents the total number of attributes where the attribute of A is 1 and the attribute of B is 0.
M_{00}	Represents the total number of attributes where A and B both have a value of 0.
$M_{11} + M_{01} + M_{10} + M_{00} = n$.	Each attribute must fall into one of these four categories.

Cosine similarity is a measure of similarity between two vectors of *n* dimensions by finding the angle between them, often used to compare documents in text mining. Given two vectors of attributes, *A* and *B*, the cosine similarity, θ, is represented using a dot product and magnitude as in Equation 1.11:

$$\boldsymbol{\theta} = \arccos\frac{\boldsymbol{A}.\boldsymbol{B}}{||\boldsymbol{A}||\,||\boldsymbol{B}||}$$

Equation 1.11 : Cosine Similarity

Since the angle, θ, is in the range of [0,π], the resulting similarity will yield the value of π meaning exactly the opposite, π / 2 meaning independent, 0 meaning exactly the same and in-between values indicating intermediate similarities or dissimilarities.

This cosine similarity metric may be extended such that it yields the Jaccard coefficient in the case of binary attributes and can be represented as the Tanimoto coefficient of *T*(*A*,*B*) (Equation 1.12):

$$\boldsymbol{T}(\boldsymbol{A},\boldsymbol{B}) = \frac{\boldsymbol{A}.\boldsymbol{B}}{||\boldsymbol{A}||^2 + ||\boldsymbol{B}||^2 - \boldsymbol{A}.\boldsymbol{B}}$$

Equation 1.12 : Tanimoto Coefficient

Table 1.5 shows variation of the Tanimoto coefficient applied to chemical information.

Table 1.5 : Tanimoto coefficient used in chemoinformatics.

Continuous variables	**Dichotomous variables**
$S(A,B) = \frac{\left[\sum_{j=1}^{j=n} xjA\,xjA\right]}{\left[\sum_{j=1}^{j=n}(xjA)^2 + \sum_{j=1}^{j=n}(xjB)^2 - \sum_{j=1}^{j=n} xjA\,xjB\right]}$	$S(A,B) = c/[a+b-c]$
Range from -0.333 to +1	Range from 0 to +1

The equations found in Table 1.5 are derived from Equation 1.13 and Table 1.6 explains the logic behind this:

$$\boldsymbol{a} + \boldsymbol{b} - \boldsymbol{c} = |\boldsymbol{XA} \cup \boldsymbol{XB}|$$

Equation 1.13

Table 1.6 : Definition of $a + b - c = |XA \cup XB|$ [35].

Attribute values are restricted to 0 and 1. If stated that objects A and B are characterized by vectors XA and XB respectively containing n binary values. Thus: $a = \sum_{j=1}^{j=n} xjA$ Number of bits "on" in A $b = \sum_{j=1}^{j=n} xjB$ Number of bits "on" in B $c = \sum_{j=1}^{j=n} xjAxjB$ Number of bits "on" in both A and B $d = \sum_{j=1}^{j=n} (1 - xjA - xjB + xjAxjB)$ Number of bits "off" in both A and B Hence: $n = a + b - c + d$	The definition of *a* and *b* shown here are those commonly used in chemical information. If we then define *XA* as the set of all elements *x*jA in vector XA whose values is 1 (on) and *XB* as the set of all elements *x*jB in vector XB whose values is 1. Then: $a = \|XA\|$ $b = \|XB\|$ $c = \|XA \cap XB\|$ $d = n - \|XA \cup XB\|$ And as the number of bits "on" in at least one molecule can be given as: $a + b - c = \|XA \cup XB\|$ (P Willet, 1998)

The popularity of the Tanimoto coefficient is due to the fact that Hamming and Euclidean distances are only useful for relative distance comparison, that is the distance of two molecules to the same target, where Tanimoto can be used for the absolute comparison of two independent molecules as reported by James *et al.*

It should be remembered that small molecules are most likely to have few bits set in a fingerprint. Since the Tc does not account for a common absence of features and $c \leq min(a,b)$, it can be expected that low-similarity values will be obtained and this could lead to a biased sized distribution in the library. A solution to this was developed by Pharmacia and Upjohn and entails the use of a composite coefficient. This involves both the Tc and simple matching coefficients which is the complement of the normalized Hamming distance (Maldonado, Doucet *et al.*, 2006).

In a study by Adamson and Bush, Willet and Winterman, Tc performed superior to that of Hamming and Euclidean distances. Tc was favored due to subjective evaluation of similarity search rankings and also the fact that the calculation does not involve a square root, making it faster (Maldonado, Doucet *et al.*, 2006).

6. Substructure Searching

As was mentioned before, similar molecules tend to often have similar properties. Extending this train of thought, molecules that possess similar structural moieties would also often have similar properties. Substructure searching relates to the presence of a structural moiety within a molecule. Extracting substructures from target molecules consists in essence of molecular graph handling. Substructure searches are based on subgraph isomorphism or common-subgraph-isomorphism and maximum-common-subgraph-isomorphism (MCS). Figure 1.5 shows and example of isomorphic graphs.

Subgraph isomorphism involves testing a series of topological graphs for the existence of a subgraph isomorphism within a specific query graph. In other words a subgraph isomorphism exists if all the nodes (atoms) of one graph (G_Q) can be mapped to a subset of the nodes of the other graph (G_F) in such a way that the edges (bonds) of (G_Q) simultaneously map to a subset of the edges in (G_F).

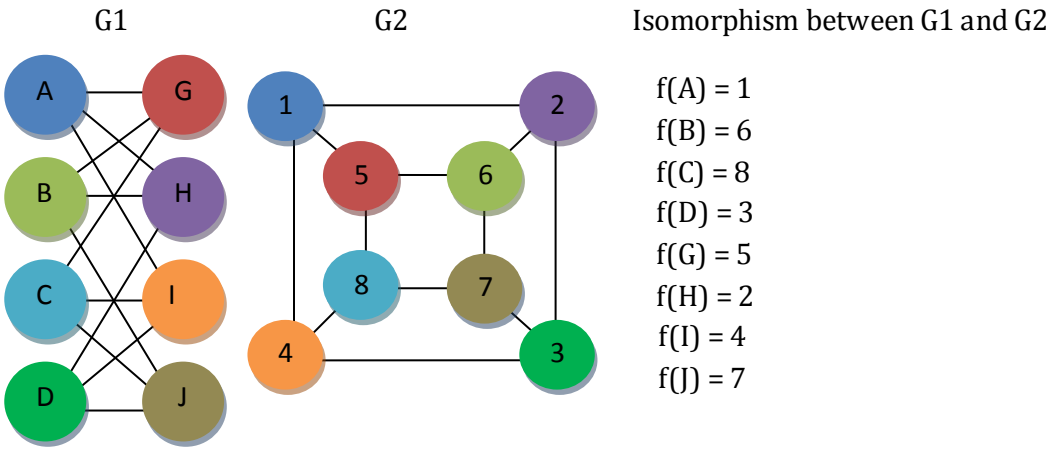


Figure 1.5 : Example of two graphs that are isomorphic.

A formal definition of maximum-common-subgraph-isomorphism can be given as follows: If G1 and G2 are the two input graphs, what is the largest subgraph of G1 isomorphic to a subgraph of G2? One possible solution for this problem is to build a modular product graph, in which the largest clique represents a solution for the MCS problem.

A modular product of graph G1 and G2 is such, that the vertex set of the modular product of G1 and G2 is the Cartesian product, V(G1) x V(G2) and also any two vertices (u, v) and (u', v') are adjacent in the modular product of G1 and G2 if and only if, either u is adjacent with u' and v is adjacent to v' or u is not adjacent to u' and v is not adjacent to v'.

The most commonly used approach to achieve CSS search is the Lesk's algorithm, which uses a geometrical approach directed towards the search for a pre-defined structural moiety (Willett, 2000; Maldonado, Doucet *et al*., 2006).

7. Open Source, the Web and Chemoinformatics

The current price tag to bring a new drug on the market is approximately US $800 million, as reported by DiMasi *et al* (DiMasi, Hansen *et al*., 2003). Looking at these figures, it is understandable that pharmaceutical companies are looking for methods that would optimize the costs connected to Research and Development (R&D) (Geldenhuysa, Gaaschb *et al*., 2006). The combination of combinatorial chemistry and HTS was quickly adopted as a method to reduce costs. Nevertheless, setting up a combinatorial chemistry program and HTS is still costly and is not able to consider the specific needs of many biological systems (Geldenhuysa, Gaaschb *et al*., 2006). *In silico* methods are becoming an attractive alternative in the identification of new lead compounds with promises of nearly a 50% decrease in cost and a decrease in time. Computer-aided drug design software is created by a variety of different vendors. These include commercial companies, academic institutes and open-source software developers. Free and open-source software have gained popularity over the last few years, resulting in an increase in the number of web sites that house such packages, one example being www.sourceforge.net.

Mostly academic scientists find open-source software appealing, because they require less funding to invest in their projects, as commercially available packages are often accompanied by a prohibitive price tag. Apart from being freely obtainable, other advantages of open-source packages are the ability to rapidly download packages from the Internet and customize them according to a particular project. Open-Source software is not without faults and users as well as developers should be wary of these pit-falls. Two of the most common problems are that open-source software may not be well written and poorly documented which in the end could be problematic for user

interaction. Open-source software has made a slow impact on drug discovery compared to bioinformatics in terms of the number of available packages (Geldenhuysa, Gaaschb *et al*., 2006).

The increasing power of web applications supported by advanced scripting languages, such as Java and even Python, has made it possible to equip researchers with powerful and easy to use molecular processing at their desk through web browsers. The use of web-based applications has to some extent overcome the problems of multiple operating systems within a single institution, as web-based applications are platform independent and only rely on a compatible browser. Maintenance on such software is done on a single server compared to several desktop computers. These internet tools would mostly not require any special training, benefiting the novice user (Ertl, 1998; Bembenek, Tounge *et al*., 2004).

Not all chemoinformatics applications can effectively be used by means of a web-based application, especially with regard to advanced visualization. To have an effective and easy to use web application the user interface should be as simple as possible. But the use of a simple interface does not preclude a complex processing back-end. It should be kept in mind that as web-based applications are used by bench biologists with a sometimes limited knowledge of chemoinformatics, great care should be taken that only valid, useful results are displayed in a selective fashion (Ertl, Muhlbacher *et al*., 2003).

The use of web-based chemoinformatics applications offers great potential for biological researchers, which may be further enhanced when deployed as an integrated part of a larger information management system.

8. Conclusion

The major driving forces behind the influence of chemoinformatics in drug discovery, are the large increase in chemical data obtained from methods such as HTS and the economic pressure to decrease time and money put into drug discovery and development. A variety of chemoinformatics methods that have an influence in drug discovery have been discussed, and it is a combination of these methods that make chemoinformatics so powerful. Due to the large amount of chemical data being produced, methods have been developed with which these large amounts of data can be reduced and grouped into manageable subsets. The grouping of chemical data into subsets is based on the principle of similarity, which is well known in various areas of science. Several different approaches have been developed to solve similarities between molecules as can be seen in Table 6. The most popular similarity coefficient that provides a quantitative measure of similarity is the Tanimoto coefficient. The popularity of the Tanimoto coefficient is due to the fact that it can be can be used for the absolute comparison of two independent molecules as reported by James *et al* 2005.

A few years ago the term chemoinformatics emerged and swiftly gained extensive usage. So why did this term become so popular and why is it needed? Theoretical calculations based on first principles are often unable to solve problems in chemistry that are too complex, an example of this is the relationship between the structure of a compound and its biological activity. For this particular case the known experimental data is analyzed and a representive model is built. Another important factor is that chemistry produces vast amounts of data by methods such as combinatorial chemistry and high-throughput screening. The need for a discipline such as chemoinformatics can be summarised by the above mentioned examples.

Over the past 40 years great advances have been made such that the extent of this field has recently been documented in a *Handbook of Chemoinformatics*, which covers 73 contributions by 65 scientists over a course of 1850 pages in four volumes (Gasteiger, 2003).

It has been realised that chemoinformatics has gained such importance, that a few universities have integrated chemoinformatics topics in other disciplines and even provide a full chemoinformatics curricula to satisfy the need.

It is clear that chemoinformatics has come a long way since the first representation of chemical data. It can be speculated at what stage of development chemoinformatics is at this moment, but of more importance is that chemoinformatics has made a huge difference in the approach of drug discovery by introducing a knowledge based approach.

Incorporating selected chemoinformatics methods and principles discussed in Chapter 1 into an integrated web application could indeed provide biologist with an easy-to-use chemoinformatics tool, which could be used to explore the chemical space of a particular system and/or pathway and in the end transforming data into knowledge.

9. Problem Statement

The need for a new anti-malarial drug is clear, and the polyamine pathway in malaria has been recognized as a potential target for chemotherapy agents. It is thus desirable to be able to generate libraries of compounds that may be tested for inhibition of enzymes in the polyamine pathway. By exploring the chemical space of the polyamine pathway of Plasmodium using a chemoinformatics-based approach, insight could potentially be obtained into possible polyamine analogues in terms of properties such as molecular weight, polar surface area (PSA), molecular refractivity (MR), LogP and others that have been proven to be important in the identification of lead compounds and drug discovery.

A web-based chemoinformatics application was developed in the context of the FunGIMS suite of tools (to be described), specifically providing biological users with some of the most common

chemoinformatics functions such as (1) the representation of chemical compounds, (2) chemical data, (3) databases and data sources, (4) structure search methods, (5) methods for calculating physical and chemical data, ADME-Tox or QSAR, (6) calculation of structure descriptors.

This system was then used to perform an investigation of compounds in the Plasmodium polyamine pathway.

10. Specific Goals

1. The development of a suitable data model to store small molecule information and integrate this in to the extensive data model of FunGIMS (Chapter 2).
2. The design and develop a chemoinformatics module in the context of the FunGIMS system that would allow biologists to perform chemoinformatics analysis and manage their small molecule information (Chapter 2).
3. An exploration of the polyamine pathway in *Plasmodium* in order to develop a library of compounds with specific structural and chemical properties for further investigation in docking studies (Chapter 3).

Chapter 2

Design and Implementation

This chapter describes the development of a data model and web-based application for chemoinformatics, in the context of the greater FunGIMS suite of tools (to be described). The chemoinformatics module can be divided into the following four components 1) a molecular viewer, 2) a search tool, 3) a molecular descriptor calculator and 4) management area. The molecular viewer component displays information and 3D-structure information regarding a specific molecule. The use of SMILES enables the search tool to perform similarity and substructure searches. For every molecule viewed a series of molecular descriptors are calculated from SMILES, these molecular descriptors are used for identifying specific drug-like molecules. The ability to create a library of results obtained from a particular search provides a means of organizing and managing libraries and the ability to download a library as a Comma Separated File, SDF or SMILES-file provides compatibility with other systems. Developing a library of molecules is not useful if the molecules within the library do not possess the required properties. Thus, by adding a filtering functionality a library can be customized with regards to certain criteria specified by the user.

1. FunGIMS

1.1 Overview of FunGIMS

The Functional Genomics Information Management System (FunGIMS) aims to create a web-based environment where biological researchers can manage a series of different functional-genomics data types, which include public and user-generated data. Additionally, it provides functionality to perform common types of analysis using existing applications which have been intergrated into the system. FunGIMS consists of a core module, together with various data-specific modules for different functional genomics data types that are commonly used by biological researchers. Core module functions include user authentication, group management and database access. Data-specific modules include general data, sequence data, microarray data, genotyping data, structural data, comparative genomics/phylogenetics data and small molecule data.

Chemical data has been presented in some format for many years being either printed or *in silico* as is currently the standard. Chemical or small molecule data now functions under the name

chemoinformatics (Chapter 1) which has become a key component of chemical research in modern science. Chemoinformatics now deals with the handling of large amount of chemical data in order to discover compounds that may have specific biological functions. Due to this the importance of small molecule data could not be ignored when it came to deciding what data types should be handled by FunGIMS.

The FunGIMS system was found to be a stable platform to integrate the small molecule module into. The reasons and advantages of this can be seen in the following list:

1. Inheritance of core functionalities such as security, user privileges to name only two.
2. FunGIMS being a web based application.
3. The ease at which a data type can be added to the FunGIMS database.
4. In future versions of FunGIMS data links between data types will be created.

This is indeed not a complete list but shows the advantages of use within the FunGIMS system.

1.2 FunGIMS Technologies

The FunGIMS platform comprises of a series of software technologies. The Python-based TurboGears framework allows for the rapid prototyping of web based applications. The SQLAlchemy Object Relational Mapper was chosen to handle communication with a SQLite database that was used for development and later migrated to a MySQL database server. The use of SQLAlchemy proved to be suitable, as many of the views and retrievals of objects from the database involves complex joins between different tables in order to utilize all the information, and the security model relied on polymorphic inheritance from a common data object. In order to maintain a stable development and source code, version control was performed using Subversion. The entire FunGIMS project is based on a spiral development method which is discussed further in section 4.

The FunGIMS core model supplies data specific modules with a basic framework. In this framework security features, access to public data and private user-generated data, and relations between entries can be inherited. The polymorphic inheritance from the Identifiable class provides the certainty that the necessary safety measures with regard to access management are adhered to. The Identifiable class also serves as the extension point for any other data type that exists within the FunGIMS database.

Security management of FunGIMS is founded on a group access system, alike to the Linux security system. Where groups are created and users within groups have access to the data that belongs to the group. This method of security is also used in the popular TurboGears web-development

system, and this was then preserved to handle security. The registration module allows for a simple registration system in which users can be registered, deleted and managed. The FunGIMS databases is equipped to deal with notes added to entries, relationships between entries, relationship types, uploaded files, descriptions of entries and a search results object.

1.3 Core Functionalities

Various modules are supported by the system core and provide each module with the following functionalities: Access to the system and user registration and management, a common security layer through which access to the underlying data is provided, interactions between the modules.

The built-in TurboGears security system in combination with the registration model provides access to the system. The security system is based on a two tier security model, with data being assigned to users, users belonging to groups, and groups of users being allowed to see the data that belong to the group. The registration model allows for user registration, group creation, and member assignment to groups.

Two major classes are found in the FunGIMS system and all the data can be divided in either one of the two. The first being the *WORLD*-group which possesses only public data, that is data that can be viewed by all users. The second group would be a PRIVATE-group which only allows users of a specific group to have access to data held within that group. This system has been designed to be extremely flexible and was done as simply as possible to avoid unnecessary complications. Thus, there are no limitations to how many groups a user may belong to, or how many members are allowed in a group. Group sizes may vary from 1, where a user has his own private data group, to hundreds, where groups of researchers share data. The deletion of entries is currently limited to the owners of the data, with the public data not being removable. These security features are attached to the *Identifiable* object and access to data can only take place through this object.

A system identifier (sid) is used to establish interaction between different modules. This means that each entry in the system has a unique URI to which all references can be created. For example the ChEBI entry 160 references to Protein Data Bank entry 9jjb and two PubMed entries 159 and 753. Using the relationship link table, these entries are then associated with each other by the respective URIs (Uniform Resource Identifier), pmid:159, pmid:753 and pdb:9jjb

Functional extensions to the system are made available for all modules, thus preventing the reimplementation of analysis tools in the different modules. The basic idea is that for most functional tools it is only necessary to send a *sid* together with the necessary parameters to a function to be able to invoke that particular analysis. This makes a large pool of analysis tools

available to the users and developers of the system. The core system also provides the functionality to create manual annotations to data entries, in which notes regarding the respective entry can be stored. These notes are also protected by the security system of the FunGIMS core.

1.4 FunGims DataModel

The basis for any large information management system is a high-quality data model. The FuGE data model was selected as the starting point for data management in FunGIMs. FuGE (Functional Genomics Experiment) is a model of the common components in dissimilar functional genomics domains. FuGE makes possible the development of data principles in functional genomics (Jones, Miller *et al*., 2007). Due to the fact that many of the functionalities of FuGE were not intended to be used it was decided to base the FunGIMS data model on FuGE instead of adapting the FuGE model. This resulted in the FunGIMS data model which is able to easily handle notes added to entries as well relationships between data types which is not available within the FuGE data model. For the specific modules various databases schemas were consulted and relevant parts modified and incorporated into the FunGIMS data model.

1.5 FuGE

Every entity of the FuGE data model is a descendent of either one or both of two base classes (Figure 2.1) which are themselves arranged in a hierarchy. The top class in the hierarchy is Describable and all classes in FuGE extend from here.

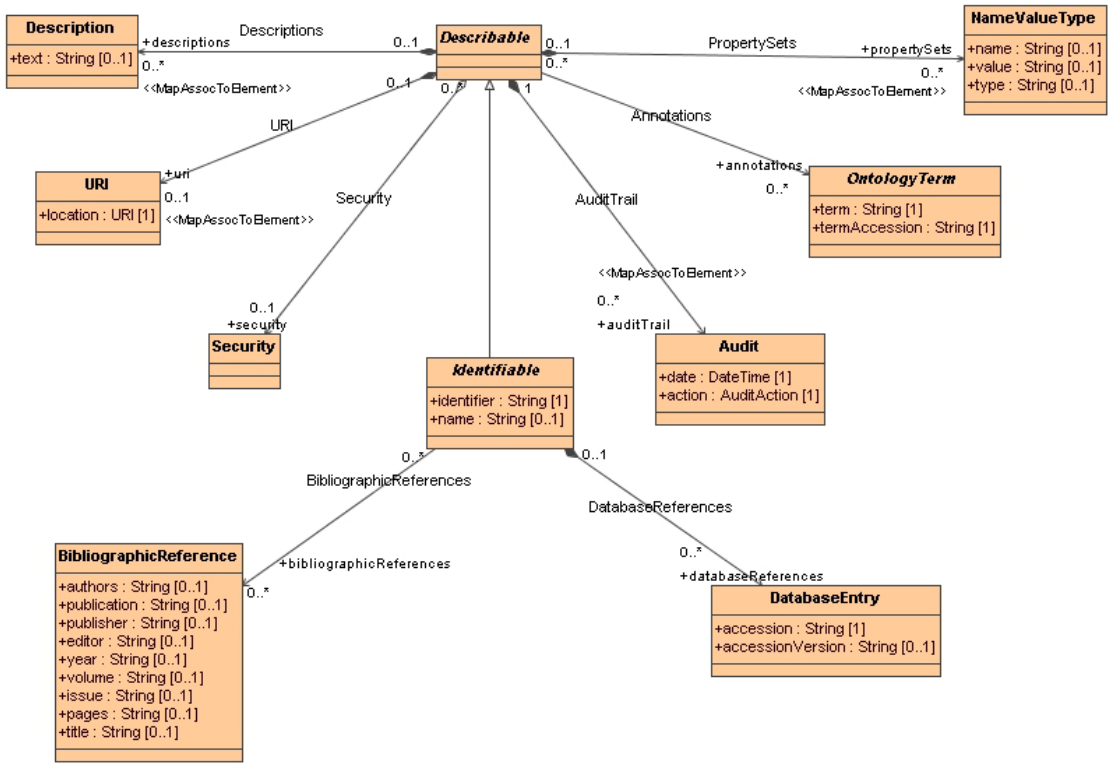


Figure 2.1 : The hierarchy of base classes in FuGE.

The other class in FuGE such as, Identifiable inherits from Describable and adds a referencing mechanism. Classes that inherit from Identifiable are able to be referenced by other classes using the identifier attribute. The identifier attribute is understood to be a globally unique string that resolves a particular instance of the object. The "globally unique" restriction implies that there is exactly one and only one object with a particular identifier. The name attribute stores a human readable name for the object that need not be unique. Classes that inherit from Identifiable can have associations to external database entries or bibliographic references.

2. The Chemoinformatics Module (Small Molecule Module).

The chemoinformatics module aims to incorporate selected chemoinformatics methods and principles discussed in Chapter 1 into an integrated web application that would provide research biologists with an easy-to-use chemoinformatics tool, which could be used to transform data into knowledge.

The main goal is to provide research biologist with some of the most common chemoinformatics functions such as (1) the representation of chemical compounds, (2) chemical data, (3) databases and data sources, (4) structure search methods, (5) methods for calculating physical and chemical data, ADME-Tox or QSAR, (6) calculation of structure descriptors. These functions allow the user to create and store chemical libraries that possess compounds that have the required properties.

Along with all the above mentioned functionalities that can be preformed the chemoinformatics module also aims to allows user to store their chemical informatics in a central place in an organized manner.

3. Molecular Structure Representation

Chemical compounds can be represented *in silico* in a variety of manners depending on what information is required and what the usage will be. It is needless to say that the same molecule can be represented in different ways depending on the situation. For example, representation of a molecule for the purpose of quantum chemistry and drug discovery differ as both require different types of information.

The following section describes the representation of chemical compounds *in silico* as incorporated into the chemoinformatics module of FunGIMS.

3.1 Structure-Data-Files (SDF)

Structure Data Format (SDF) files are simple text files that adhere to a strict format for representing multiple chemical structure records and associated data fields. The general format of an SDF file consists of blocks of information with a single compound record format represented by Figure 2.2 below (Dalby, Hounshell *et al.*, 1992).

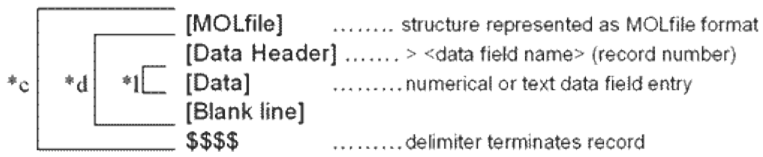


Figure 2.2 : Schematic of SDF. C : Compound record format for the length of the SDF. D: Data item format is repeated for each data item associated with a compound record. L : A separate line is used for each data value. MOL-file format is format given by MDL formatting for the storage of chemical structure information (Dalby, Hounshell *et al*., 1992).

Figure 2.3 shows a sample of a SDF file of two small molecules, 1,2-trans-dichloroethene and bromochloroacetonitrile, each containing 4 data fields.

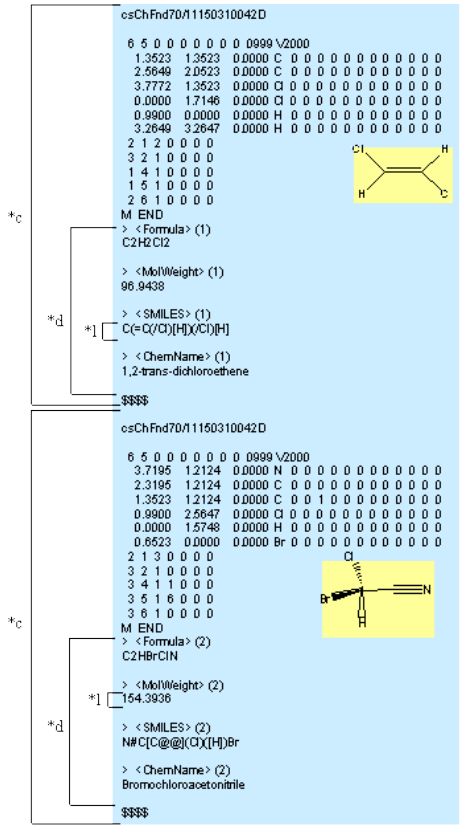


Figure 2.3 : SDF of 1,2-trans-dichloroethene and bromochloroacetonitrille. A single SDF can have more than one molecule present as long as they are properly separated. As is the case here where 1,2-trans-dichloroethene and bromochloroacetonitrille are separated by "$$$".

3.2 Tripos Mol2 Format

Mol2 files by Tripos are ASCII files that represent molecules in 3D accompanied by a large amount of molecular attributes. A total of 32 unique attributes can be found in a mol2 file. Each attribute is found in a single subsection within the file. Record Type Indicators (RTI) are used to identify each attribute subsection. In the case of mol2 files this is a line starting with "@" followed by the attribute name and ending with an end line character. An attribute subsection is terminated with a new RTI or when the end of the file is reached (Tripos, 2005).

3.3 SMILES Representation

SMILES (Simplified Molecular Input Line Entry System) is an intuitive notation for the entering, representation and computation of molecules and molecular reactions *in silico,* developed by David Weininger (David, 1988). In other words, SMILES structures are linear strings of characters that present chemicals as graphs, where atoms are represented as vertices and bonds between them as edges. The nature and simplicity allows novice users to easily and accurately represent their molecules as SMILES by following the simple rules regarding the construction. Below, the general rules regarding SMILES construction is explained.

Atoms

In SMILES atoms are represented by their atomic symbols enclosed in square brackets normal atoms are presented in uppercase and aromatic atoms are presented in lowercase as can be seen in Figure 2.4.

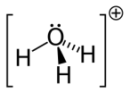
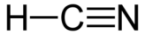
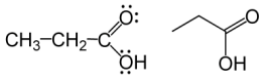
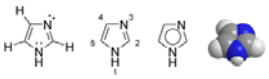
Rules Regarding	Structure	SMILES-structure	Description
Atoms		[OH2]	Water(H_2O)
Additional information		[OH3+]	Hydronium cation
Bonds		C#N	Hydrogen cyanide
Branches		CCC(=O)O	Propionic acid
Cyclic structures		n1c[nH]cc1	Imidazole

Figure 2.4 : Examples of SMILES structure of different molecules illustrating the design rules of SMILES-structures. Water is shown that atoms are illustrated by their atomic symbols as is the case with oxygen (O) and hydrogen(H). A triple bond is represented by # as in hydrogen cyanide. Branched elements are seen in parentheses in propionic acid. Imidazole shows how a cyclic structure is represented.

Additional Atom Information

The charge and number of hydrogens attached to an atom have to be specified within the square brackets. In the case of attached hydrogens these are presented by "H" followed by a digit indicating the number of hydrogens if more than one is present. The charge of an atom is indicated by "+" or "-"symbols. In case of the charge being larger than one, the charge is indicated by sequential positive or negative symbols to the same number as the charge or by a single "+" or "-"symbol accompanied by the number of the charge as in Figure 2.4.

Bonds

Notation for single, double, triple and aromatics bonds are "-", "=", "#" and ":" respectively and can be seen in Figure 2.4. When two atoms are adjacent and no bond is indicated it is assumed that a single or aromatic bond is present, depending on whether the atoms are indicated to be aromatic.

Branches

Branches in molecules are specified by placing the branched substructure in parentheses (Figure 2.4). Sub-branches can be nested in higher level branches, while consecutive branches can be stacked sequentially and indicating multiple branches at one atom position.

Cyclic Structures

Cyclic structures are represented by breaking a bond between two atoms in the cyclic structure and labeling those ring closure atoms at the position with a digit immediately after the atom symbols. If an atom takes part in more than one ring closure, those ring closures are indicated by sequentially adding closure digits to the end of the atom symbol. Double-digit closures have a "%" directly to the left of the digits in the string in order to distinguish them from sequential single digit closures.

Power and Usefulness

The usefulness of SMILES over that of connection tables comes from the fact that SMILES is a linguistic construct rather than a computer data structure. This enables SMILES to be used as "words" in other languages used to store chemical information. In contrast to other methods of representing chemical structure, SMILES are quite compact. For instance a SMILE representation would take up to 50%-70% less space than an equivalent connection table. The power of SMILES structures comes from the existence of unique and standard SMILES. In the case of standard SMILES the name of a molecule is synonymous with its structure, in contrast with unique SMILES where the name is universal (http://www.daylight.com/).

Examples of uses for SMILES are:

- Keys for database access
- Mechanism for researchers to exchange chemical information
- Entry system for chemical data
- Part of languages for artificial intelligence or expert systems in chemistry

3.4 OpenBabel Fingerprints

As mentioned previously mlecular fingerprints are composed of bits of molecular information such as the type of ring, functional groups, and other types of molecular and atomic data. Three different types of fingerprints are found in OpenBabel, these are FP2, FP3 and FP4 (http://openbabel.org/wiki/Main_Page).

3.4.1 FP2

FP2 indexes small molecule fragments based on linear segments of up to 7 atoms in length. Linear fragments of length, 1-7 atoms, are identified by analyzing the structure of a molecule. When atoms form a ring the fragment is terminated, single atoms fragments of C, N and O are ignored. For each of these fragments the atoms, bonding and whether they constitute a complete ring is recorded and saved in a set so that there is only one of each fragment type. Chemically identical versions are identified and only a single canonical fragment is retained. The remaining fragments are a hash number from 0 to 1020 which is used to set a bit in a 1024 bit vector (2006; Guha, Howard *et al.*, 2006).

3.4.2 FP3 and FP4

FP3 and FP4 are fingerprint methods created from a set of SMARTS patterns defining functional groups (2006; Guha, Howard *et al.*, 2006). SMARTS is a language that allows you to specify substructures using rules that are straightforward extensions of SMILES.

3.5 Frowns

Fingerprints generated by Frowns are similar to fingerprints constructed by the Daylight software package. Fingerprints are generated by applying a hash function to a molecule. Hash functions are used to turn data into a relatively small number that serves as a digital fingerprint as in Figure 2.5. A hash algorithm “chops and mixes” (i.e. substitutes or transposes) the data to create such a fingerprint (Bebis, Georgiopoulos *et al.*, 1998).

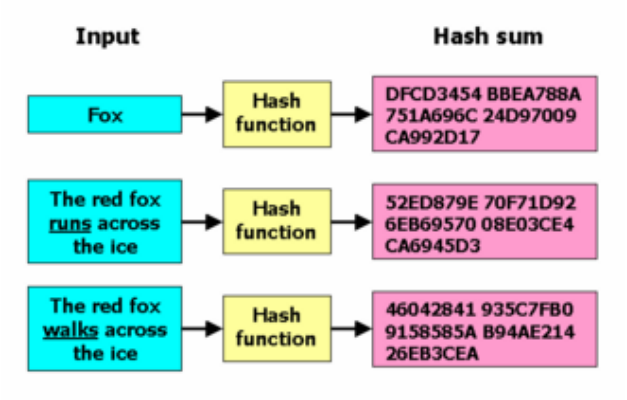


Figure 2.5 : Diagrammatic example of how a hash function works. A hash function is any well-defined procedure or mathematical function for turning some kind of data into a relatively small integer that may serve as an index into an array. The values returned by a hash function are called hash values, hash codes, hash sums, or simply hashes.

The sums of a hash are commonly used as indices into hash tables or intermediate hash files. A hash or Frowns fingerprint is thus a binary vector stored as a list of integers (http://frowns.sourceforge.net/).

4. Technologies Used

This section describes the technologies incorporated into the FunGIMS chemoinformatics module.

4.1 Programming

4.1.1 Python

Python can be described as a general-purpose, high-level programming language, its design attitude highlights programmer productivity and code readability. Multiple programming paradigms, primarily functional, object oriented and imperative, are supported by Python. Python features a fully dynamic type system and automatic memory management. The primary programming language for this project is Python (Python, ; Sanner, 1999). Python was chosen due to its

popularity in the field of bioinformatics making a wide variety of third-party libraries and packages available and due to its code readability future programmers will be able to extend this module.

4.1.2 TurboGears

In order to create a web-based application TurboGears was used. TurboGears allows the creation of powerful data-base driven web-applications written in Python(http://turbogears.org/). TurboGears is a Python web application framework consisting of several underlying components such as MochiKit, SQLObject, CherryPy and Kid (Figure 2.6). MochiKit is an elective piece of TurboGears, a JavaScript library to make programming in JavaScript more Pythonic. It is mostly used for employing Ajax features as it provides an interface to get JSON data streams in an asynchronous manner. CherryPy functions as the middleware that permits web applications to be programmed by writing event handlers that return data to the templates. Kid is a XHTML front-end templating engine where all templates are valid XHTML or XML files that are usually made in a way that allows opening these templates as simple XHTML files to check the design. At the same time features are provided to embed snippets of Python code (http://turbogears.org/).

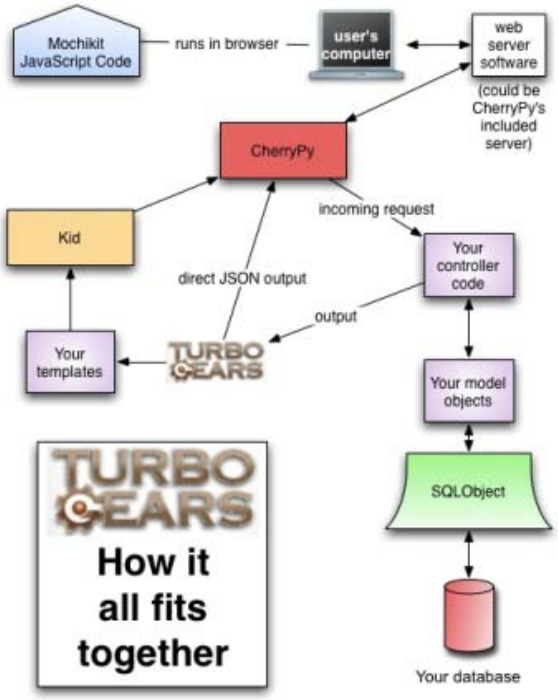


Figure 2.6 : TurboGears: How it all fits together. This diagram illustrates the inner workings of TurboGears and how all the different part fit together to create a web-application (http://turbogears.org/).

4.1.3 SQLAlchemy

Instead of using direct SQL commands to query the database, it was decided that it would be more efficient if entries in the database could be handled as objects. This was achieved by the implementation of SQLAlchemy, which makes it possible to map database tables to object classes, allowing database entries to be handled as objects, thus enforcing an object orientated programming approach (http://www.sqlalchemy.org/). SQLAlchemy was used as many of the views and retrievals of objects from the database involves complex joins between different tables in order to utilize all the information and the security model relied on polymorphic inheritance from a common data object.

4.2 Resources

4.2.1 CHEBI

As a start an existing database of small molecules was implemented. The chemoinformatics module is developed with the intended purpose of being useful primarily to research biologists, thus the database should contain relevant data. Accordingly, CHEBI was chosen as it composes of **Ch**emical **E**ntities of **B**iological **I**nterest (Degtyarenko, de Matos *et al*., 2008).

4.2.2 OpenBabel

To handle chemical data and the representation of small molecules *in silico*, OpenBabel was the chemical toolkit of choice. The capabilities of OpenBabel are numerous and the ability to integrate with Python made the representation and calculation of certain properties possible (2006; Guha, Howard *et al*., 2006).

4.2.3 Frowns

Frowns is another chemoinformatics toolkit written almost entirely in Python with only a small portion written in C++. Frowns is based on PyDaylight written by Andrew Dalke (http://frowns.sourceforge.net/).

4.2.4 Jmol

Jmol is a cross-platform molecule viewer that is freely obtainable by students, educators, and researchers in chemistry and biochemistry. Jmol can be integrated into a web page by means of a JmolApplet. Jmol can also function as a standalone Java application that runs on most desktop computers. Available as a open-source package the JmolViewer functions as a development toolkit that can be integrated into other Java applications. Features of Jmol include: animations, vibrations, basic unit-cell support, schematic shapes for secondary structures, measurements such as distance and torsion angle, RasMol/Chime scripting language support. Jmol also has the capability of exporting .jpg, .png, .ppm, .pdf, and PovRay and supports a wide range of file formats (http://jmol.sourceforge.net/)(Jmol).

4.2.5 JME

JME Molecular Editor developed by Peter Ertl at Comenius University Bratislava and later enhanced at Ciba-Geigy Basel, is a Java applet which allows drawing and editing of molecules and reactions. JME is capable of generating Daylight SMILES or MDL mol files from a 2D chemical structure (http://www.molinspiration.com/jme/).

5. Development

5.1 Development Process

The software development model that fits the development of this particular module, as well as FunGIMS to a great extent, would be the spiral model and to a lesser extent that of extreme programming. The spiral model is a systems development method (SDM) used in information technology (IT) that combines elements of both design and prototyping-in-stages in an effort to combine advantages of top-down and bottom-up concepts. This model of development combines the features of the prototyping model and the waterfall model. The spiral model is often utilized for large, expensive and complicated projects (Barry, 1988).

Implementation of the spiral model approach was done since the start of the project. This was accomplished by first developing a stable integrated data model and only after this was completed

and tested, further development commenced by adding small functionalities at a time and testing these before new functions were added (Figure 2.7).

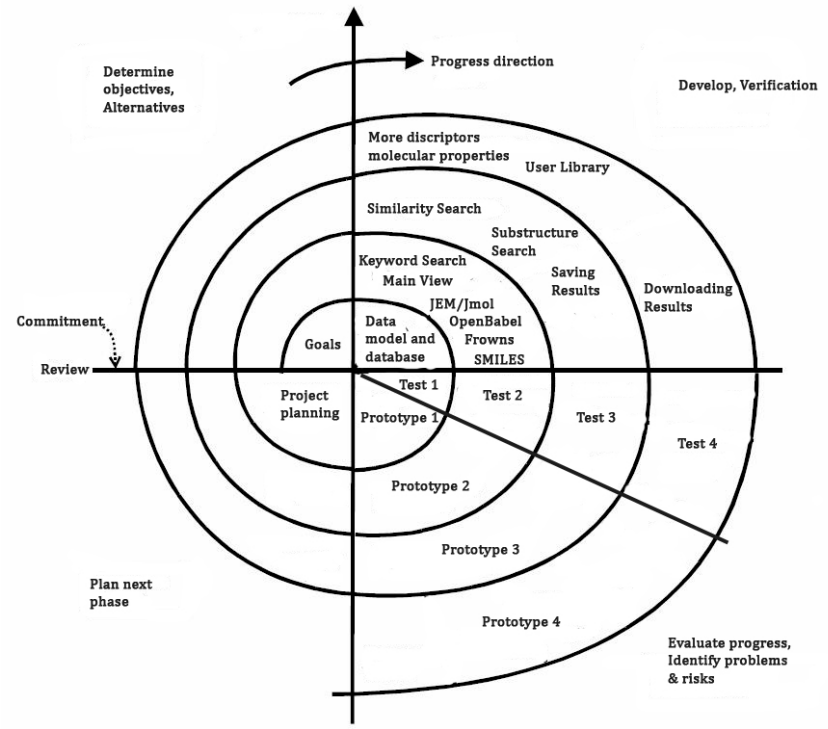


Figure 2.7 : Spiral development model followed during the development of FunGIMS chemoinformatics module. The first step was to develop a working data model, after which additional functionalities could be added as progress through the spiral was made.

5.2 Web Interface

The web interface is responsible for user interaction, gathering of user input and communicating user requests to the underlying analysis and database. Due to the fact that Python was chosen as the major programming language, TurboGears (section 4.1.2) was used to create a front-to-back web-application.

The actual web-pages are Kid templates which are valid XML files, in other words Kid templates are responsible for what the user sees and interacts with. User input or requests are handled by CherryPy, which takes every incoming request and tries to match it up to the object hierarchy within the controller classes. CherryPy is also responsible for passing parameters from the controller class to the Kid templates in order to display the results of a user request.

Jmol is responsible for the viewing of the 3D-structure of molecules where JME takes on the task of user input in terms of 2D-structure.The 2D input structure is then converted onto a SMILES structure by means of JME, which in turn is passed to the controller by means of CherryPy. Jmol and JME where incorporated into the web interface by means of Java applets and Java scripting without any risk due to the nature of Kid templates.

5.3 Python Back-end

The Python back-end to the chemoinformatics module takes on the following responsibilities: accessing the database by means of SQLAlchemy, web application supported by TurboGears and module specific analysis such as similarity searches, substructure searching and calculation of molecular properties. Programming in Python proved to be much easier and faster than it would have been in C++, for example. Unlike older programming languages like C++, Python standard libraries support commonly accepted functionalities such as garbage collection and memory management. Another advantage of developing in Python, which became apparent, was the fact that additional libraries could be found to handle small molecule data as well as commonly used function in chemoinformatics, which refers in particular to OpenBabel and Frowns. Python, being an interpreted language, is a bit slower than executable and compiled languages like C++, however, this can be overcome by enforcing proper programming methodologies. The back-end was development in a Linux environment and done in Eric3, which is a full featured Python editor and IDE.

6. Architecture

6.1 Database Architecture

As mentioned, the data within FunGIMS chemoinformatics module is obtained from CHEBI. The data-model is thus a highly modified version of the CHEBI database, as illustrated in Figure 2.8.

The largest difference between the two data models is the manner in which structural data is handled. In contrast to FunGIMS, CHEBI holds all structural data within a single table called Structure. To increase the specificity of which type structural data is retrieved FunGIMS has two tables that deal with different types of structural data (Figure 2.8). Furthermore the Ontology, References and Comments tables found in CHEBI were removed from FunGIMS data model. The reason for removing the Ontology table is due to the fact that it was decided not to display

ontologies. With regards to References and Comments these were removed as the FunGIMS core functionalities and data model was developed in order to handle such information.

The main entry point for the FunGIMS small molecule data model would be the Compound object class. This inherits from Identifiable and Describable in FuGE this inheritance can be seen in Figure 2.8, thus providing advance features of FuGE which also holds all the references to other data objects.

In addition it stores the CHEBI recommended name and definition. The Compound object is not the only part of the small molecule data-model. The Compound object is complemented by various other objects. The DatabaseAccession object holds all the manually curated database links and registry numbers available in CHEBI. Synonyms and IUPAC names are stored in the CompoundName object and a single Compound object can have more than one CompoundName object referenced to it. The ChemicalData object holds all the formulae of each compound object. The Structure object of the original CHEBI was altered in the sense that it was split up into two different objects namely SMILES and MOL. These objects store SMILES and mol data respectively. The Vertice object links the compound entries to the ontology. A Compound object can have more than one reference to a Vertice object.

The Relation object has two composite aggregations with the Vertice objects. Each aggregation signifies the beginning and the end of a relationship between two vertices.

The final object of the small molecule data-model is the ChemicalLibrary object. This object has no connection with CHEBI and any of the other abovementioned objects. This objects stores user created chemical libraries and the molecules within these libraries. Figure 2.8 shows a UML diagram of the described data model.

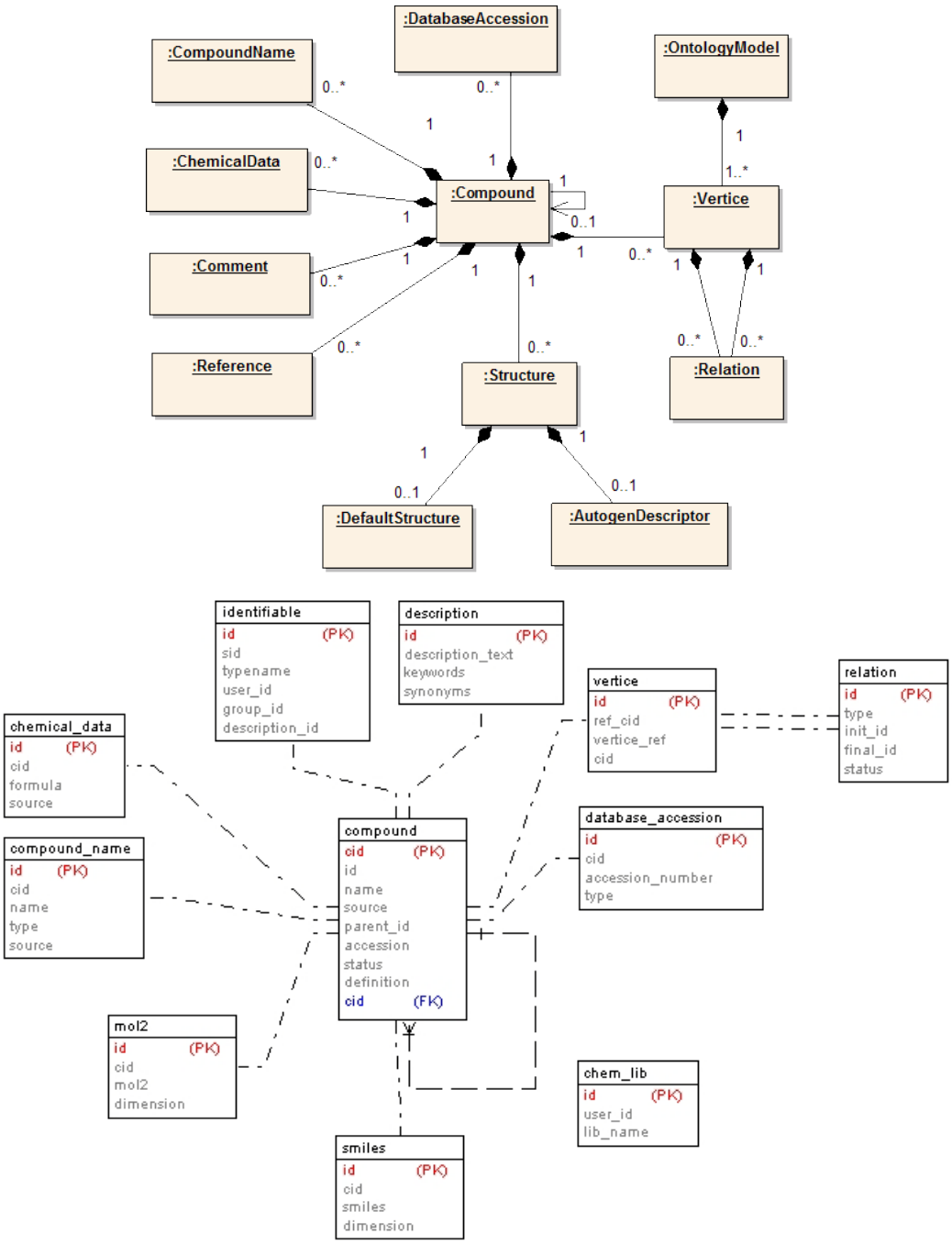


Figure 2.8 : CHEBI-database and Chemoinformatics data model found in FunGIMS database. As with CHEBI the compound table is the primary entry point of small molecule data (Degtyarenko, de Matos *et al.*, 2008). Compound inherits from identifiable and describable and thus also core functionalities (not shown). The preceding table all poses different properties shown.

6.2 Development Architecture

The web interface, controllers, module specific analyses and FunGIMS database are the four main elements that form the higher level architecture (Figure 2.9).

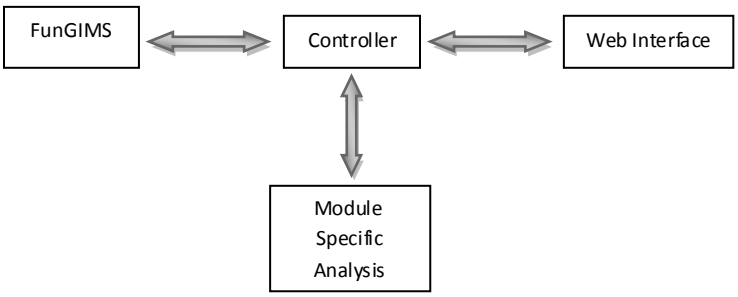


Figure 2.9 : Schematic showing the higher level architecture and how it interacts. This is a very simplified illustration of higher level architecture. Small molecule data is stored in the FunGIMS database as described above by means of SQLAlchemy, and the controller interacting with the database. The controller then passes user query results on the web interface and is displayed to the user. Module specific analysis-obtained query results from the controller processes the data, passes it back to the controller and it is passed to the web page and displayed.

Users interact with a web interface that is constructed of a series of Kid templates which is connected to the TurboGears main controllers by means of CherryPy. The main controller passes on requests and variables to different sub-controllers. Sub-controllers are responsible for views and searches. Sub-controllers can then access the FunGIMS underlying database by means of SQLAlchemy and perform module specific analysis on the retrieved data. Results of analysis can be passed back to the Kid templates again via CherryPy and displayed to the user in the web interface or to be stored in the database.

6.3 Lower level Architecture

The chemoinformatics module is made up of seventeen Kid templates, consisting of ten functions within the main controller as can be seen in Figure 2.10. The function uses two sub-controllers, a

view-sub-controller and a search-sub-controller with thirteen and three functions respectively, see Figure 2.11. It also uses three utility scripts (Figure 2.12). Within the model, the chemoinformatics module is responsible for eleven tables, eleven class and eleven mappers Figure 2.13.

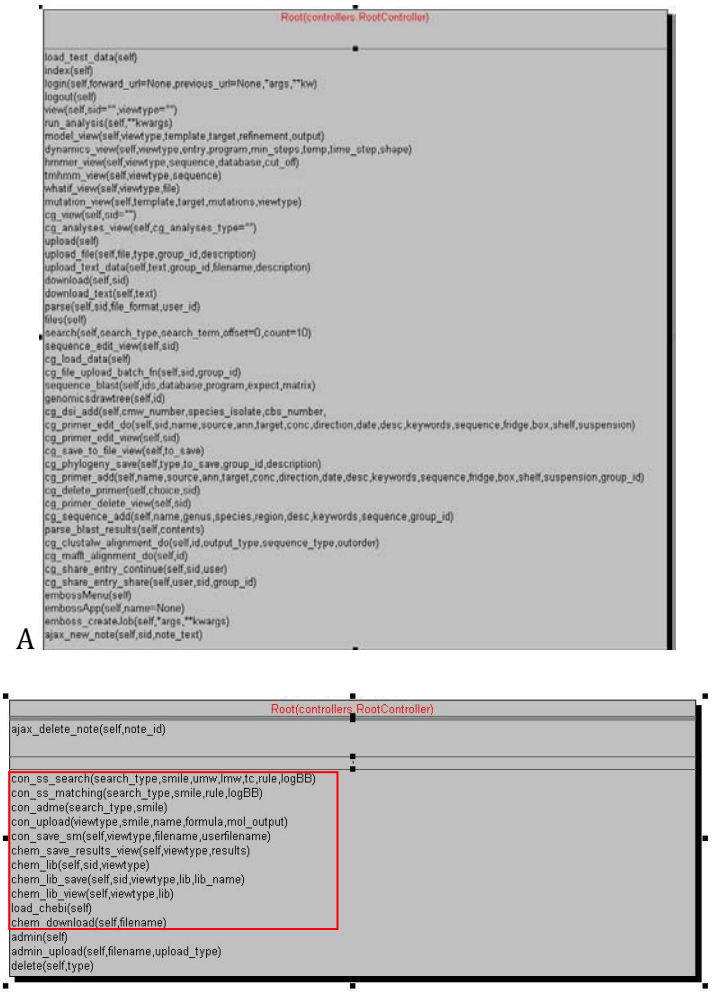


Figure 2.10 : Diagrammatic representation of controller.py. The purpose of the functions (red box) in the main controller is only to pass variables to the sub-controller.

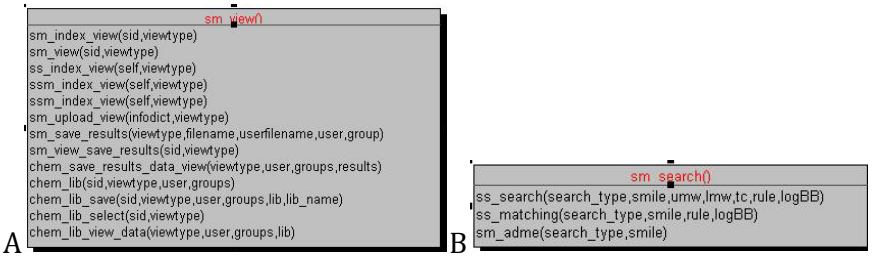


Figure 2.11 : Diagram of sub-controller. The view-sub-controller (A) is responsible for the main view page of each functionality or analysis. The search-sub-controller (B) in return is responsible for searching the database.

Figure 2.12 : Diagram of utility scripts. These scripts possess module specific functions and are imported as needed. Thus can be re-used and to not overcrowd the controllers.

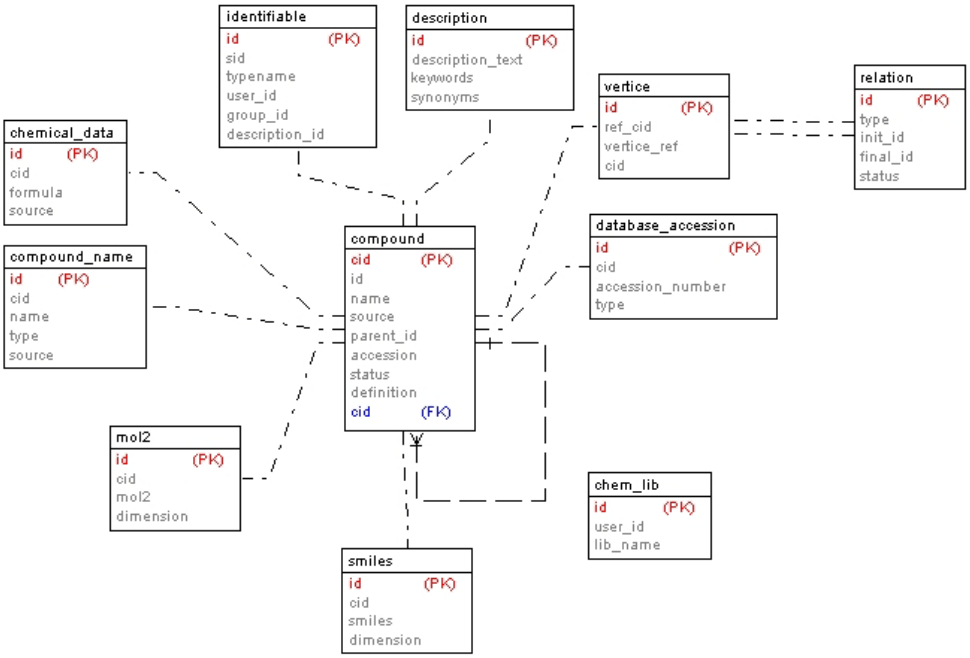


Figure 2.13: Diagram of Data model responsible form chemoinformatics module. The data model is processed by TurboGears and the mySQL database is created, by means of SQLAlchemy the different tables are maped by means of mappers to class found in model.py.

None of the chemoinformatics functions take place within the main controller.py, the sole action of the functions within the main controller.py is to pass variables on to the correct function within one of two sub-controllers. The view sub-controller is responsible for the starting view of all analysis where the sub-search controller is responsible for searching the database for a particular user query. The view controller imports from model.py, controller.py and SQLalchemy in order to gain access to the FunGIMS database and core functionalities. The view-controller also imports ADME.py and SRS.py from the UTILS-directory within the project directory

ADME.py contains a function that is responsible for the calculation of molecular descriptors, after generating a dictionary containing the molecular descriptor, calculation results are returned to the view controller. ADME.py in return imports libraries from OpenBabel and Frowns which are responsible for many of the chemoinformatics computations and calculations. SRS.py is responsible for saving information to the database and for saving user results as well as creating sdf libraries. The search controller also imports from MODEL.py, Controller.py and SQLAlchemy in order to gain access to the FunGIMS database and core functionalities. Of importance to the search controller is the ability to import libraries from OpenBabel, as many functions rely on the capabilities provided by OpenBabel.

7. Methodology

7.1 User Capabilities

7.1.1 Main View

Once a user requests to view a molecule he/she is directed to the main molecule view page. In the main molecule view page the displayed information can be divided into 3 sections namely General Information, 3D-Structure and Molecular Descriptors or Molecular Properties (Figure 2.14).

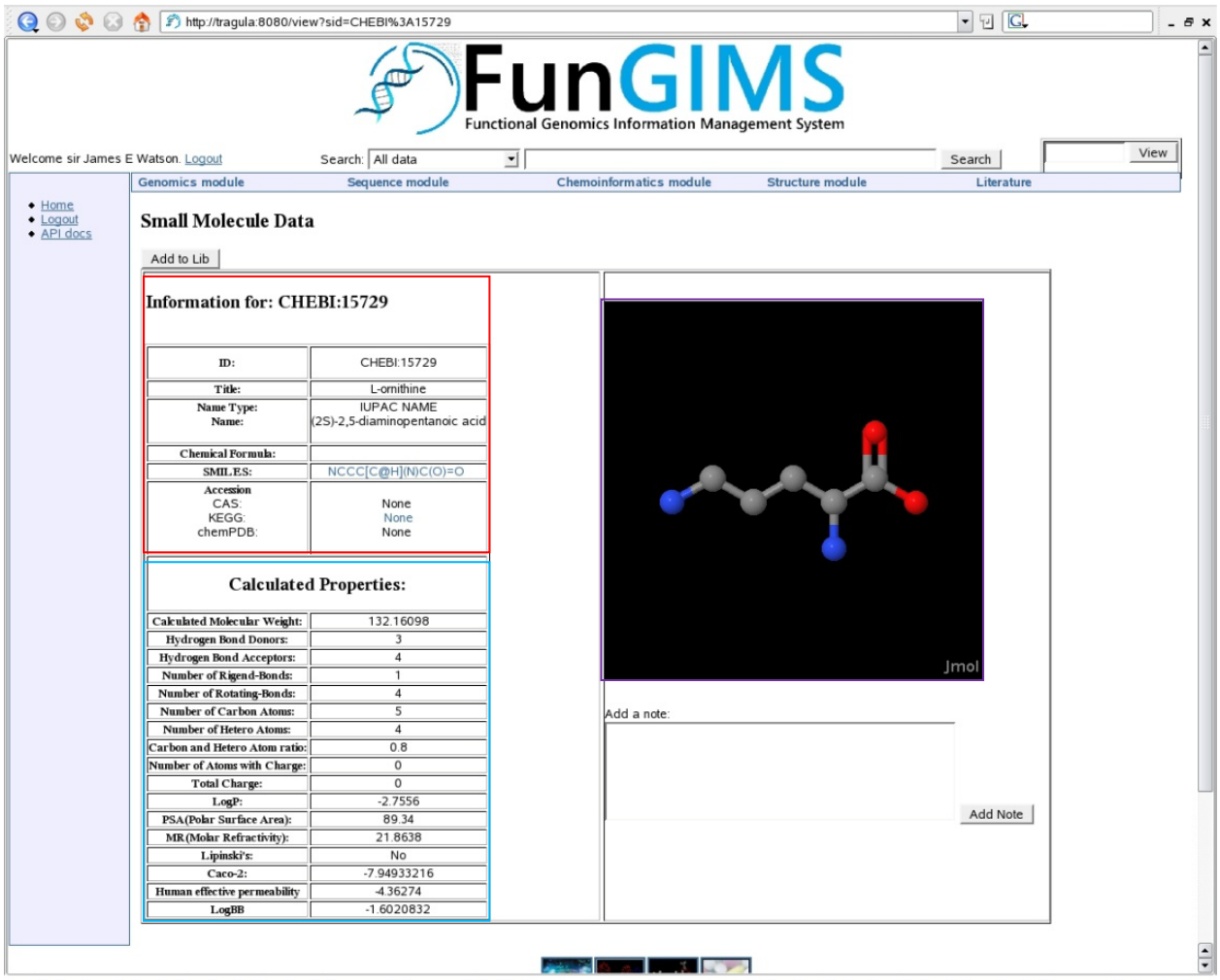


Figure 2.14: Screen capture of Main View. On the main view a user can find general information (red box), calculated properties (blue box) and the 3D-structure (purple box)

General Information

Under general information the following can be found:

- ID: FunGIMS database identifier
- Title: reference name of molecule
- Molecule name: these include common names as well as IUPAC names.
- Chemical formula found in database.
- SMILES found in database.
- Accession, this provides links to the viewed molecule in other databases available
- Ontology
- Relationships: relationship link between a small molecule and protein.

3D-structure

The 3D-structure of the molecule is viewed by means of Jmol Applet and all functionalities of Jmol are available.

Calculated Properties

The properties in the table below are calculated on the fly and are not found within the FunGIMS database. The reason behind this is that storing this amount of information would dramatically increase the amount of data within the database and it was found that on the fly calculation was not computationally nor time expensive. The different properties can be seen in Table 2.1.

Table 2.1 : Molecular Properties and descriptors found on main view page.

Molecular Descriptor or Property	Description	Additional Information
Calculated Molecular Weight:	The mass of one molecule of that substance, relative to the unified atomic mass unit u (equal to 1/12 the mass of one atom of carbon-12).	Calculated with OpenBabel
Hydrogen Bond Donors:	Nitrogen or oxygen atoms with one or more hydrogen atoms	Calculated with help of Frowns and additional coding
Hydrogen Bond Acceptors:	Nitrogen or oxygen atoms	Calculated with help of Frowns and additional coding
Number of Rigid-Bonds:	Double and Triple bond or Single aromatics bonds	Calculated with OpenBabel
Number of Rotating-Bonds:	Single bonds	Calculated with OpenBabel
Number of Carbon Atoms:	Total number of C atoms	Calculated with help of Frowns and additional coding
Number of Hetero Atoms:	Total number of atoms that are not Carbon or Hydrogen	Calculated with help of Frowns and additional coding
Carbon and Hetero Atom ratio:	Ratio of the number of Carbon atoms to Hetero atoms	Python script
Number of Atoms with Charge:	Atoms that carry a formal charge	Calculated with help of Frowns and additional coding
Total Charge:	Total Charge of a molecule	Calculated with help of Frowns and additional coding
LogP:	Log of partition (P) or distribution coefficient (D)	OpenBabel
PSA(Polar Surface Area):	PSA of a particular molecule is defined as the area of its van der Waals surface that arises from oxygen and nitrogen atoms or hydrogen atoms attached to oxygen and nitrogen atoms	OpenBabel
MR(Molar Refractivity):	Molar refractivity is a representation of the real volume of a molecule where radiation is of infinite wavelength.	OpenBabel
Caco-2	Caco-2 permeability thus refers to the passage of drugs through a Caco-2 monolayer.	Equation described in chapter 1
HEP	The rate and extent of intestinal drug absorption can be described by means of the effective intestinal permeability (P_{eff}).	Equation described in chapter 1
LogBB	LogBB can be defined as the ratio of the steady-state concentrations of a drug molecule between the brain and blood	Equation described in chapter 1

7.1.1.1 Inner Workings

The following section describes the actions taken when the main molecule view is displayed.

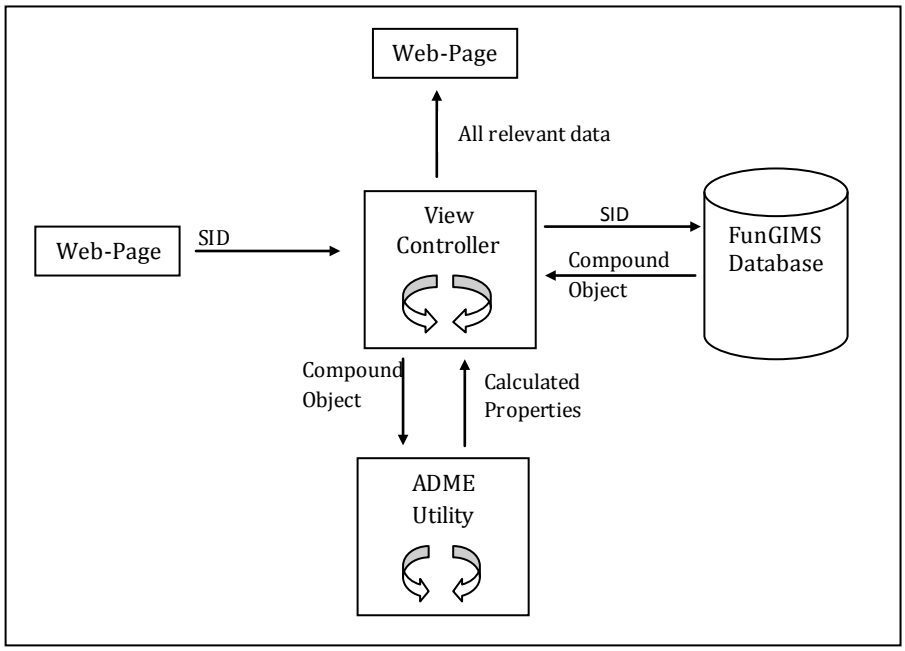


Figure 2.15:Is a schematic explanation of the inner workings of the main view. The view controller searches the FunGIMS database within the Compound table for the specific molecule by means of a sid. Once found, a Compound object is created and this contains the majority of information found in the general information section. The 3D information is collected from the database and written to a mol or sdf file in a temporary location, this is done due to the fact that Jmol requires a file located in a directory. Importing the function ADME from ADMED.py is responsible for supplying the calculated properties to the view-sub-controller. Once all information has been gathered and parsed, it is passed to the Kid template shown in Figure 2.13 and displayed to the user.

7.1.2 Keyword Search

The keyword search capability is not module specific and thus handled by the FunGIMS core. A particular keyword can be search across the entire FunGIMS database or be filtered with regards to a particular data type being either a small molecule data type or genomics data type to name only two.

7.1.3 Similarity Search

7.1.3.1 User Input

The main purpose of this functionality is to search the database for molecules that are structurally similar to the molecule provided by the user.

It is required that the user provides the SMILES structure of a molecule. This can be done by typing in a valid SMILES structure or by drawing the particular molecule within JME and generating a SMILES structure. The user is also required to provide a Tc (Tanimoto coefficient) cut off value, as seen in the screen shot shown in Figure 2.16.

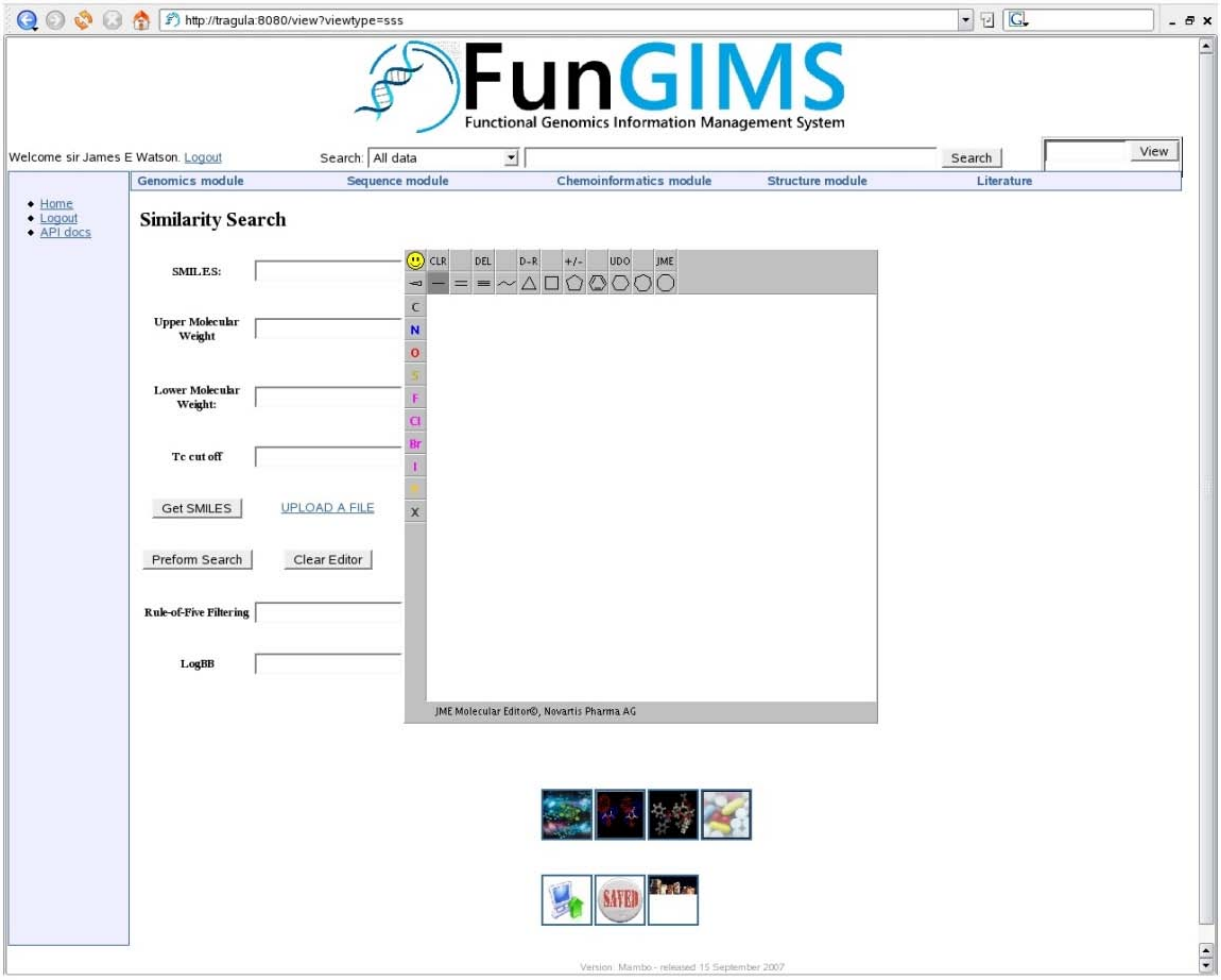


Figure 2.16 : Screen capture of Similarity Search Main Page. The main similarity page requires the user to provide information that will be used during the search.

7.1.3.2 Inner Workings

The following section describes the actions taken when a similarity search is performed.

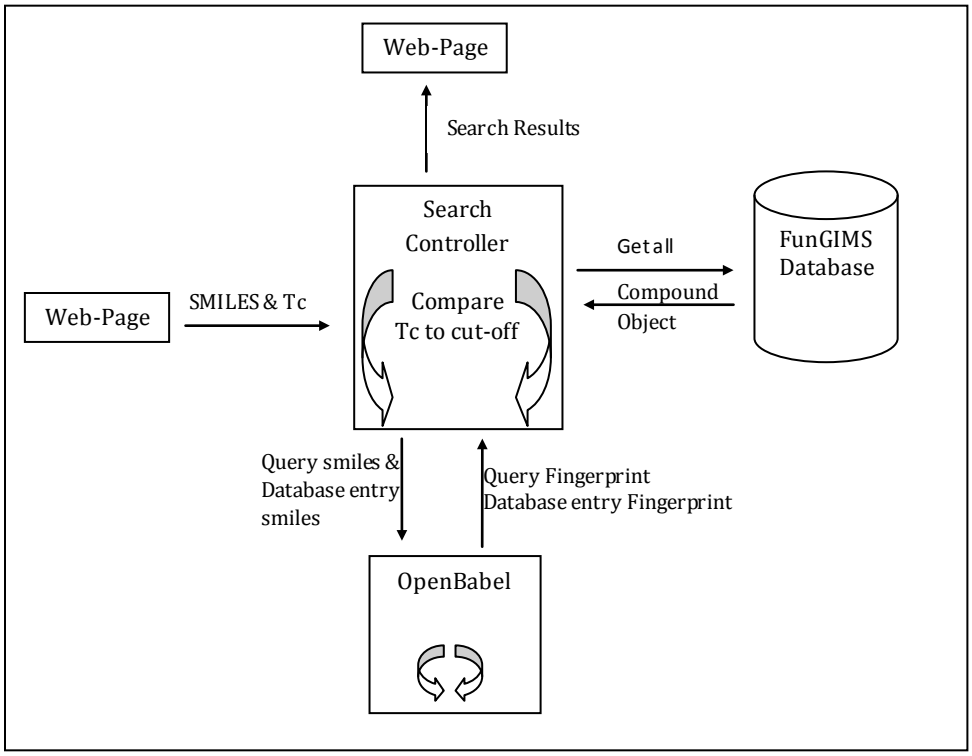


Figure 2.17: Schematic workings of similarity search.

The SMILES structure of the target molecule and parameters are passed to the appropriate function within the search controller. Here the SMILES structure of the target molecule is used to create an OpenBabel FP2 type fingerprint, after which the database is queried for all available SMILES structures. For each found SMILES structure an OpenBabel FP2 type fingerprint is calculated. The fingerprint of the user molecule and that of database molecules are used to calculate a Tc-value. When a molecule meets the required demand, it is stored in a list as a search results object. This in turned is passed to a Kid template and displayed to the user Figure 2.18.

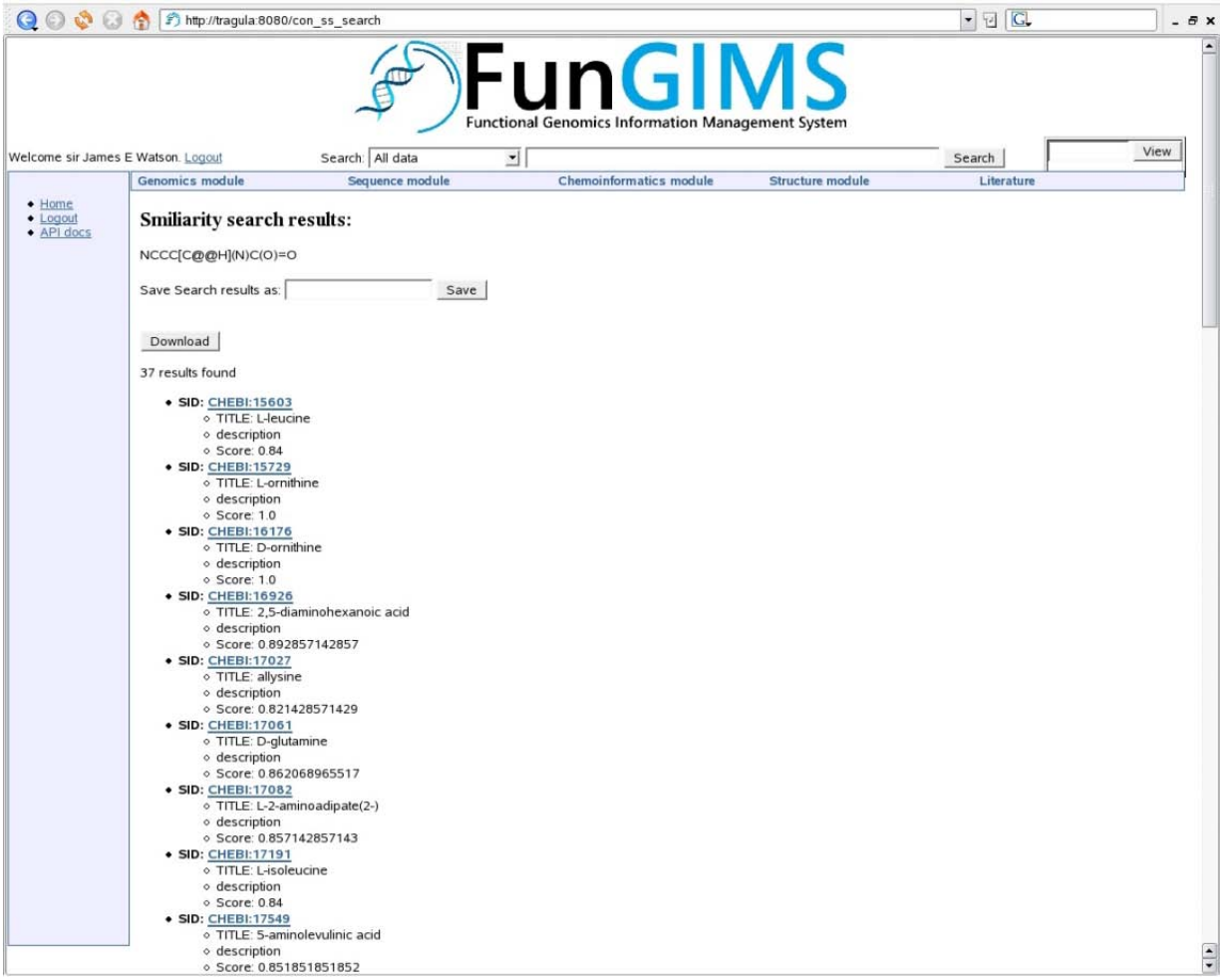


Figure2.18 : Screen shot of Similarity Search results of ornithine.

7.1.4 Substructure Searching

7.1.4.1 User Input

The main purpose of this functionality is to search the database for molecules that possess similar structural moieties.

The substructure search page also requires that the user provides the SMILES structure of a molecule. This can be done by typing in a valid SMILES structure or by drawing the particular molecule within JME and generating a SMILES structure (Figure 2.19).

Figure 2.19 : Screen capture of Substructure Search Main page. The main substructure page requires the user to provide information that will be used during the search.

7.1.4.2 Inner Workings

The following section describes the inner workings of the substructure search functionality.

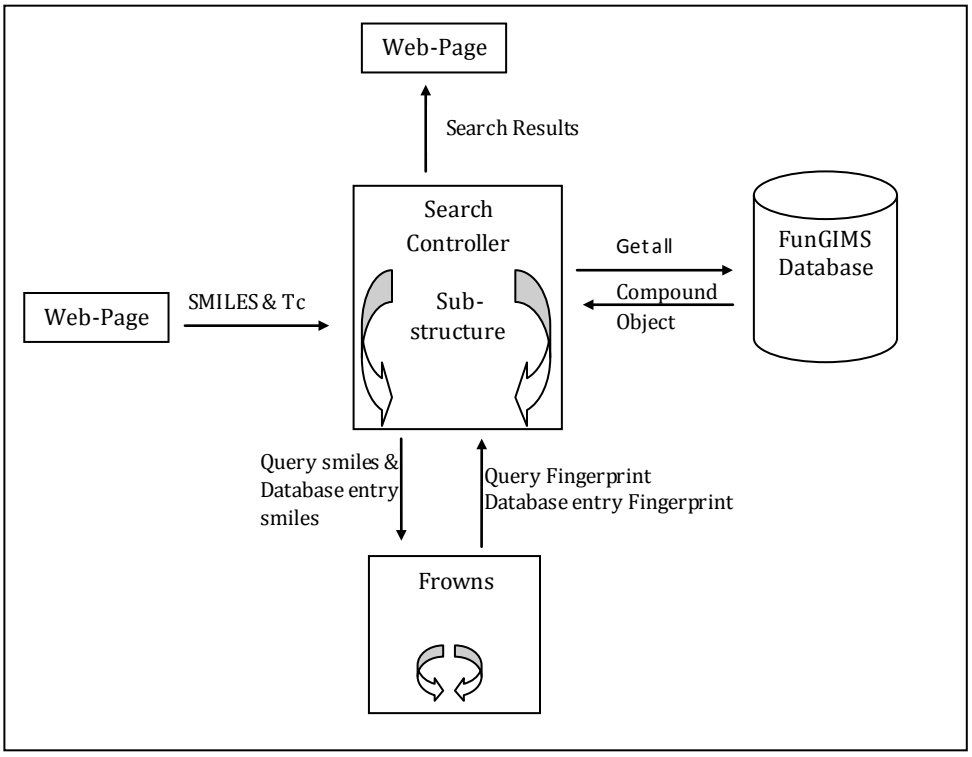


Figure2.20: Schematic workings of substructure search.

The SMILES structure of the target molecule and the parameters are passed to the appropriate function within the search controller. Here the SMILES structure of the target molecule is used to create a Frowns fingerprint. The database is then queried for all available SMILES structures. For each found SMILES structure a Frowns fingerprint and other molecular descriptors are calculated. These molecular descriptors are used in order to filter the results and the Frowns fingerprints are used to determine the presence of any substructures. When a molecule meets all the required demands it is stored in a list as a search result object and finally passed to a Kid template and displayed to the user.

7.1.5 Filtering of Libraries

7.1.4.1 User Input

A library of small molecules is only useful if it has an intended purpose and the quality of a library is determined by the molecules within. A library containing hundreds of thousands of molecules that do not possess the correct properties for the intended purpose is not highly useful. Filtering a library with regards to a series of properties can increase the use and value of a library for its intended purpose.

The chemoinformatics module of FunGIMS allows users to filter their libraries. A user selects the library to be filtered as well as the filtering criteria (explained in Table 2.2), once filtered a new library can be created that has all the functionalities as discussed in section 7.1.6.

Table 2.2 : Criteria of filtering applied by the chemoinformatics module of FunGIMS.

Property	**Assigned values**	**Option within FunGIMS**
Calculated Molecular Weight:	<=480g/mol	Rule-of-Five
Hydrogen Bond Donors:	<= 5	Rule-of-Five
Hydrogen Bond Acceptors:	<= 10	Rule-of-Five
LogP:	<= 5.6	Rule-of-Five
MR(Molar Refractivity):	<= 130	Rule-of-Five
Caco-2	>= -6	Permeability
HEP	>=-3.69	Permeability
LogBB	>= 0.3 (positive) <= -1.00(negative)	LogBB negative and LogBB positive

7.1.5.2 Inner Workings

Once a user has selected a library and criteria for filtering the library name and criteria are sent to the appropriate controller. The controller fetches the library from the FunGIMS database. In the controller the library is read one sid at a time and for each sid a molecule is retrieved from the databases as a compounds object. The compound object is passed to the ADME utility where the properties are calculated and returned to the controller. Back in the controller the properties are

compared and checked against the criteria set by the user and if a molecule passes it is added to results object list. After completion the results object list is passed to the webpage and displayed to the user where he can choose to save the new library or download the library. A schematic representation can be seen in Figure 2.21.

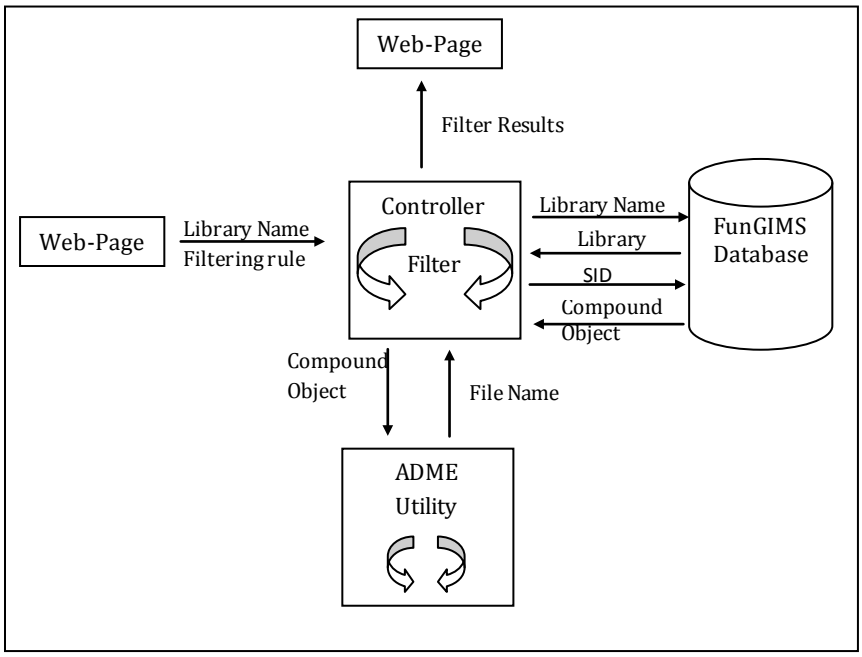


Figure2.21: Schematic workings filtering.

7.1.6 Library, Saving Search Results and User Uploads

7.1.6.1 Library

The results of any search can be saved as a new library within FunGIMS allowing a user's libraries to be located in a single place and be easily accessible. Users are also able to create their own personal library of molecules from the main molecule view page (Figure 2.14). Once a user decides to add a molecule to a library he/she is prompted to choose an existing library or create a new library. User libraries are stored within the FunGIMS database in the table ChemicalLibrary as a data blob. Within this data blob is a list of sids. When a user wishes to view molecules in a library, the data is collected from the FunGIMS database and displayed to the user.

7.1.6.2 Downloading Results

Users are able to download results obtained from any search performed as either a csv-file seen in Figure 2.22 or a SMILES-file. In the case of a csv file the library is retrieved from the database and for every molecule within the library the same properties that are displayed in the main view page are calculated. Finally the data is written to a file which can then be downloaded. If the user wishes to have all the molecules in a specific library in a single SMILES-file he only has to download the library as a SMILES-file from the search result page. In this case the library is retrieved from the database. For every molecule in the library the compound object is retrieved from the database. From here the SMILES-structure along with a chemical identifier is written to a file.

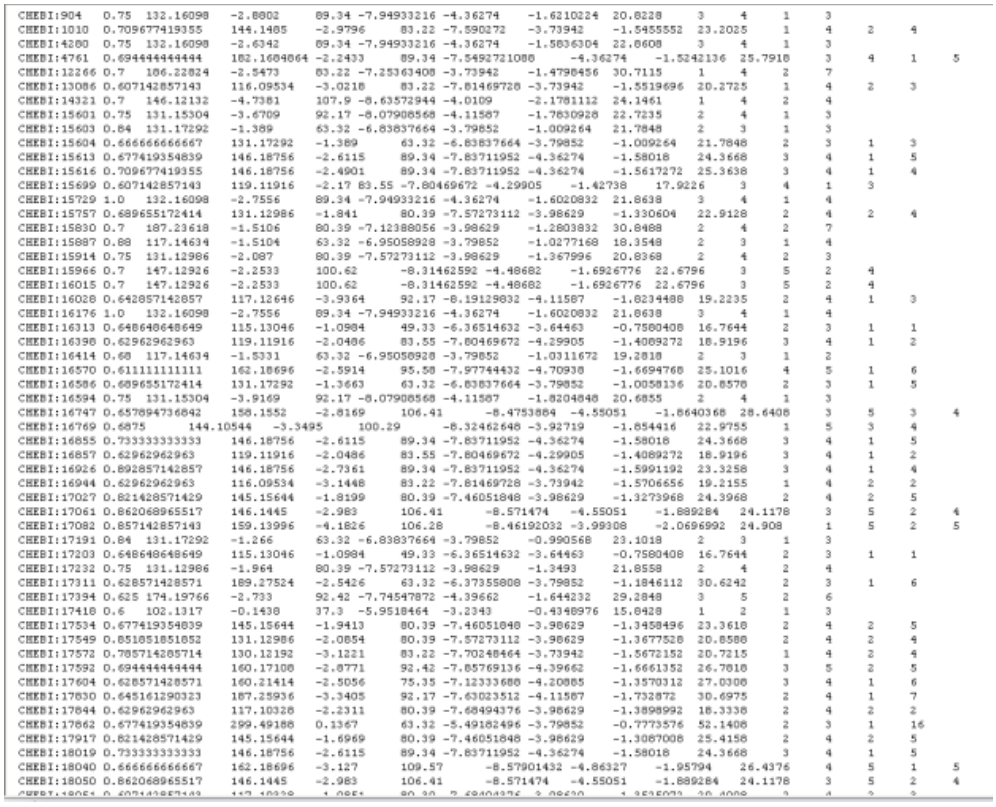


Figure 2.22: Example of search results that have been downloaded.

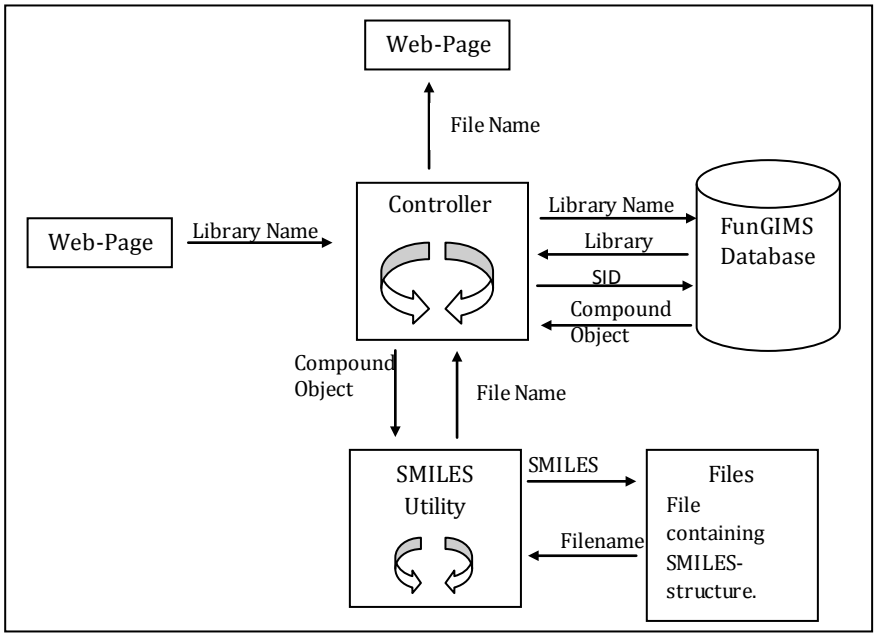


Figure 2.23 Process of downloading a library as SMILES-file.

7.1.6.3 User Uploads

If a user finds that a molecule of particular interest is not contained within the FunGIMS database he/she is able to upload the molecule into the database. This can be done in one of two ways. The first would be to upload a .mol, .mol2 or .sdf as shown in Figure 2.24, and these files are then parsed into the FunGIMS database. The second method of entry is where a user is required to input relative information again shown in Figure 2.24, these include name, formula, SMILES structure and a mol file created by JME. This information is parsed and the molecule is added to the FunGIMS database.

Figure 2.24: Screen Capture of Upload page. Users are able to upload molecules either by uploading a file (blue box) or by supplying the required parameters such as name formula and by means of JME the SMILES and mol are created.

7.1.7 User Information

Core functionalities prohibit users from viewing information that does not belong to the user, as described in section 3. Thus, a FunGIMS user can be assured that his/her data is private and secure.

8. Example Usage

8.1 ACE and Captopril

Users are required to be registered as a FunGIMS user before any major functionality and analyses are made available. Once a user logs in, they are able to perform a keyword search and specify the data type as a small molecule. In Figure 2.25 a keyword search example for captopril is illustrated.

Figure 2.25 : Keyword search for captopril

After the search has been preformed the search results are displayed as seen in Figure 2.25, each result is a link to a specific molecule and by clicking on a result the main view of that molecule is displayed.

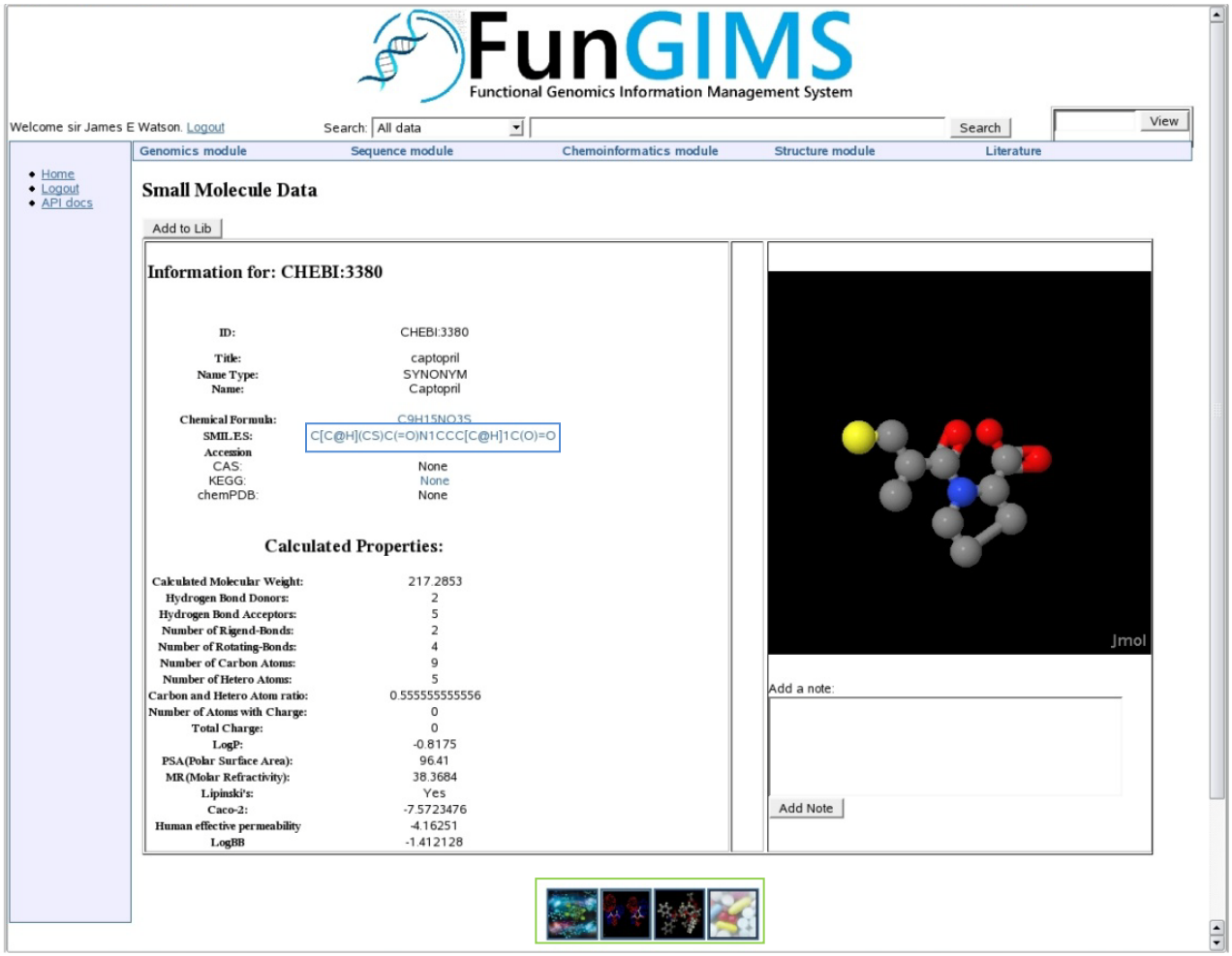


Figure2.26: Main view of captopril

Within the main view (refer to Figure 2.26) the user is presented with as much useful information as possible related to the selected molecule. A complete explanation is given in section 7.1.1.

Figure2.27: Similarity search of captopril

If the user finds all the information required within the main view page their interaction with FunGIMS is complete at this step. For argument's sake, if the user wants more data he can continue his FunGIMS session from this step. Should the user then wish to find molecules that are similar to captopril, the user first copies the SMILES-structure from the main view page as displayed in Figure 2.26 (blue box) and navigates to the similarity search page using the navigation pictures at the bottom of the main view page shown in Figure 2.26 (green box). Once the similarity page has been opened the user pastes the SMILES-structure in the appropriate input field and continues to fill in the rest of the fields as shown in Figure 2.27.

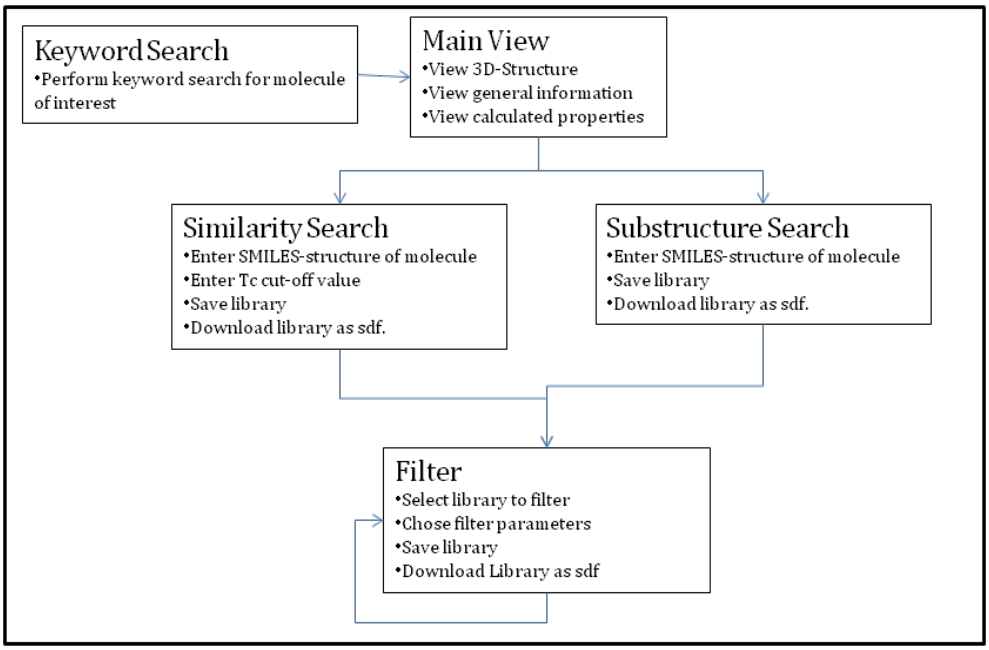


Figure 2.28: Workflow of the chemoinformatics module

On completion of the similarity search a results page is displayed containing all the molecules, along with their Tc-score, that met the criteria set by the user. As was the case with the previous results each result is a link to the main view of that particular molecule. The user is then able to create a new library containing these results as well as download the library as csv of sdf. If the user wishes to perform a substructure search the same applies as for similarity searches, expect that a substructure search requires less user input.

After creation of a library a typical tasks would be to filter the library. Accessing the filtering page displays a drop down menu from which a library can be chosen as well as check-boxes for the criteria. For captopril the similarity and substructure libraries where filtered with the same criteria, namely Lipinski's Rule-of-Five. Two new libraries where obtained containing 55 and 5 molecules. The filtered similarity library has 9 less molecules than before filtering and the substructure has 2 less. Figure 2.28 shows a typical workflow of the chemoinformatics module.

8.2 Comparison to other systems

The chemoinformatics module of FunGIMS will be compared to FAF-Drug (Miteva, Violas *et al.*, 2006) which is another proven web-based tool, in this section.

Table 2.3 shows the calculated properties of captopril obtained from FunGIMS and FAF-Drug respectively. As can be seen in Table 2.3 there is indeed no large difference in the calculated properties except that FunGIMS which includes predicted values for permeability and LogBB.

Table 2.3 : Calculated properties of Captopril from FunGIMS and FAF-Drug.

Property	**FunGIMS system**	**FAF-Drug**
Molecular weight	217.2853 g/mol	217.2 g/mol
Hydrogen Bond Donors	2	1
Hydrogen Bond Acceptors	5	4
Flexible Bonds	4	4
Rigid Bonds	2	7
Ring Number	N/A	1
Ring Length	N/A	5
Carbons	9	9
Non-Carbons	5	5
Ratio Non-Carbons/Carbons	0.55556	0.555556
Number of Charges	0	0
Total Charge	0	0
LogP	-0.8175	0.24
PSA	96.41	95.91
Caco-2	-7.57	N/A
HEP	-4.16	N/A
LogBB	-1.412	N/A

* The difference in the number of hydrogen-bond donors and acceptors as well as rigid bonds calculated by the different programs is due to the fact that each program has its own criteria for these descriptors.

The 2 pre-filtered libraries obtained for captopril were downloaded as sd-files from FunGIMS and uploaded to FAF-Drug in order to compare filtering. The comparison of the final filtered libraries obtained from FunGIMS and FAF-Drug can be seen in Table 2.4.

Table 2.4: Library size after filtering.

Library	FunGIMS-System	FAF-Drug
Similarity Library	55	22
Substructure Library	5	5

9. Discussion

The methodology illustrated in chapter 2 was successfully put into practice within the chemoinformatics module. Implementation of the different methodologies allows users to examine a small molecule with regards to different molecular properties and molecular descriptors along with the ability to view the molecule within Jmol. Furthermore, users are able to perform similarity searches and substructure searches, users guide the search by providing a template molecule in SMILES format along with other criteria. Search results can then be viewed online or downloaded as a file where the user can perform further analysis on the given data. The ability to create small user libraries provides a way in which small molecules of interest can be organized and viewed.

SMILES proved to be an effective way to work with small molecules *in silico.* If SMILES were not available and complex calculation could not be performed on SMILES, it would have been necessary to look towards 3D-structure, molecular and quantum mechanics. Implementation of molecular and quantum mechanics would have been an enormously challenging tasks and would have had a serious impact on development time and computational time required to perform searches and calculations.

OpenBabel and the accompanying Python libraries and Frowns are mainly responsible for calculations done within the chemoinformatics module. Due to the fact that OpenBabel has proved itself as a successful chemoinformatics tool, it was decided to reuse OpenBabel functionalities and capabilities. The same approach was used when the decision was made to use JME and Jmol as an important part of the chemoinformatics module.

The methods developed that perform the functionalities of the chemoinformatics system are believed to be sufficient as they are computationally inexpensive and provide results in what is believed to be an adequate time frame. The use of SQLAlchemy in order to map database tables to class objects did indeed decrease database query time, compared to the use of RPC-servers (remote procedure calls servers) done in a previous version of FunGIMS, and made handling of query results much easier and thus more efficient. The unseen processes handled by TurboGears in order to

maintain a working website, increased the ease and speed at which a working website with minimal functionalities could be started and allowed additional functionalities to be added without any major difficulties.

Adhering to fundamental software methodologies and techniques was taken into consideration in the design process, which resulted in a design capable of handling the scientific data that it was intended to and that will be easy to maintain and edit by future developers.

The chemoinformatics module of the FunGIMS system was used in a practical example, by investigating an angiotensin converting enzyme (ACE) inhibitor known as captopril. The FunGIMS system was able to find captopril by means of a keyword search within a few seconds. After the keyword word search had been preformed the main view of captopril was examined and a fast array of useful information as well as an interactive 3D-structure was observed. In order to create some libraries captopril was subjected to similarity and substructure searches. For the similarity search after entering the SMILES-structure and Tc cut-off value of 0.6 the system produced results within approximately 20min, this is believed to be within a reasonable time frame. The substructure search was found to take longer in performing its tasks, only producing results after approximately 2 hours. The delay in time was the result of a large amount of recursive work that had to be done and unlike C or C++, Python handles recursion differently and not as fast as these lower level languages. Never the less, the time to complete the search is still considered within reason. Both libraries where subjected to filtering applying only the Rule-of-Five. Filtering was completed within seconds producing two new libraries each showing a decrease in number of molecules.

The comparison between FAF-Drug and the chemoinformatics module of FunGIMS was done to validate that FunGIMS could reproduce results obtained by other web applications such as FAF-Drug.

Table 2.3 lists properties calculated by each system respectively. One would believe that all the properties should have identical values, in contrast to this, all the properties do not have equal values. In fact the agreement in PSA values are very good, logP is indeed very different. The chemoinformatics module uses OpenBabel to calculate logP and FAF-Drug makes use of Xtool. This is indeed an interesting point and it is believed that this should not be taken lightly, especially as the two logP values are significantly different. On the other hand Xtool has been proven to be useful in calculating logP values and can thus not be discarded. Taking OpenBabel into consideration also provides a useful logP value as this is close to -1.10 reported by Lin *et al*. To summarize both applications use different methods of calculating logP and both are considered useful. Further comparison shows that FAF-Drug calculates additional properties related to ring-structures, where these properties are not calculated by the chemoinformatics module of FunGIMS. Unlike FAF-Drug the chemoinformatics module provides three additional properties that is regarded as import and is believed to show increasing importance in research to come. These are methods of calculating Caco-2, Human effective absorption and logBB.

Comparison of each of the two applications' filtering procedures reveals that the filtering done by the chemoinformatics module of FunGIMS is indeed faster than that done by FAF-Drug. When the four libraries are compared as in Table 2.4 it is seen that FunGIMS produces a larger similarity library of filtered results that FAF-Drug does in contrast to the substructure library that produces the same results. The difference in the results produced is a product of how each application calculates the different properties and the criteria set for each property within the filtering procedure.

To conclude it is proposed that the chemoinformatics module of FunGIMS can indeed compete with other web-based chemoinformatics applications such as FAF-Drug. It is also apparent that the chemoinformatics system has areas on which improvements can be made as well additional features and analysis can be added to increase the use and appeal. The ability to store and organize libraries within an organized system is indeed a useful feature as libraries can easily become unorganized and be lost.

Chapter 3

Exploration of the chemical space of the polyamine pathway in *Plasmodium falciparum*

Introduction

There was a short period in history where it seemed that the war against malaria had been won. The mosquitoes were eliminated by larvacides and insecticides before they could carry the parasite between victims while chloroquine was used to kill parasites that did find their way into their human host. But alas, in the 1970's mosquito and parasite resistance emerged against insecticides and drugs being administered at the time. The war against malaria commenced.

This chapter focuses on malaria, to be more specific on the polyamine pathway in *Plasmodium* and the likelihood that the polyamine pathway is a potential target for chemotherapy. This is done through the exploration of the chemical space of the polyamine pathway in *Plasmodium* using the chemoinformatics module of FunGIMS. Analogue libraries were generated and a maximum common substructure (MCS) for each library obtained from FunGIMS was determined by means of clustering with the aid of Library MCS from ChemAxon. This provided insight in to the diversity of each library and the MCS gave an indication of the common structural moieties, and so doing provided a means of evaluating the chemical libraries. This was followed by a docking study on one selected enzyme from the polyamine pathway, spermidine synthase, and a specific library obtained from FunGIMS. This particular docking study was done to attempt to demonstrate that it would be possible to dock compounds found in a chemical library obtained from the FunGIMS system as result of a similarity search and filtering.

1. Malaria

More than 40% of the world population is at risk of contracting malaria as can be seen by the spread of malaria in Figure 3.1. In 2004 the number of malaria infections was between 350 and 500 million cases. The number of deaths due to malaria world wide is approximated at 1.1-1.3 million per annum, as stated in the World Health Organization (WHO) report of 1999-2004 (Yeh and Altman, 2006; Tuteja, 2007).

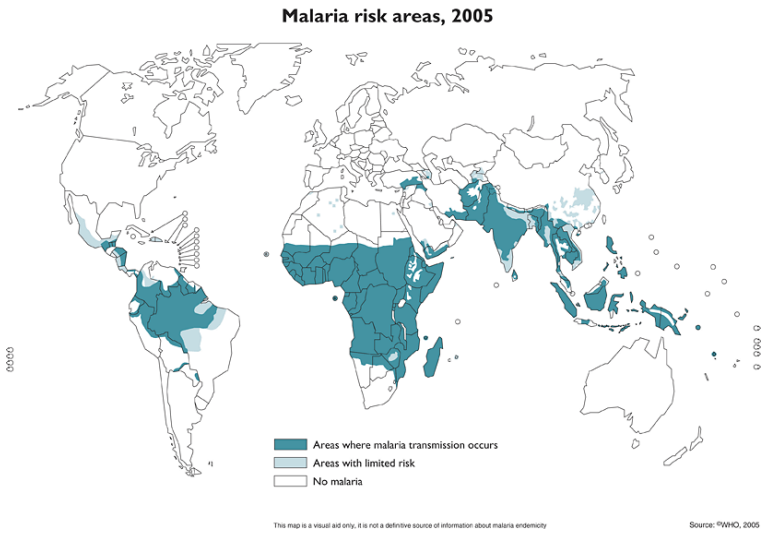


Figure 3.1 : Malaria spread across the world as reported by WHO in 2005 (http://www.nathnac.org/travel/factsheets/malaria.htm, 2007).

Malaria is found in subtropic and tropic regions where sub-Saharan Africa accounts for the majority of all malaria cases. Malaria is passed on through the bite of a contaminated female *Anopheles* mosquito (Figure 3.2). In the entire *Anopheles* genus about 60 are malaria vectors of which 30 are of major importance (Tuteja, 2007).

Figure 3.2: Close-up view of Anopheles mosquito (Simpson, 2001).

1.1 Parasite

Malaria parasites are eukaryotic single celled microorganisms belonging to the genus *Plasmodium.* Five species are believed to be infectious to humans and they are *Plasmodium falciparum, Plasmodium vivax, Plasmodium ovale, Plasmodium malariae* and *Plasmodium knowlesi.* Of these *P. falciparum* is the cause of severe potentially fatal cerebral malaria, and *P. vivax* is found to be the most wide spread malaria but infection by *P. vivax* is rarely fatal. Both *P. vivax* and *P. falciparum* can cause blood loss (anemia). In the case of *P.vivax* a mild anemia is more common, whereas *P. falciparum* results in a more severe anemia (Tuteja, 2007).

1.1.1 Life Cycle of *Plasmodium*

The malaria parasite life cycle involves two stages, the sexual and asexual stages. Sexual replication is completed in the Anopheles mosquito while asexual replication takes place in the human host. Figure 3.3 follows the life cycle of *Plasmodium* through each of the different stages.

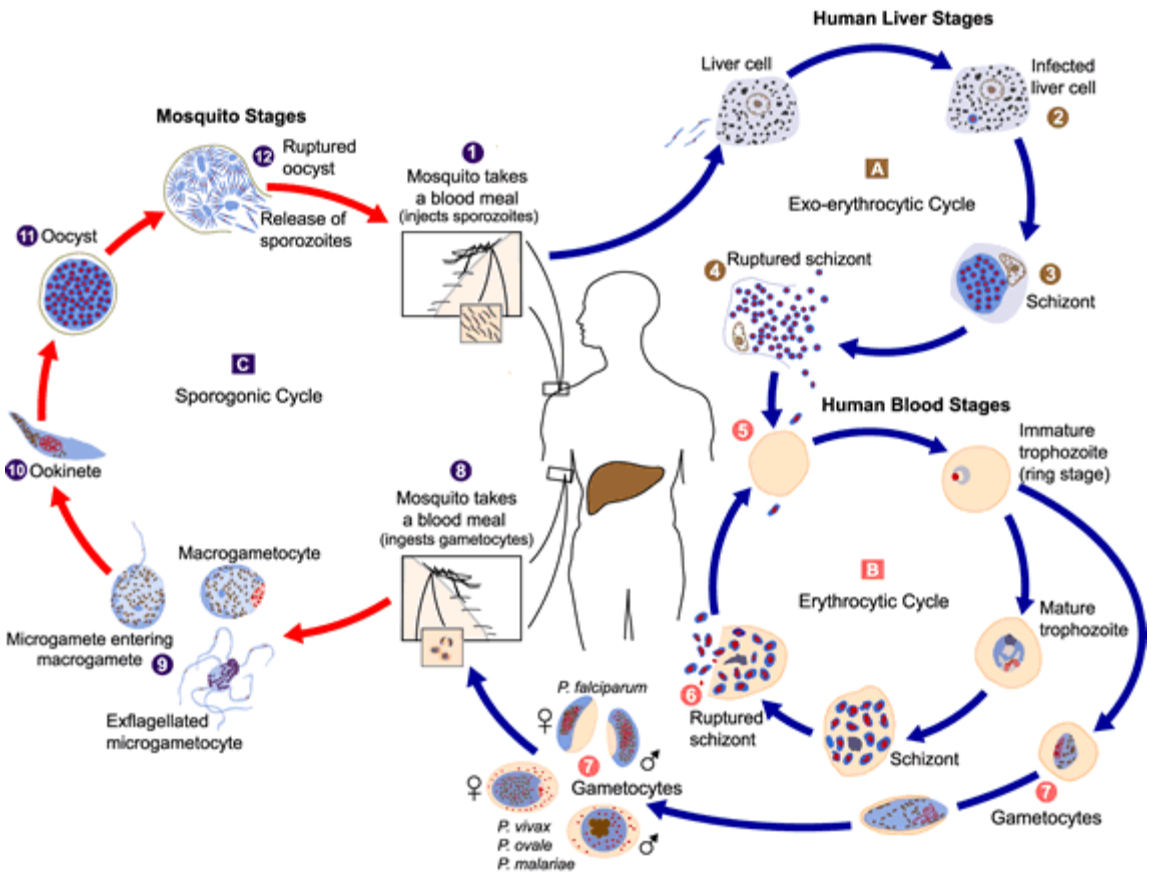


Figure 3.3 : Life cycle of *Plasmodium*(http://www.cdc.gov/malaria/biology/life_cycle.htm, 2006).

During a blood meal, a malaria-infected female *Anopheles* mosquito inoculates sporozoites into the human host. Sporozoites infect liver cells and mature into schizonts, which rupture and release merozoites. After this initial replication in the liver known as exo-erythrocytic schizogony, the parasites undergo asexual multiplication in the erythrocytes (erythrocytic schizogony) illustrated in Figure 3.4. Merozoites infect red blood cells. The ring stage trophozoites, so called due to their characteristic morphology, mature into schizonts, which rupture releasing merozoites.

Some parasites differentiate into sexual erythrocytic stages (gametocytes). Blood stage parasites are responsible for the clinical manifestations of the disease. The gametocytes, male (microgametocytes) and female (macrogametocytes), are ingested by an Anopheles mosquito during a blood meal. The parasites' multiplication in the mosquito is known as the sporogonic cycle. While in the mosquito's stomach, the microgametes penetrate the macrogametes generating zygotes. The zygotes in turn become motile and elongated (ookinetes) which invade the midgut wall of the mosquito where they develop into oocysts. The oocysts grow, rupture, and release sporozoites, which make their way to the mosquito's salivary glands. Inoculation of the sporozoites into a new human host perpetuates the malaria life cycle.

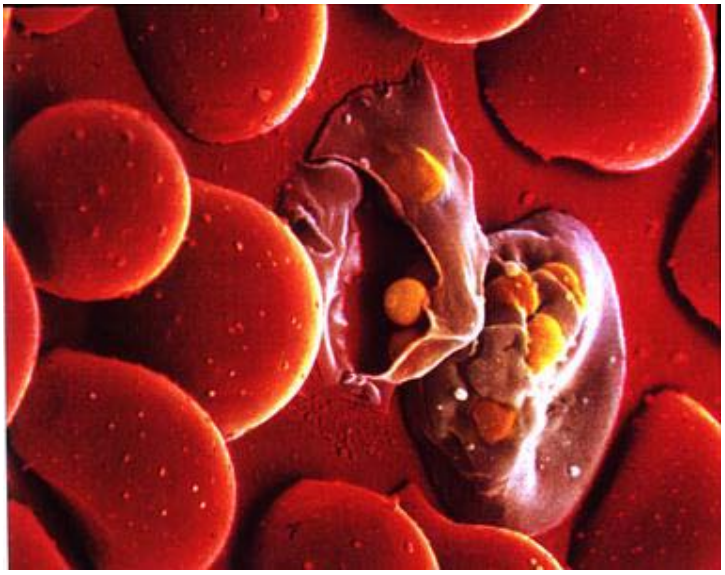


Figure 3.4 : Representation of infected red blood cells (Griffen-Smith, 2008).

2. Amines and Polyamines

The parent moiety of all amines is ammonia as shown in Figure 3.5. Amines are therefore called organic derivates of ammonia.

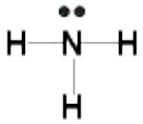


Figure 3.5 : Ammonia – parent moiety of amines.

Depending on the number of organic substituents attached to nitrogen, amines are classified as primary (RNH_2), secondary (R_2NH) or tertiary (R_3N) (Figure 3.6).

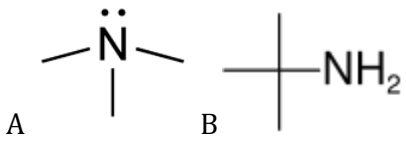


Figure 3.6 : Classification of amines depending on the number substituents.(A) A tertiary amine (B) A primary amine.

Amines are sp^3-hybridized at the nitrogen atom: three substituents for the three corners of a tetrahedron and the lone pair of electrons occupying the fourth corner. Figure 3.7 shows an electrostatic potential map of ammonia and shows that the negative region (blue) coincides with the lone-pair orbital on nitrogen(McMurry, 1999).

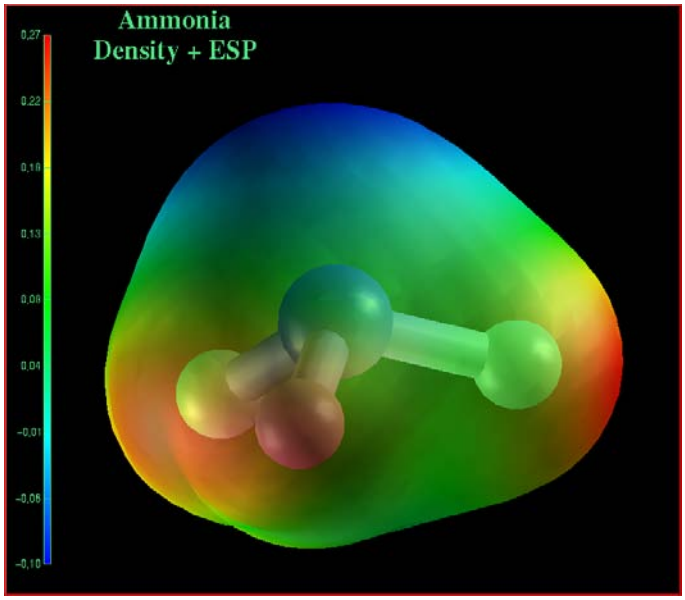


Figure 3.7 : Electrostatic potential map of ammonia showing a lone-pair of electrons on nitrogen (blue region) .

As a result of tetrahedral geometry, amines with three different substituents on nitrogen are chiral. Most chiral amines can't be resolved as the two enantiomers rapidly interconvert by pyramidal inversion.

Amines with less than five carbon atoms are generally water soluble, primary and secondary amines are able to form hydrogen bonds and are highly associated. Amines are also known for their

odour, such as trimethylamine which has a fishlike smell and cadaverine (1,5-pentanediamine) which has a self explanatory name.

The lone pair of electrons found on nitrogen dominates the chemistry of amines and due to this lone pair of electrons, amines are both basic and nucleophilic (McMurry, 1999).

Polyamines are organic compounds having two or more primary amino group such as putrescine, spermidine, and spermine and can be seen in Figure 3.8.

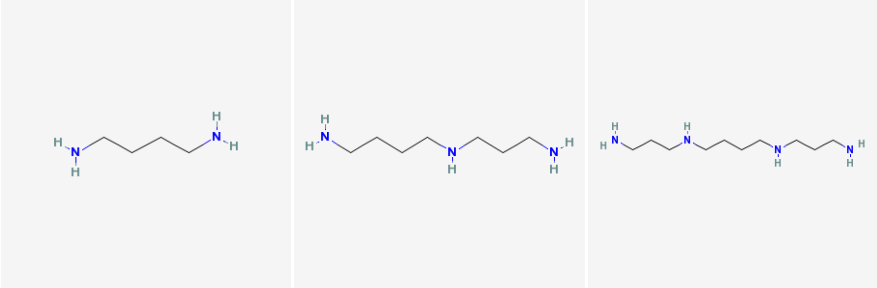


Figure 3.8 : Structures of common polyamines found in most mammalian cells. From left to right, putrescine, spermidine and spermine

3. Polyamine Biosynthesis

3.1 Mammalian Cells

Amino acid synthetic pathways, the urea cycle or cell transport supply precursors such as Met and Arg. Enzymes involved in polyamine synthesis are regulated at the transcriptional, translational and post-translational levels. A schematic representation of polyamine biosynthesis in a mammalian cell can be seen in Figure 3.9 (Muller, Coombs *et al.*, 2001).

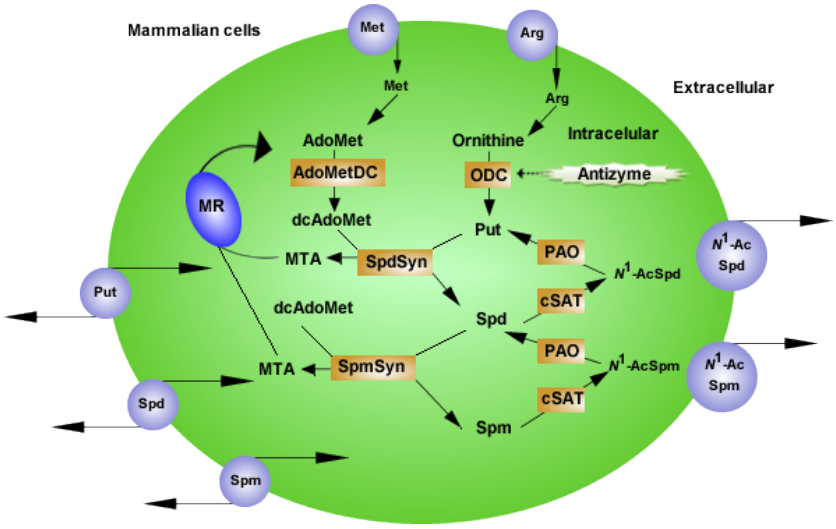


Figure 3.9 : Schematic representation of polyamine biosynthesis in a mammalian cell. Met and Arg, are supplied by amino acid synthesis, urea cycle and transport into cell. ODC and AdoMetDC have very short half-lives (15 to 35min respectively). Polyamines can be back converted by the 'interconversion pathway' involving cSAT and PAO. MTA, the by-product of synthesis is recycled to Met adapted from (Muller, Coombs *et al*., 2001).

3.2 Plasmodium

Polyamine biosynthesis in *Plasmodium* appears to be less complicated than in mammalian cells. In contrast to mammalian biosynthesis ornithine decarboxylase and S-adenosylmethionine decarboxylase occurs as a bifunctional protein. Also worth mentioning is that *Plasmodium* appears to lack an interconverting pathway as illustrated in Figure 3.10 (Muller, Coombs *et al*., 2001).

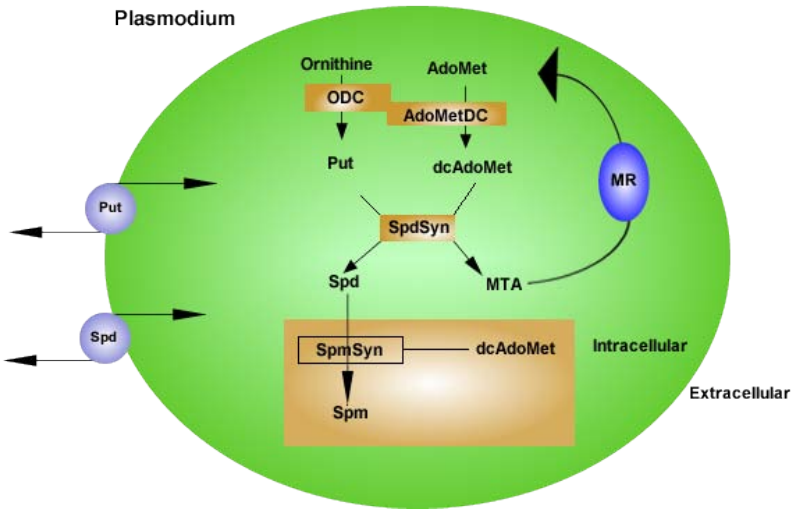


Figure 3.10 : Schematic representation of polyamine biosynthesis in *Plasmodium*. Put and dcAdoMet are supplied to SpdSyn by means of a bifunctional protein ODC-AdoMetDC. *Plasmodium* appears to lack an interconverting pathway, adapted from (Muller, Coombs *et al.*, 2001).

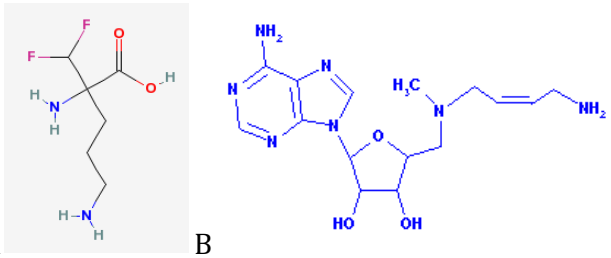
Ornithine decarboxylase (ODC) catalyses the first step in polyamine biosynthesis by the decarboxylation of ornithine to putrescine. S-Adenosylmethionine decarboxylase (AdoMetDC), which is a bifunctional enzyme along with ODC, produces decarboxylated S-adenosylmethionine (dcAdoMet), which supplies the aminopropyl group donor for spermidine and spermine synthesis. Spermidine synthesis is catalysed by spermidine synthase from putrescine and dcAdoMet supplied by ODC-AdoMetDC. Spermidine and dcAdoMet supplied by spermidine synthase and ODC-AdoMetDC respectively form the substrate of spermine synthase which produces spermine.

4. Polyamines as Likely Drug Targets against Malaria

Polyamines are omnipresent and play a fundamental role in cell growth and differentiation. The decarboxylation of ornithine to putrescine is the major limiting factor of polyamine biosynthesis. After decarboxylation, aminopropyl groups are subsequently attached to the terminal amino substituent of putrescine to form spermidine and spermine respectively (Muller, Coombs *et al.*, 2001; Yeh and Altman, 2006).

Anti-tumor and anti-parasitic effects have been observed when polyamine biosynthesis has been inhibited, as well as in cases where natural polyamine functions have been interfered with by means of polyamine analogues. Difluoromethylornithine (Figure 3.11), an inhibitor of ornithine decarboxylase (ODC, EC 4.1.1.17), blocks erythrocytic schizogony of *P. falciparum* in culture and

reduces parasitemia in *Plasmodium berghei*-infected mice. Inhibition of S-adenosylmethionine decarboxylase (AdoMetDC, EC 4.1.1.50) by MDL 73811 has a plasmodicidal effect *in vitro*. Rodent malaria was found to be curable by a combination of difluoromethylornithine and bis(benzyl)polyamine (Figure 3.11). This indicates that polyamine biosynthesis might be a potential target for a chemotherapeutic attack against *P. falciparum* in the blood stage form(Assaraf, Golenser *et al*., 1984; Byers, Casara *et al*., 1992; Sufrin, Meshnick *et al*., 1995; Muller, Da'dara *et al*., 2000; Muller, Coombs *et al*., 2001; Heby, Roberts *et al*., 2003; Yeh and Altman, 2006).



A B

Figure 3.11 : Molecules with known effect against malaria (A) difluoromethylornithine (DMFO) (B) MDL 73811

5. Docking

This refers to the physical fit between a target protein and small molecule, the small molecule being a pharmaceutical compound, substrate, inhibitor or activator (Figure 3.12). Docking algorithms find the optimal fit of a compound inside a defined cavity in a target protein, such as to minimize the total energy of the complex.

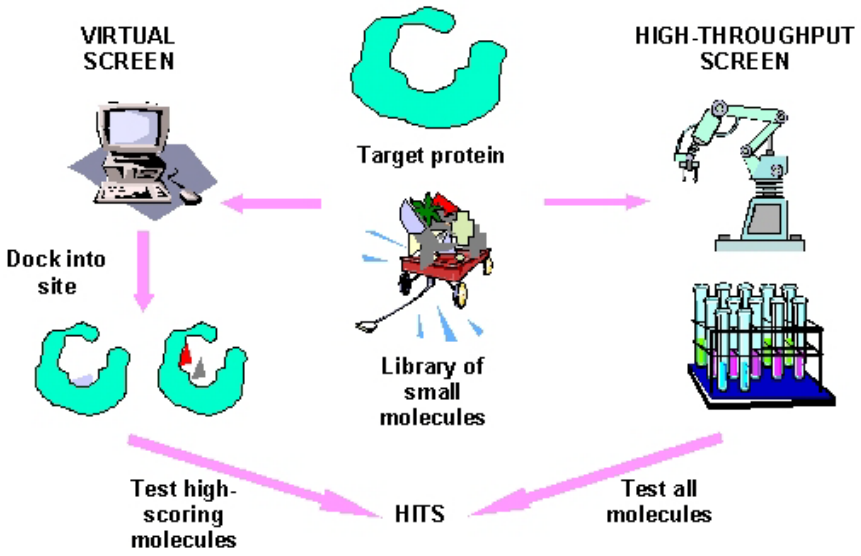


Figure 3.12 : Cartoon representation of molecular docking. Target protein and library of small molecules can take either one of two paths. The target protein and the entire library can be screened by high-throughput screening within an experimental laboratory. The other path involves *in silico* docking to find only the best molecules to continue with.

Docking techniques are mostly used in virtual screening of large databases of available chemicals in order to select likely drug candidates.

Molecular docking over a large virtual compound database may be thought of as virtual HTS technology. Molecular docking is used to predict the bound conformation of the ligand to the receptor. It connects structural bioinformatics and chemoinformatics. Molecular docking can be divided into a search algorithm and scoring function. A search algorithm should be efficient enough to find the lowest energy configuration or conformation. The scoring function should be able to distinguish a correct binding mode from other putative modes. Molecular docking provides not only lead compounds, but may also provide suggestions on how to modify the leads.

6. Clustering

Clustering describes the process of dividing objects into several different classes. Small molecules can be clustered based upon their structural similarities, where closely related molecules are grouped into different classes, Figure 3.13. The clustering of small molecules has a large variety of parameters of similarity and differences, based upon structural and non-structural features of the chemical structure (Barnard and Downs, 1992; Stahl and Mauser, 2005).

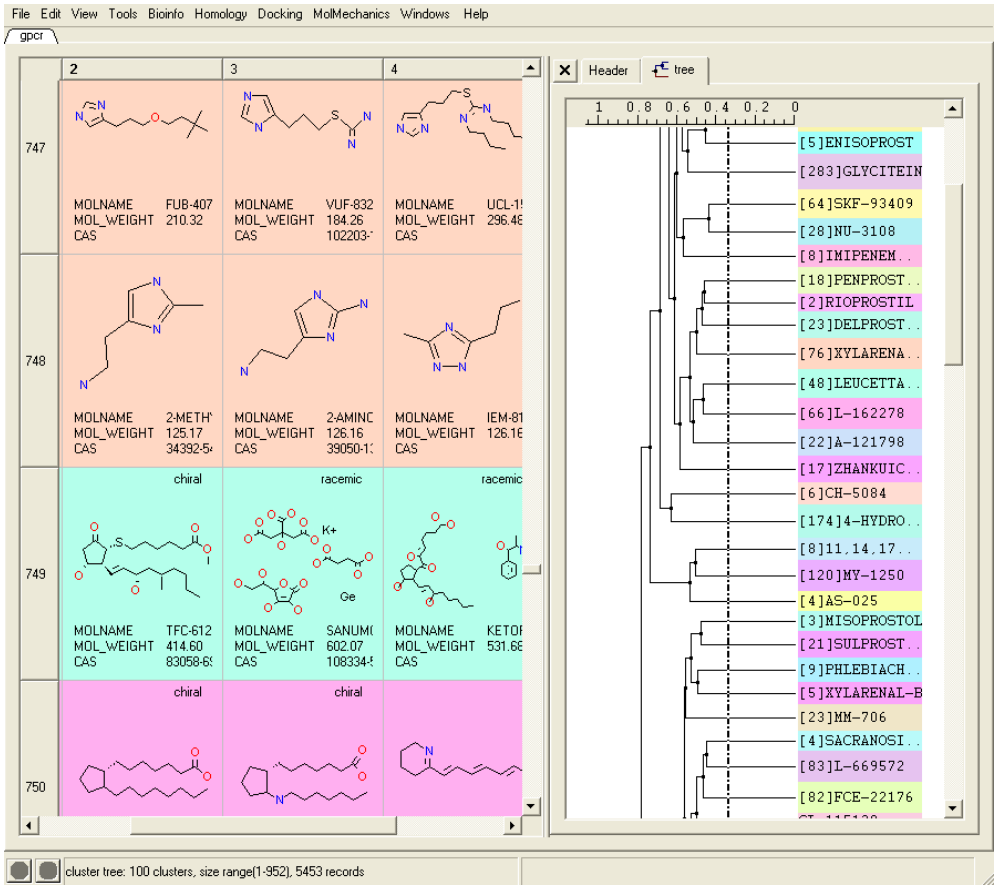


Figure 3.13 : Example of chemical structure clustering. Small molecules can be clustered based upon their structural similarities. Closely related molecules are grouped into different classes, pink, green and skin. These three would then form part of a larger class.

6.1 Hierarchical Clustering Methods.

Two basic approaches can be used to generate a hierarchical cluster namely, Agglomerative algorithms and Divisive algorithms. Agglomerative algorithms build up a cluster from the bottom up. This is done first by merging individual compounds into clusters, followed by the merging of clusters into super-clusters. This is continued until merging brings all compounds into a single cluster. Divisive algorithms operate in a top-down fashion, dividing the file into smaller subsets, by means of binary splitting.

All the agglomerative methods use in essence the same algorithm, in which a matrix of (dis)similarities is measured for each pair of compounds in use. The most similar pairs are combined in a cluster and the (dis)similarities matrix recalculated. A new cluster is regarded as a single unit. This is repeated until all clusters are merged together. The main difference between

these algorithms is the way in which the (dis)similarities matrix is recalculated. For example in single-linkage clustering the similarities between a pair of clusters is defined to be the closest similarity between any pair of compounds from each cluster and the two clusters that are closest are linked. With complete-link hierarchical clustering, at each step the two clusters that have the smallest distance between them are merged (Barnard and Downs, 1992; Stahl and Mauser, 2005).

6.2 Non-hierarchical Clustering Methods.

Nonhierarchical clustering implies the division of structures into a number of non-overlapping subsets, resulting in a partition where no hierarchical relationship exists. The simplest method for doing so is the single-pass or leader algorithm. This entails the comparison of each compound to the clusters formed so far and either adding the compound to the most similar cluster or it can be used to start a new cluster if there isn't any closely related cluster (Barnard and Downs, 1992; Stahl and Mauser, 2005).

6.3 Choice of Clustering Algorithms

The choice of a clustering algorithm depends on several factors, such as the computing resources required. Then there is the consideration that some algorithms are intrinsically better at identifying certain types of clusters. For example single-link hierarchical agglomerative algorithms are practically good at identifying long stringy clusters, while other algorithms tend to identify more compact globular clusters. The obvious question is "Which is better to use? " This depends on the nature of the dataset to be clustered and of greater importance, what the cluster would be used for (Barnard and Downs, 1992; Stahl and Mauser, 2005).

7. Spermidine Synthase

Spermidine synthases belong to the aminopropyltransferase class of proteins, which generally consist of a small N-terminal domain and a large catalytic C-terminal domain (Rossmann-like fold). The preponderance of the spermidine synthases are homodimers, conversely in thermophiles these proteins happen to be tetramers. Spermidine synthase found in *Plasmodium falciparum* (PfSpdSyn) is a dimer in solution with an approximate subunit molecular mass of 36.6 kDa (Burger, Birkholtz *et al*., 2007). The monomer of PfSpdSyn has an N-terminal domain that consists of a six-stranded β-

sheet where the C-terminal domain consist of a seven-stranded β -sheet flanked by nine α-helices (Figure 3.14) (Burger, Birkholtz *et al*., 2007)

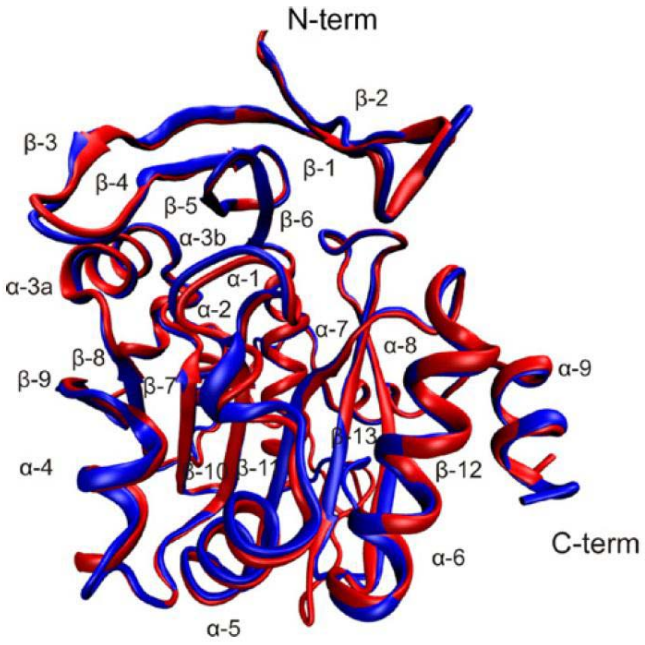


Figure 3.14 : A structural alignment of a PfSpdSyn homology model (red) and crystal structure (blue; PDB: entry 2HTE) This shows the overall structural of PfSpdSyn. Note the six β-sheet at the N-terminal (β-1 to β-6) and the C-terminal with its mixture of β-sheet and α-helices (Burger, Birkholtz *et al*., 2007).

Ikeguchi *et al*. proposed a mechanism of catalysis via a SN2 reaction. This resulted in an inverse configuration of the methylene carbon undergoing nucleophilic attack by putrescine, this is achieved via an attacking nitrogen, which is mediated by a gate-keeping loop (Ikeguchi, Bewley *et al*., 2006).

8. Goals

Malaria is a life-threatening disease which affects the lives of over three billion people. Factors like drug resistance and the absence of an effective vaccine are hindering the effective control of the malaria parasite. Thus, there is an urgent need for the identification and validation of new drug targets and chemotherapy agents.

This was attempted by exploration of the chemical space of the polyamine pathway in *Plasmodium* which has been shown to be a likely drug target. It was postulated that by performing similarity searches and substructure searches chemical libraries that contain structural analogues can be obtained. These libraries in turn can be filtered with regards to properties that have been shown to be important to acquire drug-like molecules. This would result in chemical libraries that contain structural analogues to the chemical space of the polyamine pathway. These libraries can be evaluated by means of clustering to determine their diversity as well as determination of MCS. As a proof of concept one enzyme with its related compounds was selected for further study. It was foreseen that it would be possible to successfully dock the compounds found in a chemical library to SpdSyn, finding one or more molecules that bind stronger than the substrate.

It was, thus envisioned to obtain good quality chemical libraries by similarity and substructure searches along with filtering, that could function as a starting point for new chemotherapy agents against SpdSyn in Plasmodium.

9. Materials and Method

9.1 Experimental Design

The chemical space of the polyamine pathway was explored with the aim of finding structural analogues that could be used as lead-like molecules, as the pathway has been identified as a potential target for chemotherapy. Initially the following polyamines were used as the basis of exploration: methylthioadenosine, ornithine, putrescine, spermidine and spermine along with DMFO and MLD 73811. Similarity and substructure searches were performed on each. This resulted in chemical libraries containing similar molecules. Similarity was based upon an OpenBabel fingerprint and scored by means of a Tc coefficient. A separate search involving the same query molecules provided libraries based on substructure matching. These libraries were then filtered with regards to molecular properties. The filtered libraries were subsequently clustered in order to determine a maximum common structure (MCS) for each library respectively. This, in the end, provided libraries containing possible lead-like compounds that could function as chemotherapeutic agents against malaria. As a proof of concept, spermidine synthase related compounds were then taken further. And the libraries obtained for spermidine was subjected to high-through-put docking into the crystal structure of spermidine synthase. Spermidine synthase was chosen due to the availability of a crystal structure and inhibition studies showing spermidine synthase to be a suitable drug target against malaria. The availability of known inhibitors also made it possible to compare results.

9.2 Keyword Search

Each of the chosen polyamines along with DMFO and MDL were entered as a keyword search into FunGIMS in order to determine their presence in the FunGIMS database.

9.3 Similarity Search

For each query molecule the SMILES-structure was obtained from the FunGIMS database, this in turn was used as the input structure. Each molecule underwent similarity searches with a Tc cut-off of 0.8. If a Tc of 0.8 resulted in too few compounds being obtained, the Tc cut-off was lowered to 0.6. Table 3.1 shows each query molecule and its respective similarity run. Each search performed was saved as a separate library.

Table 3.1 : Summary of similarity searches preformed on each of the five molecules

Molecule	Tc-value first run	Tc-value second run
methylthioadenosine	0.8	0.6
ornithine	0.8	0.6
Putrescine	0.8	0.6
spermidine	0.8	0.6
spermine	0.8	0.6
DMFO	0.8	0.6
MDL	0.8	0.6

9.4 Substructure Search

For each query molecule its SMILES-structure was obtained from the FunGIMS database, this in turn was used as the input structure. Each molecule underwent a single substructure search. Each search performed was saved as a separate library.

9.5 Additional Filtering

After completion of similarity and substructure searches, the obtained libraries underwent filtering using the FunGIMS chemoinformatics filtering tool. Each library underwent three different filtering runs, these runs differed in terms of the set criteria and Table 3.2 and 3.3 summarize the filtering criteria.

Table 3.2 Summary of filtering runs performed on each library produced by similarity search.

Molecule	**Rule-of-Five(ROF)**	**Permeability (Caco-2 and HEP) (Perm)**	**LogBB crosses BBB (LogBB+)**	**LogBB does not cross BBB (LogBB-)**	**Combinations**
methylthioadenosine	Yes	Yes	Yes	Yes	• ROF & LogBB-
putrescine	Yes	Yes	Yes	Yes	• ROF & LogBB- • ROF &Perm • Perm & LogBB-
ornithine	Yes	Yes	Yes	Yes	• ROF & LogBB
spermidine	Yes	Yes	Yes	Yes	• ROF & LogBB- • ROF &Perm • Perm & LogBB-
spermine	Yes	Yes	Yes	Yes	• ROF & LogBB- • ROF &Perm Perm & LogBB-
DMFO	Yes	Yes	Yes	Yes	• ROF & LogBB- • ROF &Perm
MDL	N/A	N/A	N/A	N/A	N/A

Table 3.3 : Summary of filtering runs preformed on each library produced by substructure search.

Molecule	**Rule-of-Five(ROF)**	**Permeability (Caco-2 and HEP) (Perm)**	**LogBB crosses BBB (LogBB+)**	**LogBB does not cross BBB (LogBB-)**	**Combinations**
methylthioadenosine	Yes	Yes	Yes	Yes	• ROF & LogBB-
putrescine	Yes	Yes	Yes	Yes	• ROF & LogBB- • ROF &Perm • Perm & LogBB-
ornithine	Yes	Yes	Yes	Yes	• ROF & LogBB
spermidine	Yes	Yes	Yes	Yes	• ROF & LogBB- • ROF &Perm • Perm & LogBB-
spermine	Yes	Yes	Yes	Yes	• ROF & LogBB- • ROF &Perm Perm & LogBB-
DMFO	Yes	Yes	Yes	Yes	• ROF & LogBB-
MDL	N/A	N/A	N/A	N/A	N/A

9.6 Clustering and MCS

JChem from ChemAxon was used to perform the clustering and determine the MCS. The filtered libraries were downloaded as SMILES-files and uploaded into LIBMCS. Due to the different filtering that was applied, a series of libraries were obtained, in order to remain consistent the only libraries that were clustered were the libraries obtained from ROF filtering alone. This is due to the fact that ROF takes into consideration bioavailability and oral activity leading to the decision that Caco-2 and HEP can be put aside during filtering. Within LIBMCS the MCS options were changed from the default values, the MCS mode was set to normal and minimal MCS size was changed depending on the query molecule.

9.7 Docking

The chemical library obtained from the similarity search of spermidine after filtering was chosen as the library that would be docked. This library was selected, due to it being identical to equivalent spermine library in terms of the number of hits obtained and on closer examination they contain the same compounds. The equivalent putrescine library had too few compounds and was thus, not used in the docking study. The libraries for spermidine obtained from substructure searches where not chosen as the clustering results showed that the library obtained from similarity is a good representative of the data. In general lead-like compounds are searched for in libraries that are similar to the substrate of the target protein, by choosing the particular spermidine library a different approach is taken where similarity to the product of the target protein is used as starting point.

The molecules found in the filtered library of spermidine as well as the substrate decarboxylated S-adenosylmethionine and the inhibitors 4MCHA and AdoDATO were docked into the crystal structure of spermidine synthase (PDB:2pPT Figure 3.15) (Dufe, Qiu *et al*., 2007) and their binding energies calculated by docking were compared. The purpose of docking S-adenosylmethionine, 4MCHA and AdoDATO was to provide controls to which the molecules found in the library could be compared.

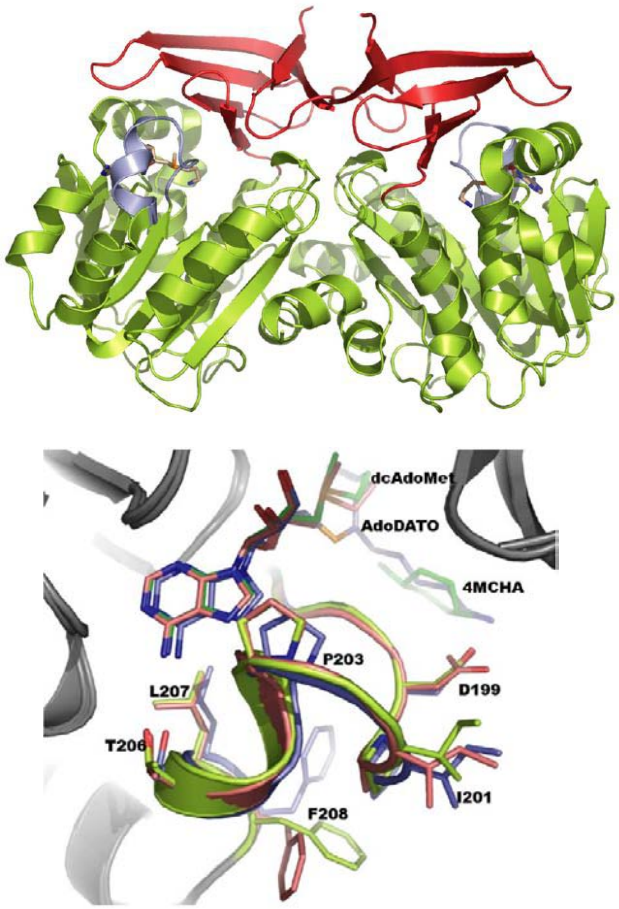


Figure 3.15 : The overall structure of pfSPDS. (Top) A ribbon representation of the three-dimensional structure of P. falciparum SPDS. The N-terminal domain is red and the C-terminal domain is green. The gatekeeper loop is shown in light blue. The bound dcAdoMet is shown in stick representation to mark the position of the active site of the enzyme as well as docking site (Bottom) A superposition of the structures of the complexes of pfSPDS with AdoDATO (blue), dcAdoMet orange) and dcAdo-Met–4MCHA (green), showing the region of the gatekeeper loop. The only notable change is found in the conformation of the side chain of Phe208 (Dufe, Qiu *et al*., 2007).

Polar hydrogen atoms were added to the protein and the ligands and the partial charges were assigned using python scripts from MGL-Tools. The energies were calculated using AutoGrid. For each ligand docked into spermidine synthase, AutoDock was run with the parameters shown in Figure 3.16. The remaining parameters used by the genetic algorithm were set to their default values. As an output AutoDock produced a docking file for each ligand. The first step in analyzing the results was to build a list sorted by energies of the lowest energy pose for each ligand. This was achieved by a python script provided by MGL-Tools which summarizes the results of each docking file. After summarization, the summary file was sorted based on lowest energy conformation in the largest cluster.

```
outlev 1                          # diagnostic output level
intelec                           # calculate internal electrostatics
seed pid time                     # seeds for random generator
ligand_types A C HD N             # atoms types in ligand
fld salam_A.maps.fld              # grid_data_file
map salam_A.A.map                 # atom-specific affinity map
map salam_A.C.map                 # atom-specific affinity map
map salam_A.HD.map                # atom-specific affinity map
map salam_A.N.map                 # atom-specific affinity map
elecmap salam_A.e.map             # electrostatics map
desolvmap salam_A.d.map           # desolvation map
move molecule00146.pdbqt          # small molecule
about 5.7317 -0.0118 0.0          # small molecule center
tran0 random                      # initial coordinates/A or random
quat0 random                      # initial quaternion
ndihe 6                           # number of active torsions
dihe0 random                      # initial dihedrals (relative) or random
tstep 2.0                         # translation step/A
qstep 50.0                        # quaternion step/deg
dstep 50.0                        # torsion step/deg
torsdof 6 0.274000                # torsional degrees of freedom and coefficient
rmstol 2.0                        # cluster_tolerance/A
extnrg 1000.0                     # external grid energy
e0max 0.0 10000                   # max initial energy; max number of retries
ga_pop_size 150                   # number of individuals in population
ga_num_evals 2500000              # maximum number of energy evaluations
ga_num_generations 27000          # maximum number of generations
ga_elitism 1                      # number of top individuals to survive to next generation
ga_mutation_rate 0.02             # rate of gene mutation
ga_crossover_rate 0.8             # rate of crossover
ga_window_size 10                 #
ga_cauchy_alpha 0.0               # Alpha parameter of Cauchy distribution
ga_cauchy_beta 1.0                # Beta parameter Cauchy distribution
set_ga                            # set the above parameters for GA or LGA
sw_max_its 300                    # iterations of Solis & Wets local search
sw_max_succ 4                     # consecutive successes before changing rho
sw_max_fail 4                     # consecutive failures before changing rho
sw_rho 1.0                        # size of local search space to sample
sw_lb_rho 0.01                    # lower bound on rho
ls_search_freq 0.06               # probability of performing local search on individual
set_sw1                           # set the above Solis & Wets parameters
compute_unbound_extended          # compute extended ligand energy
ga_run 10                         # do this many hybrid GA-LS runs
analysis                          # perform a ranked cluster analysis
```

Figure 3.16 : Autodock dpf , showing docking parameters that were used.

10. Results

10.1 Keyword Search

Methylthioadenosine, ornithine, putrescine, spermidine, spermine and DMFO were all found within the FunGIMS database. MDL on the other hand could not be found in the database by means of a keyword search, which meant that MDL had to be drawn using JME when performing the different searches.

The results obtained from a keyword search for each query molecule might not seem significant, but this means that the FunGIMS database may contain enough polyamines to function as a useful data set.

10.2 Similarity Search

After completion of a similarity search with a Tc value equal to 0.8 it was decided to lower the Tc value to 0.6 due to libraries being too small to be of any significant use for further filtering. The results obtained for similarity searches using methylthioadenosine, ornithine, putrescine, spermidine, spermine, DMFO and MDL as query molecules are summarized in Table 3.4, the entire dataset can be found at http://www.bi.up.ac.za/~jurgens/.

Table 3.4 : Summary of similarity search results.

Molecule	Run	Tc-value	Number of Hits
methylthioadenosine	1	0.6	219
ornithine	1	0.6	211
putrescine	1	0.6	82
spermidine	1	0.6	395
spermine	1	0.6	395
DMFO	1	0.6	40
MDL	1	0.6	0

The presence of the template molecule within the search results serves the purpose of acting as a positive control for each query molecule. The presence of the template molecule along with a Tc score equal to 1, illustrates the correct workings of the FunGIMS similarity search. A closer examination of the similarity runs of ornithine and methylthioadenosine reveals the presence of DMFO. By performing a relatively simple similarity run structural analogues, that may possess inhibitory effects, were found. This is shown by the presence of DMFO, which is known to have an inhibitory effect on polyamine biosynthesis, within certain similarity libraries.

Noteworthy is the difference in number of similarity hits between ornithine, methylthioadenosine, spermidine and spermine, compared to putrescine and DMFO and MDL which produce far fewer hits. This may be explained by simple visual comparison where ornithine, methylthioadenosine, spermidine and spermine, form longer carbon chains or are larger molecules than putrescine.

10.3 Substructure Search

The results obtained for substructure searches using methylthioadenosine, ornithine, putrescine, spermidine, spermine, DMFO and MDL as query molecules are summarized in Table 3.5 and the entire dataset can be can be found at http://www.bi.up.ac.za/~jurgens/.

Table 3.5 : Summary of substructure search results.

Molecule	**Number of Hits**
methylthioadenosine	16
ornithine	164
putrescine	749
spermidine	311
spermine	311
DMFO	4
MDL	0

In contrast to the similarity searches only a single substructure search was performed, unlike similarity searches, no parameters are set. It is not possible to say that in general substructure searches performed better than similarity, as the results differ significantly between molecules. This can be seen by comparing methylthioadenosine and putrescine where for methylthioadenosine substructure search produced far less results compared to its similarity search, in contrast to this the substructure search for putrescine, produced far more hits than its similarity search. This may be explained on the basis of the structural "complexity" of the different molecules where methylthioadenosine can be considered a far more complex structure than putrescine. It may seem that substructure searches did not produce useful results but examination of the substructure search results of ornithine and putrescine reveals the presence of DMFO, showing that a structural analogue that possesses inhibitory effects can be found using a simple substructure search.

10.4 Filtering

10.4.1 Similarity and Substructure

Each library was subjected to different filtering runs as described in section 5.5. The results for similarity and substructure can be seen in Table 3.6 and Table 3.7 respectively.

Table 3.6 : Filtering results for similarity libraries. Results obtained from different filtering criteria for libraries produced from similarity searches.

Molecule	**Rule-of-Five(ROF)**	**Permeability (Caco-2 and HEP) (Perm)**	**LogBB cross BBB (LogBB+)**	**LogBB Does not cross BBB (LogBB-)**	**Combinations**		
					ROF & LogBB-	**ROF &Perm**	**Perm & LogBB-**
methylthioadenosine	165	0	0	216	163	N/A	N/A
putrescine	81	68	0	4	3	68	1
ornithine	206	7	0	182	179	N/A	N/A
spermidine	392	358	4	25	24	257	2
spermine	392	358	4	25	24	257	2
DMFO	40	3	0	37	37	3	N/A
MDL	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 3.7 : Filtering results for substructure libraries Results obtained from different filtering criteria for libraries produced from substructure searches.

Molecule	Rule-of-Five(ROF)	Permeability (Caco-2 and HEP) (Perm)	LogBB cross BBB (LogBB+)	LogBB Does not cross BBB (LogBB-)	Combinations		
					ROF & LogBB-	ROF &Perm	Perm & LogBB-
methylthioadenosine	10	0	0	16	10	N/A	N/A
putrescine	610	275	1	294	202	260	5
ornithine	85	0	0	163	84	N/A	N/A
spermidine	145	86	0	207	50	83	5
spermine	145	86	0	207	50	83	5
DMFO	4	0	0	4	4	N/A	N/A
MDL	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Each of the fourteen libraries was filtered different times and each time the filtering criterion was changed. Out of all the filtering options four were constant for each molecule but in addition to these four a combination of two criteria were selected. The construction and the number of combinations made are dependent on the results of the constant four criteria and are not set in stone. It would have been interesting to see some sort of trend between the different filtering runs but due to the nature of the data this was not possible. What can be seen, is that filtering of libraries is indeed import and definitely a worthwhile application but care should be taken when filtering libraries. Although a large set of criteria seems appealing, this can result in a library being shrunk too much leaving libraries with very few entries, especially if the library is not large to start with.

10.5 Final Libraries

Completion of filtering resulted in two separate libraries for each query molecule. Table 3.8 is a summary of the final libraries and the complete data set can be found at http://www.bi.up.ac.za/~jurgens/.

Table 3.8 : Summary of final libraries that will be used in further investigations.

Molecule	Number of molecules in Final library as obtained from similarity	Number of molecules in Final library as obtained from substructure
Methylthioadenosine	165	10
Ornithine	81	610
Putrescine	206	85
Spermidine	392	145
Spermine	392	145
DMFO	40	4
MDL	N/A	N/A

10.6 Clustering and MCS

Clustering of each library respectively was done in order to evaluate each library with regards to it being representative of the polyamine pathway along with diversity.

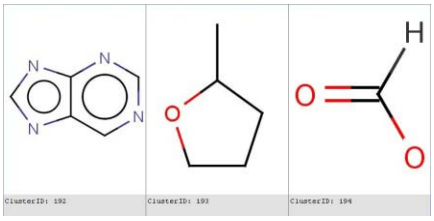
Clustering by means of LIBMCS was done successfully as a cluster as well as a MCS could be produced. The substructure libraries for methylthioadenosine and DMFO are the only ones to produce a single MCS, other libraries resulted in 2 or more. By decreasing the minimal size of the MCS the number of MCS could be decreased. This seems favorable but resulted in a very small molecules that would not have been of any use or importance. Thus the minimal MCS size was adjusted as to keep the results meaningful.

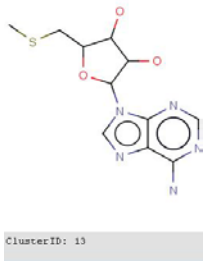
A summary of the clustering report for each library connected to a specific query molecule can be found in Table 3.9. The dendograms of each cluster run performed can be found at http://www.bi.up.ac.za/~jurgens/.

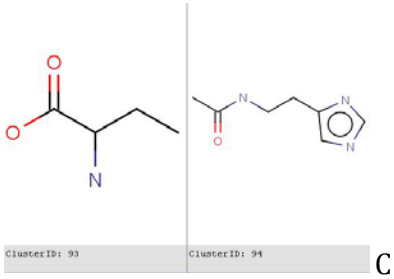
Table 3.9: Results produced from clustering in order to determine MCS.

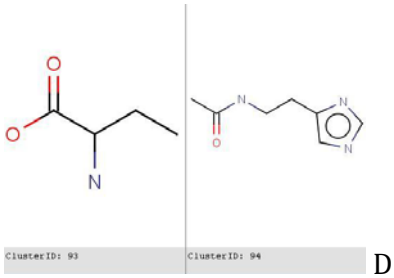
Molecule	Similarity Library		Substructure Library	
	Level Count	Cluster Count at top level	Level Count	Cluster Count at top level
Methylthioadenosine	6	3	3	1
Ornithine	3	2	3	2
Putrescine	5	4	4	5
Spermidine	5	3	4	5
Spermine	5	3	4	5
DMFO	5	2	2	1
MDL	N/A	N/A	N/A	N/A

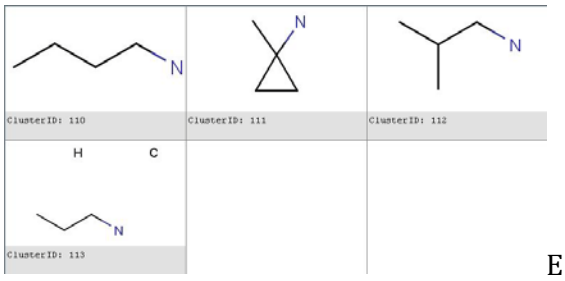
The molecules found in the top level of each cluster are considered to be the MCS for that particular library. The chemical structures for of the particular MCS are displayed in Figure 3.16.

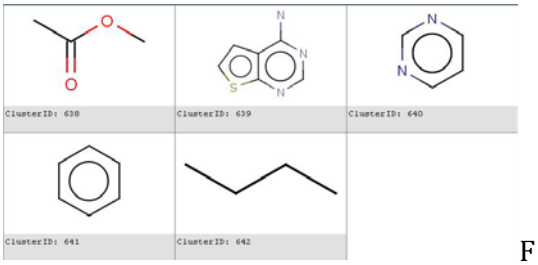

ClusterID: 192
ClusterID: 193
ClusterID: 194
A


ClusterID: 13
B

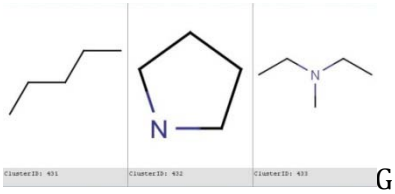

ClusterID: 93
ClusterID: 94
C


ClusterID: 93
ClusterID: 94
D

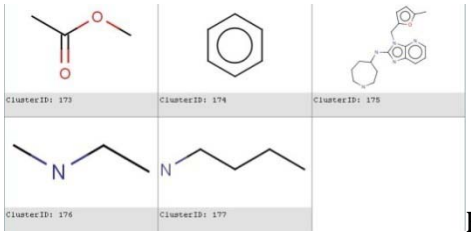

ClusterID: 110
ClusterID: 111
ClusterID: 112
ClusterID: 113
E



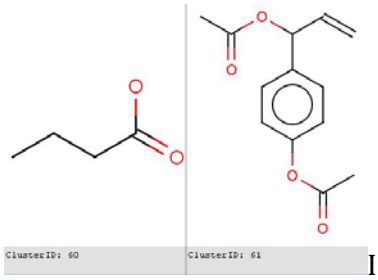
F



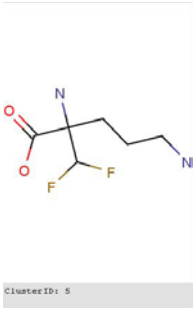
G



H



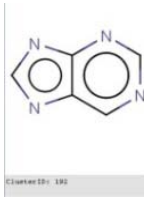
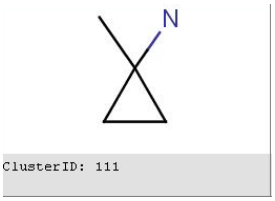
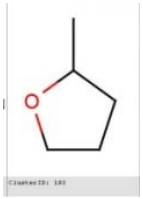
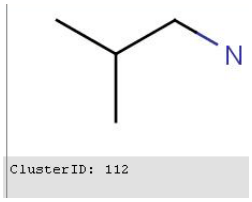
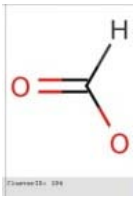
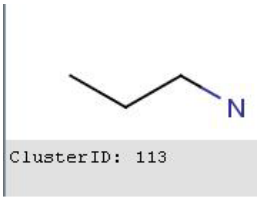
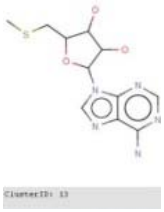
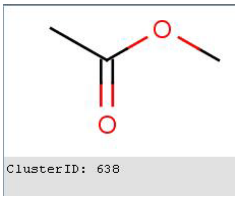
I

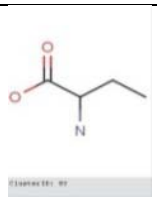
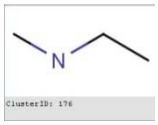
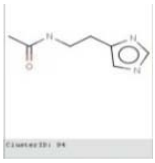
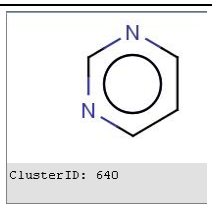
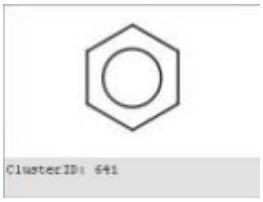
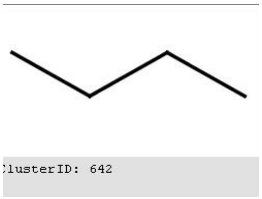
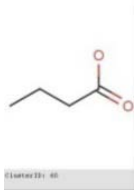
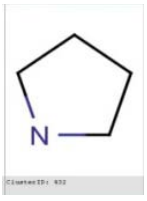
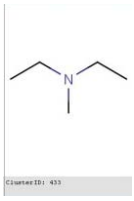
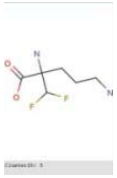
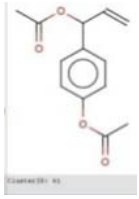


J

Figure 3.17 : (A) MCS results for similarity library of Methylthioadenosine. (B) MCS results for substructure library of Methylthioadenosine. (C) MCS results for similarity library of Ornithine. (D) MCS results for substructure library of Ornithine. . (E) MCS results for similarity library of Putrescine. (F) MCS results for substructure library of Putrescine. (G) MCS results for similarity library of Spermidine and Spermine. (H) MCS results for substructure library of Spermidine and Spermine. (I) MCS results for similarity library of DMFO. (J) MCS results for substructure library of DMFO.

Table 3.10: Representative moieties for all libraries.

	8,9-dihydro-7H-purine		1-methylcyclopropan-1-amine
	2-methyloxolane		2-methylpropan-1-amine
	formic acid		butan-1-amine
	2-amino-9-[5-(ethylsulfanylmethyl)-3,4-dihydroxyoxolan-2-yl]-3H-purine		methyl acetate

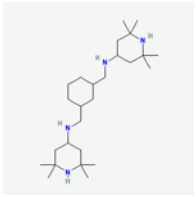
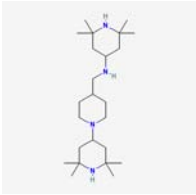
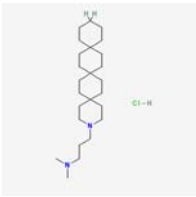
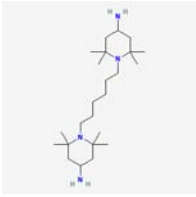
	2-aminobutanoic acid		N-methylethanamine
	N-[2-(3H-imidazol-4-yl)ethyl]acetamide		pyrimidine
	benzene		butane
	butanoic acid		pentane
	pyrrolidine		N-ethyl-N-methylethanamine
	2,5-diamino-2-(difluoromethyl)pentanoic acid (DMFO)		4-[(1S)-1-acetyloxyprop-2-enyl]phenyl] acetate

Clustering and calculation of a MCS for each library provides some insight into the overall representation of chemical moieties within in each library, pooling of all MCS also provides information regarding the chemical space of the polyamine pathway. The previous statement can be justified by Table 3.10 which shows different chemical moieties such as carbon chains, benzene rings, amines, purines, pyrimidines and carboxylic acid are present, all of which are present within the query molecules. Although it has been stated that a chemical library should have similar properties, it is also important that a certain degree of diversity is maintained. After clustering it was observed that all libraries had a degree of diversity, this is observed in the levels of clustering along with the number of MCS that was produced by each library.

10.7 Docking

After completion of docking the spermidine related compounds to spermidine synthase, the top 10 ligands were extracted from the sorted list and can be seen in Table 3.11.

Table 3.11: Top 10 AutoDock results for the library produced by spermidine.

Name	Structure	Binding Energy	kI
2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine		-14.44	26.15pM
2,2,6,6-tetramethyl-N-[[1-(2,2,6,6-tetramethylpiperidin-4-yl)piperidin-4-yl]methyl]piperidin-4-amine		-13.16	224.11pM
3-(15-azatrispiro[5.2.2.5^{12}.2^{9}.2^{6}]henicosan-15-yl)-N,N-dimethylpropan-1-amine hydrochloride		-12.68	510.89pM
1-[6-(4-amino-2,2,6,6-tetramethylpiperidin-1-yl)hexyl]-2,2,6,6-tetramethylpiperidin-4-amine		-12.35	881.65pM

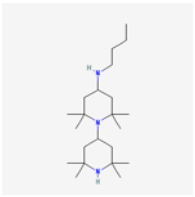
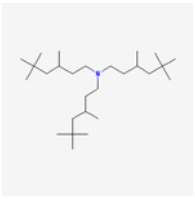
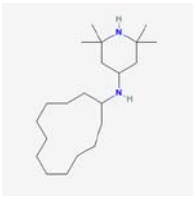
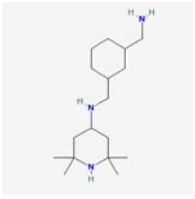
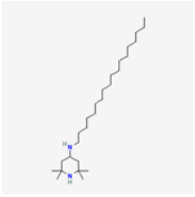
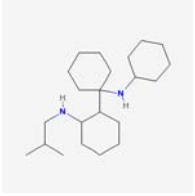
N-butyl-2,2,6,6-tetramethyl-1-(2,2,6,6-tetramethylpiperidin-4-yl)piperidin-4-amine		-11.7	2.65nM
3,5,5-trimethyl-N,N-bis(3,5,5-trimethylhexyl)hexan-1-amine		-11.54	3.5nM
N-cyclododecyl-2,2,6,6-tetramethylpiperidin-4-amine		-11.32	5.0nM
N-[[3-(aminomethyl)cyclohexyl]methyl]-2,2,6,6-tetramethylpiperidin-4-amine		-10.65	15.48nM
2,2,6,6-tetramethyl-N-octadecylpiperidin-4-amine		-11.3	5.21nM
N-cyclohexyl-1-[2-(2-methylpropylamino)cyclohexyl]cyclohexan-1-amine		-10.95	1.45nM

Table 3.11 reveals that the best docking results were obtained for 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine which has a binding energy of -14.44 kcal/mol, which is the sum of the intermolecular energy and the torsional free-energy penalty, and a Ki equal to 26.15pM ,which is the inhibition constant.
This is why 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine was taken as the best docked molecule.

Table 3.12 : Binding energies and kI values for substrate and known inhibitors.

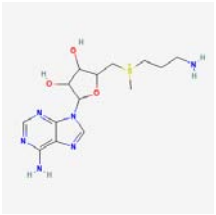
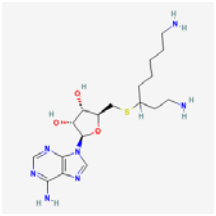
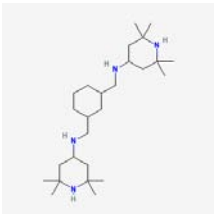
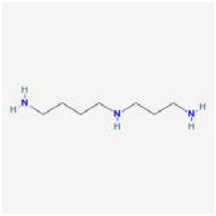
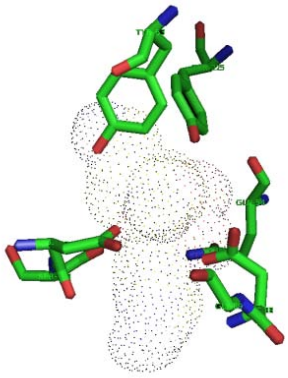
Name	Structure	Binding Energy	kI
Decarboxylated S-adenosylmethionine		-6.1	34.05 µM
S-adenosyl-1,8-diamino-3-thio-octane (AdoDATO)		-7.04	6.97µM
2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine		-14.44	26.15pM
Spermidine		-5.21	152.51 µM

Table 3.12 provides a means by which the binding energies of 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine can be compared to a substrate (decarboxylated S-adenosylmethionine) and a known inhibitor S-adenosyl-1,8-diamino-3-thio-octane (AdoDATO). Table 3.12 reveals that the binding energy of AdoDATO is larger than that of decarboxylated S-adenosylmethionine.
It would be favorable to have an inhibitor that has a binding energy greater than of the substrate, and this is observed in the binding energies of AdoDATO and decarboxylated S-adenosylmethionine. Although the difference in binding energies between AdoDATO and decarboxylated S-adenosylmethionine is not large AdoDATO has been shown to be an effective inhibitor.

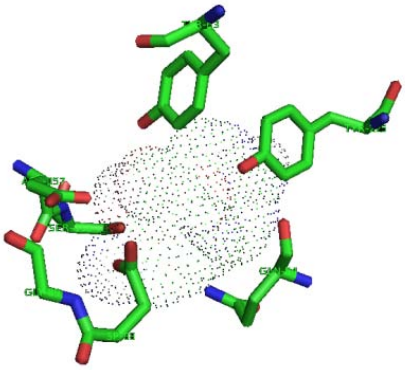
The comparison of the binding energies of 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine and AdoDATO reveals that 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine has a binding energy double that of AdoDATO and far greater than that of decarboxylated S-adenosylmethionine, meaning that in theory 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine should bind better to SpdSyn that either the substrate or a known inhibitor.

Examination of the structures of the molecule found in Table 3.11 reveals the absence of purines, oxygen atoms and sulphur found in decarboxylated S-adenosylmethionine and AdoDATO. The absences of these chemical moieties are possibly due to their absence or low numbers within the particular library and this is due to the origin of the particular library.

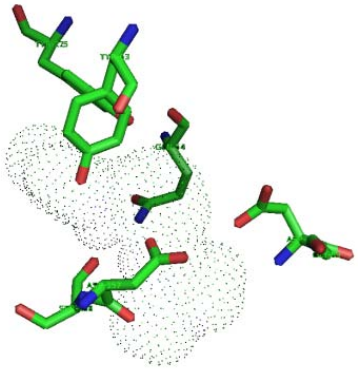
Figure 3.18 is a visualization of decarboxylated S-adenosylmethionine, AdoDATO and 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine docked into SpdSyn. In Figure 3.18 (A-D) it is observed that all three molecules are within the same position surrounded by the same amino acids. An overlay of all three molecules as shown in Figure 3.18D shows that nearly the same space is occupied by each molecule.



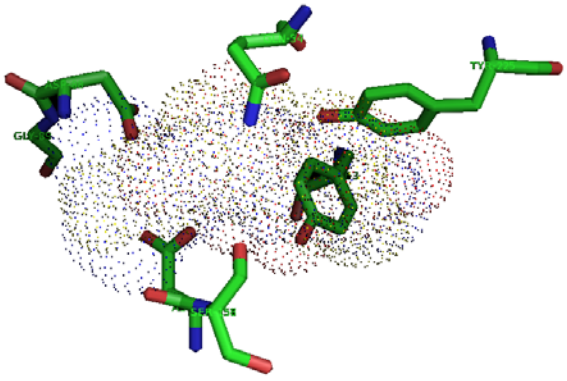
A



B



C



D

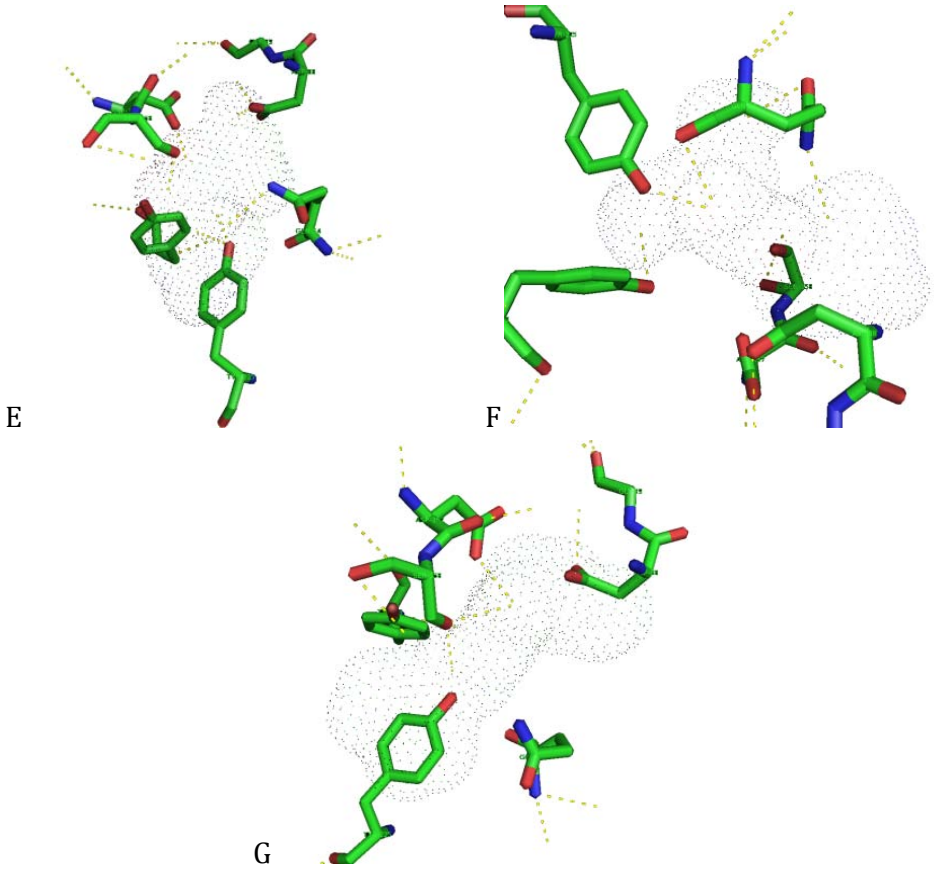


Figure 3.18 : Different substrates docked into SpdSyn. (A) AdoDATO docked into SpdSyn, surrounded by Gln54, Asp88, Asp157, Ser158 and Tyr225 (B) decarboxylated S-adenosylmethionine docked into SpdSyn, surrounded by Gln54, Asp88, Asp157, Ser158 and Tyr225 (C) 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine docked into SpdSyn, surrounded by Gln54, Asp88, Asp157, Ser158 and Tyr225 (D) Overlay of all three molecules AdoDATO shown in yellow decarboxylated S-adenosylmethionine in red and 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine as blue. (E) Possible polar interactions (yellow line) between decarboxylated S-adenosylmethionine and SpdSyn. (F) Possible polar interactions (yellow line) between AdoDATO and SpdSyn. (G) Possible polar interactions (yellow line) between 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine and SpdSyn.

The determination of polar interactions (Figure 3.18E-G) between SpdSyn and each of the other molecules reveals the possibility of five polar interactions between SpdSyn and AdoDATO this is two less than the seven possible interactions between SpdSyn and decarboxylated S-adenosylmethionine. It is observed that the possible interaction between ASP88 and ASP157 are not found in AdoDATO compared to decarboxylated S-adenosylmethionine. The reset of the possible polar interaction are the same between the two molecules. Looking at the possible polar interaction between 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-

yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine far less interaction are observed and the only two interactions are between SER158 and ASP157, this is indeed far less interactions than the other molecules .

Visualization (not shown) of 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine docked into SpdSyn reveals potential for the formation of hydrogen bond between ligand and macromolecule. It is also hypothesized that the nitrogen atoms present in 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine could be involved in a nucleophilic reaction.

11. Discussion

Inhibition of ODC by means of DFMO resulted in the life cycle of *Plasmodium* being interrupted at several stages. The *in vitro* effect of DMFO on *P. falciparum* was found to be more cytostatic than cytotoxic. This could be as a result of poor uptake of the drug or the use of exogenous polyamines by the parasite. Other studies have shown that inhibition of AdoMetDC can be therapeutic even if it has not been identified as primary target (Yeh and Altman, 2006).

Structural analogues show the highest interfering potential with regards to polyamine actions. A supposed benefit of polyamine analogues is that they might have numerous effects either by interfering with biosynthesis and salvage pathways or by interfering with binding of polyamines themselves. This combination of polyamine synthesis inhibitors and polyamine structural analogues is indeed a promising approach in the war against malaria (Muller, Coombs *et al.*, 2001; Yeh and Altman, 2006).

All five molecules of the polyamine pathway along with DMFO and MDL were subjected to similarity searches. Each molecule was entered as a SMILES-structure along with a Tc-value equal to 0.8 which changed to 0.6 due to libraries being too small to be of any significance. In general a Tc equal to 1 will result in a similarity search only producing the template molecule itself and by lowering the Tc cut off value, an increase in the number of hits will be obtained, and this is indeed no surprise as the stringency of similarity is decreased as the Tc cut off is decreased. In general a Tc-value equal to 0.8 is regarded as the minimum workable Tc. Any Tc value below 0.8 will resulted in a dramatic increase in the number of hits and the usefulness of these are questionable. Never the less for this particular study it was found that a Tc value equal to 0.8 produced far too little results. Therefore the Tc value was decreased to 0.6 which resulted in a dramatic increase in size of each library without being excessively large.

The results obtained from substructure searches should indeed not discourage the use of substructure searches but rather set limitations to the performance of FunGIMS substructure search capabilities.

A chemical library that contains hundreds of thousands of molecules can be useful depending on the intended use. In most cases a particular goal is set and thus requires molecules to have certain properties. In this case a library with hundreds of thousands of molecules that do not have the correct properties is not of much use. Thus libraries are filtered with regards to properties that have shown to be important to absorption and properties that produce drug-like or lead-like molecules. From the results obtained it is clear that filtering of libraries should be done with care and filtering should be adapted to each library.

Of interest is that DMFO itself was calculated to be a MCS, this was thought of as unusual at first, however this can be due to the unique nature of DMFO which would explain the small libraries obtained from similarity and substructure searches. Another important conclusion is that although

a series of chemical libraries were obtained and filtered to provide high quality libraries the process of discovering lead-like molecules does not end with a good library.

Spermidine synthase has been shown to be a likely candidate for chemotherapy and successful results have been achieved, such as the discovery of the inhibition of SpdSyn by 4MCHA and AdoDATO. The filtered library obtained from the similarity search using spermidine was used as it is believed to be representative as explained in section 7.6

In the docking study of spermidine synthase it was possible to dock 75% of the ligands in the relevant chemical library, the 25% that could not be docked were due to a number of different reasons such as charges that could not be assigned or that the ligand could not be successfully prepared for docking as described in section 7.6. A second attempt to prepare the 25% for docking resulted in the same errors resulting in the 25% to be excluded for the docking study. These errors could be as a result of an error within the original molecule file. Also the python script used to prepare the ligands for docking could not prepare these adequately and each molecule should be examined on its own.

The top ten docking results can be found in Table 3.11, of interest is the absence of decarboxylated S-adenosylmethionine, putrescine and spermidine, which shows that although decarboxylated S-adenosylmethionine, putrescine and spermidine were successfully docked the molecules found in Table 3.11 bind more strongly to spermidine synthase than decarboxylated S-adenosylmethionine, putrescine and spermidine. The docking study done should be regarded as a proof of concept. In other words the top ten molecules in Table 3.11 can be further examined to determine if each atom in the ligand is in a chemically favorable position as well as information regarding hydrogen-bonds formed. Although this docking study was only done as proof of concept it showed that it would be possible to successfully dock compounds from a chemical library produced by the FunGIMS system.

Comparing the binding energies of 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine to AdoDATO and decarboxylated S-adenosylmethionine Table 3.12. reveals that 2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine has a binding energy better than AdoDATO and decarboxylated S-adenosylmethionine respectively. A further comparison reveals that all the molecules in Table 3.11 have binding energies better than AdoDATO and decarboxylated S-adenosylmethionine, these results indicate that the compounds found in the libraries have the potential for further study.

By exploration of the chemical space of the polyamine pathway in malaria libraries containing potential lead-like molecules were produced and a docking study involving SpdSyn was done as proof of concept, yielding potentially useful molecules for further study.

12. Conclusion

The usefulness of the FunGIMS chemoinformatics module is noticeable in the results acquired by both similarity and substructure searches and by using these results in further studies. For all molecules explored a positive control was present and molecules with known inhibitory effects were among the results. Clustering of each library provides insight into the quality of each library respectively along with an overall structural representation by examination of the MCS for each library. A docking study of spermidine synthase was done as proof of concept and revealed 2,2,6,6-tetramethyl-N-[[3-[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine as the best docked ligand within the particular library. The approach taken with regards to choosing a library which would be used in docking was indeed a risk as most libraries are taken from substrate rather than product. In the end this seemed to be a valid and successful starting point.

For any of the molecules in the possible libraries to become useable lead-like compounds, further studies are required, including laboratory assays to determine the *in vitro* effect on *Plasmodium falciparum* cultures.

Chapter 4

This particular project can be divided into two main parts. The first part involved the development of the chemoinformatics workspace for the FunGIMS project. The second part entailed the use of the chemoinformatics module to explore the chemical space of the polyamine pathway in Plasmodium in order to find related compounds that could be used as lead compounds in chemotherapy against malaria.

The chemoinformatics module was intended to be used by research biologists to aid in their research and provide them with an easy-to-use chemoinformatics workspace, where they are 1) able to search databases, 2) perform similarity and substructure searches, 3) manage and store chemical data 4) study important molecular properties 5) and filter libraries with regards to specified criteria.

The development was done using rigid development practices and procedures and the primary design was object oriented. This led to a design that should not be difficult to understand and may be expanded upon by future developers. Instead of redeveloping existing packages, it was attempted to re-use and integrate existing packages and software. There are already numerous molecular viewers and editors, JMOL and JME were chosen and easily integrated. A difficulty arose when it came to the handling of molecules in order to perform different analyses, this was solved by the use of OpenBabel and Frowns and by their ability to work and process SMILES-structures. The use of SMILES-structures made it possible to increase the speed at which processes are completed, as the use of 3D-structure would have had a detrimental effect on computational time. By making FunGIMS a web-application, made possible by TurboGears, the appeal to research biologists was increased, as no specialized chemoinformatics software training is required.

The results obtained from exploring the chemical space of the polyamine pathway in *Plasmodium* provided a means to validate the chemoinformatics module of FunGIMS and was done successfully, showing that this could indeed be a potentially useful tool in library design.

Exploration of the chemical space of the polyamine pathway in search of related molecules based on structural similarity and substructure searches, resulted in libraries of molecules in which DMFO was present, showing that known molecules with inhibitory effects could be found by searches done based on similarity and substructure. Further filtering of the results was needed as the libraries could indeed contain molecules that are not likely to be lead-like compounds, several different filtering options are available in the chemoinformatics module.

Clustering of each library showed that the libraries are diverse enough to be of further use and the MCS showed that they are indeed representative of the polyamine pathway of Plasmodium. A docking study involving spermidine synthase and a library produced by similarity search of spermidine shows possibly promising results as it was able to dock 2,2,6,6-tetramethyl-N-[[3-

[[(2,2,6,6-tetramethylpiperidin-4-yl)amino]methyl]cyclohexyl]methyl]piperidin-4-amine with better scores than AdoDATO and decarboxylated S-adenosylmethionine.

This project showed that it was possible to develop a useful chemoinformatics tool that functions as web-application and can be used without specialized training in chemoinformatics software. By exploration of the chemical space of the polyamine pathway in Plasmodium insight into the chemical space was obtained, it also provided libraries that may contain lead-like compounds as shown by a docking study of SpdSyn. In contrast to general practice where compounds similar to the substrate are focused on, the focus is changed to compounds that are found similar to the product SpdSy. The results obtained from this illustrates that this could indeed also be a valid starting point in the discovery of lead-like molecules.

References

Websites and software

http://www.jmol.org/.
http://core.ecu.edu/phys/flurchickk/AtomicMolecularSystems/molecularStructures/molecularStructures.html.
http://www.molsoft.com/screenshots.html.
http://shoichetlab.compbio.ucsf.edu/inhibitor.php.
http://www.sqlalchemy.org/.
www.turbogears.org.
http://openbabel.org/wiki/Main_Page.
http://www.cdc.gov/malaria/biology/life_cycle.htm
http://www.nathnac.org/travel/factsheets/malaria.htm.
http://www.jhsph.edu/magazineFall01/Feature1.htm.

Literature

Abraham, M. H., K. Takacs-Novak and R. C. Mitchell (1997). On the partition of ampholytes: application to blood-brain distribution. *Journal of Pharmaceutical Sciences* **86**(3): 310-5.

Assaraf, Y. G., J. Golenser, D. T. Spira and U. Bachrach (1984). Polyamine levels and the activity of their biosynthetic enzymes in human erythrocytes infected with the malarial parasite, Plasmodium falciparum. *Biochemical Journal* **222**(3): 815-9.

Barnard, J. M. and G. M. Downs (1992). Clustering of Chemical Structures on the Basis of Two-Dimensional Similarity Measures. *Journal of Chemical Information and Computer Sciences* **32**: 644-649.

Barry, W. B. (1988). A Spiral Model of Software Development and Enhancement. **21**(5): 61-72.

Bebis, G., M. Georgiopoulos and N. V. Lobo (1998). Using self-organizing maps to learn geometric hash functions for model-based object recognition. *IEEE Trans Neural Netw* **9**(3): 560-70.

Bembenek, S. D., B. A. Tounge, S. J. Coats and C. H. Reynolds (2004). A Web-based chemoinformatics system for drug discovery. *Methods in Molecular Biology* **275**: 65-84.

Blake, J. F. (2000). Chemoinformatics - predicting the physicochemical properties of 'drug-like' molecules. *Current Opinion in Biotechnology* **11**(1): 104-7.

Blake, J. F. (2004). Intergrating cheminformatics analysis in combinatorial chemistry. *Current Opinion in Chemical Biology* **8**: 407-411.

Borghini, A., D. Pietra, P. Domenichelli and A. M. Bianucci (2005). QSAR study on thiazole and thiadiazole analogues as antagonists for the adenosine A1 and A3 receptors. *Bioorganic & Medical Chemistry* **13**: 5330-5337.

Brown, F. K. (1998). Chemoinformatics: what is it and how does it impact drug discovery. *Annual Reports in Medicinal Chemistry* **33**: 375-384.

Burger, P. B., L. M. Birkholtz, F. Joubert, N. Haider, R. D. Walter and A. I. Louw (2007). Structural and mechanistic insights into the action of Plasmodium falciparum spermidine synthase. *Bioorganic & Medicinal Chemistry* **15**(4): 1628-37.

Byers, T. L., P. Casara and A. J. Bitonti (1992). Uptake of the antitrypanosomal drug 5'-([(Z)-4-amino-2-butenyl]methylamino)-5'-deoxyadenosine (MDL 73811) by the purine transport system of Trypanosoma brucei brucei. **283**(3): 755-0.

Clark, D. E. (1999). Rapid calculation of polar molecular surface area and its application to the prediction of transport phenomena. 1. Prediction of intestinal absorption. *J Pharm Sci* **88**(8): 807-14.

Clark, D. E. (1999). Rapid calculation of polar molecular surface area and its application to the prediction of transport phenomena. 2. Prediction of blood-brain barrier penetration. *Journal of Pharmaceutical Sciences* **88**(8): 815-21.

Clark, D. E. and S. D. Pickett (2000). Computational methods for the prediction of 'drug-likeness'. *Drug Discovery Today* **5**(2): 49-58.

Cramer, R. D., D. E. Patterson, R. D. Clark, F. Soltanshahi and M. S. Lawless (1998). Virtual Compound Libraries: A New Approach to Decision Making in Molecular Discovery Research. *Journal of Chemical Information and Computer Sciences* **38**(6): 1010-1023.

Dalby, A. N. J. G., W. D. Hounshell, A. K. I. Gusdhurst, D. L. Grier, B. A. Leland and J. Laufer (1992). Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited. *Journal of Chemical Information and Computer Sciences* **32**(3): 244-255.

David, W. (1988). SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. **28**(1): 31-36.

Degtyarenko, K., P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcantara, M. Darsow, M. Guedj and M. Ashburner (2008). ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research* **36**(Database issue): D344-50.

DeWitte, R. S. (2006). Avoiding physicochemical artefacts in early ADME-Tox experiments. *Drug Discovery Today* **11**(17/18): 855-859.

DiMasi, J. A., R. W. Hansen and H. G. Grabowski (2003). The price of innovation: new estimates of drug development costs. *Journal of Health Economics* **22**(2): 151-85.

Duchowicz, P. R., A. Talevi, C. Bellera, L. E. Bruno-Blanch and E. A. Castro (2007). Application of description based on Lipinski's rules in the QSPR study of aqueous solubilities. *Bioorganic & Medical Chemistry* **15**: 3711-3719.

Dufe, V. T., W. Qiu, I. B. Muller, R. Hui, R. D. Walter and S. Al-Karadaghi (2007). Crystal structure of Plasmodium falciparum spermidine synthase in complex with the substrate decarboxylated S-adenosylmethionine and the potent inhibitors 4MCHA and AdoDATO. *Journal of Molecular Biology* **373**(1): 167-77.

Ertl, P. (1998). World Wide Web-based system for the calculation of substituent parameters and substituent similarity searches. *Journal of Molecular Graphics & Modelling* **16**(1): 11-3, 36.

Ertl, P., J. Muhlbacher, B. Rohde and P. Selzer (2003). Web-based cheminformatics and molecular property prediction tools supporting drug design and development at Novartis. *SAR and QSAR in Environmental Research* **14**(5-6): 321-8.

Ertl, P., B. Rohde and P. Selzer (2000). Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. *Journal of Medicinal Chemistry* **43**(20): 3714-7.

Flower, D. R. (1998). On the Properties of Bit String-Based Measures of Chemical Similarity. **38**(3): 379-386.

Gasteiger, J. (2003). A Handbook of Chemoinformatics, Electronic Book.

Gasteiger, J. (2006). The central role of chemoinformatics. *Chemometrics and Intelligent Laboratory Systems* **82**: 200-209.

Geldenhuysa, W. J., K. E. Gaaschb, M. Watsonb, D. D. Allena and C. J. Van der Schyfa (2006). Optimizing the use of open-source software applications in drug discovery. *Drug Discovery Today* **11**(3/4): 127-132.

Griffen-Smith, B. (2008) "Malaria-Free Mosquitoes." The Journal of Young Investigators_Volume, DOI:

Guha, R., M. T. Howard, G. R. Hutchison, P. Murray-Rust, H. Rzepa, C. Steinbeck, J. Wegner and E. L. Willighagen (2006). The Blue Obelisk-interoperability in chemical informatics. *Journal of Chemical Information and Modeling* **46**(3): 991-8.

Gunturi, S. B., R. Narayanan and A. Khandelwal (2006). In Silico ADME modelling : Computational models to predict human serum albumin affinity using ant colony systems. *Bioorganic & Medical Chemistry* **14**: 4118-4129.

Hann, H. and R. Green (1999). Chemoinformatics - a new name for an old problem? *Current Opinion in Chemical Biology* **3**: 379-383.

Heby, O., S. C. Roberts and B. Ullman (2003). Polyamine biosynthetic enzymes as drug targets in parasitic protozoa. *Biochemical Society Transactions* **31**(2): 415-9.

Hrib, N. J. and N. P. Peet (2000). Chemoinformatics: are we exploiting this new science? *Drug Discovery Today* **5**(11): 483-485.

Ikeguchi, Y., M. C. Bewley and A. E. Pegg (2006). Aminopropyltransferases: function, structure and genetics. *Journal of Biochemistry* **139**(1): 1-9.

Jaworska, J., N. Nikolova-Jeliazkova and T. Aldenberg (2005). QSAR applicability domain estimation by projection of the training set in descriptor space: A Review. *American Theological Library Association* **33**: 445-459.

Jones, A. R., M. Miller, R. Aebersold, R. Apweiler, C. A. Ball, A. Brazma, J. Degreef, N. Hardy, H. Hermjakob, S. J. Hubbard, P. Hussey, M. Igra, H. Jenkins, R. K. Julian, Jr., K. Laursen, S. G. Oliver, N. W. Paton, S. A. Sansone, U. Sarkans, C. J. Stoeckert, Jr., C. F. Taylor, P. L. Whetzel, J. A. White, P. Spellman and A. Pizarro (2007). The Functional Genomics Experiment model (FuGE): an extensible framework for standards in functional genomics. *Nature Biotechnology* **25**(10): 1127-33.

Langer, T. and R. D. Hoffmann (2001). Virtual Screening: An Effective Tool for Lead Structure Discovery? *Current Pharmaceutical Design* **7**: 509-527.

Lennernas, H. (1997). Human jejunal effective permeability and its correlation with preclinical drug absorption models. *The Journal of Pharmacy and Pharmacology* **49**(7): 627-38.

Lennernas, H. (1998). Human intestinal permeability. *Journal of Pharmaceutical Sciences* **87**(4): 403-10.

Leonard, J. T. and K. Roy (2006). Comparative QSAR modeling of CCR5 receptor binding affinity of substituted 1-(3,3-diphenylpropyl)-piperidinyl amides and ureas. *Bioorganic & Medical Chemistry* **16**: 4467-4474.

Maldonado, A. G., J. P. Doucet, M. Petitjean and B. T. Fan (2006). Molecular similarity and diversity in chemoinformatics: from theory to applications. *Molecular Diversity* **10**(1): 39-79.

Marshall, G. R. (2004) "Introduction to Chemoinformatics in Drug Discovery." Chemoinformatics in Drug Discovery Volume, DOI:

McMurry, J. (1999). Organic Chemistry. Organic Chemistry, Brooks/Cole Thomson Learning.

Miteva, M. A., S. Violas, M. Montes, D. Gomez, P. Tuffery and B. O. Villoutreix (2006). FAF-Drugs: free ADME/tox filtering of compound collections. *Nucleic Acids Research* **34**(Web Server issue): W738-44.

Muller, S., G. H. Coombs and R. D. Walter (2001). Targeting polyamines of parasitic protozoa in chemotherapy. *Trends in Parasitology* **17**(5): 242-9.

Muller, S., A. Da'dara, K. Luersen, C. Wrenger, R. Das Gupta, R. Madhubala and R. D. Walter (2000). In the human malaria parasite Plasmodium falciparum, polyamines are synthesized by a bifunctional ornithine decarboxylase, S-adenosylmethionine decarboxylase. *The Journal of Biological Chemistry* **275**(11): 8097-102.

P Willet (1998). Chemical Similarity Searching. *Journal of Chemical Information and Computer Sciences* **38**: 983-996.

Padron, J. A., R. Carrasco and R. F. Pellon (2002). Molecular descriptor based on a molar refractivity partition using Randic-type graph-theoretical invariant. *The Journal of Pharmacy and Pharmacology* **5**(3): 258-66.

Python Python Programming Language.

Ritchie, T. J. (2001). Chemoinformatics: manupulating chemical information to facilitate decision-making in drug discovery. *Drug Discovery Today* **6**(16): 813-814.

Sanner, M. F. (1999). Python: a programming language for software integration and development. *Journal of Molecular Graphics & Modelling* **17**(1): 57-61.

Smith, C. (2002). Cheminformatics: Redefining the Crucible. *The Scientist* **16**(8): 40-48.

Stahl, M. and H. Mauser (2005). Database Clustering with a Combination of Fingerprint and Maximum Common Substructure Methods. *Journal of Chemical Information and Modeling* **45**: 542-548.

Stahura, F. L. and J. Bajorath (2002). Bio- and chemo-informatics beyond data management: crucial challenges and future opportunities. *Drug Discovery Today* **7**(11): s41-s47.

Sufrin, J. R., S. R. Meshnick, A. J. Spiess, J. Garofalo-Hannan, X. Q. Pan and C. J. Bacchi (1995). Methionine recycling pathways and antimalarial drug design. *Antimicrob Agents Chemotherapy* **39**(11): 2511-5.

Tetko, I. V., P. Bruneau, H. Mewes, D. C. Rohrer and G. I. Poda (2006). Can we estimate the accuracy of ADME-Tox predictions? *Drug Discovery Today* **11**(15/16): 700-707.

Tripos (2005) "SYBYL 7.1." **Volume**, DOI:

Tuteja, R. (2007). Malaria - an overview. *The FEBS Journal* **274**(18): 4670-9.

Willett, P. (1998). Chemical Similarity Searching. *Journal of Chemical Information and Computer Sciences* **38**: 983-996.

Willett, P. (2000). Chemoinformatics - similarity and diversity in chemical libraries. *Current Opinion in Biotechnology* **11**(1): 85-8.

Winiwarter, S., N. M. Bonham, F. Ax, A. Hallberg, H. Lennernas and A. Karlen (1998). Correlation of human jejunal permeability (in vivo) of drugs with experimentally and theoretically derived parameters. A multivariate data analysis approach. *Journal of Medicinal Chemistry* **41**(25): 4939-49.

Xue, L. and J. Bajorath (2000). Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. *Combinatorial Chemistry & High Throughput Screening* **3**(5): 363-72.

Yeh, I. and R. B. Altman (2006). Drug Targets for Plasmodium falciparum: a post-genomic review/survey. *Mini Reviews in Medicinal Chemistry* **6**(2): 177-202.