Discovery of medicinal molecules based on similarity networks and metrics

ICS5200 - Dissertation Progress Report

Joseph D’Emanuele
Intelligent Computer Systems
Faculty of ICT
University of Malta
joseph.demanuele.08@um.edu.mt

ABSTRACT

Computer-aided drug design (CADD) projects start with a computational search for active molecules that form a complex with a protein to trigger a response or block its function. This report presents the initial investigation being carried out to build a multi-level and scalable approach where multiple small-molecules and proteins similarity networks are bridged via known bindings. Through these networks, we envision a research to be able to discover new putative small-molecule binders for a given protein target, and to find other proteins which are likely to interact with a given small-molecule. Amongst other things, the latter is also important for side-effect prediction. A software tool which aids visualization of these networks will also be built.

Keywords

AI; Big Data; Chemoinformatics; Similarity networks; Virtual Screening

1. INTRODUCTION

We take medicines to cure diseases and to ease symptoms of an illness. In essence, medicines, or drugs, are compounds that interact with a biological system to produce a biological response. This interaction happens at the molecular level of living beings. It involves a ligand, typically a small-molecule, to bind with a targeted protein, a macromolecule that execute a particular cellular function. When this binding occurs, it triggers (or blocks) a biological response.

Drug discovery and development is a lengthy, laborious, and expensive process. For some disease areas, such as antibacterial agents [1], the total cost to market a new drug cannot be recuperated in sales. In recent years, the use of *in silico* chemistry and molecular modelling for computer-aided drug design (CADD), chemoinformatics, gained a significant momentum [2]. Medicinal clinics and laboratories are collecting more information about their findings and diagnostics, producing a large volume and valuable data for drug design. Large organisations such as the European Bioinformatics Institute (EBI) and National Center of Biotechnology Information (NCBI) in the US, formed with their mission to collect and publish freely public biological data to the scientific community.

Drug design starts by identifying a valid target, such as a protein which has a link with the disease under scrutiny. Once this target is identified, scientists start to search for a compound that when it binds with it, a biological response is triggered or blocked. Considering the enormous chemical space of 10^{60} compounds, finding a molecule that triggers the required biological response is not an easy task. Although only a portion of this space is appropriate to interact with biological targets, the number of valid compounds is still huge in the range of tens of millions of compounds. For example, ZINC database contains over 100 million purchasable compounds. An *in silico* drug design process that facilitates this search is virtual screening, the computational analogue of biological screening. Its aim is to filter a set of molecules from a larger set by scoring and ranking these structures using computational algorithms to help researchers take decisions on the tasks being carried out, such as deciding which compounds to purchase from external sources. It enables medicinal chemists to select a much smaller and manageable set to work with from the huge chemical space in the scale of millions, making it popular and important in the drug discovery process as reported by Jorgensen [18]. Virtual screening approaches can be divided into two main techniques, namely, structure-based and ligand-based [22]. Both approaches bring their own challenges. A major difficulty with structure-based approach is due to the many degrees of freedom involved in docking two irregular shapes in a 3D space. On the other hand, ligand-based techniques are indirect drug design methods that rely on the knowledge of other ligands that are known to bind with the target protein. Searching for similar ligands is especially challenging due to the multi-parameter combinatorial explosion of possible ligand-to-ligand matching. The more molecular parameters one wants to match, such as weight, shape, and structure, the bigger are the computational task.

Motivated by the importance of ligand-based virtual screening in drug design and the challenges aforementioned, this research exploits the benefits of clustered computing and the use of different search algorithms to propose a novel approach to discover new putative ligand binders for a given target protein.

The progress report presented here is split into seven sections. After this brief introduction to the subject, the aims and objectives of the study are presented. Next is the background research and literature review followed by the proposed solution together with an overview of the dataset being used and resources needed. Following these is an evaluation plan, a list of references, and a project timeline.

2. AIMS AND OBJECTIVES

2.1 Aims

Exploiting the benefits of ligand-based virtual screening and the large volume of public data available in various molecular chemistry databases, the aim of this study is to research a novel approach to discover new putative ligand binders for a given target protein from multiple similarity search algorithms results.

2.2 Objectives

The aims above will be accomplished by fulfilling the following objectives:

- Build a multi-level and scalable platform accessible via a web application through which it will be possible to discover new putative ligand binders for a given target protein.
- Determine the significance of using multiple ligand similarity algorithms to find a putative target.
- Investigate the effectiveness of different protein similarity algorithms to find new lead drug compounds.

3. BACKGROUND RESEARCH AND LITERATURE REVIEW

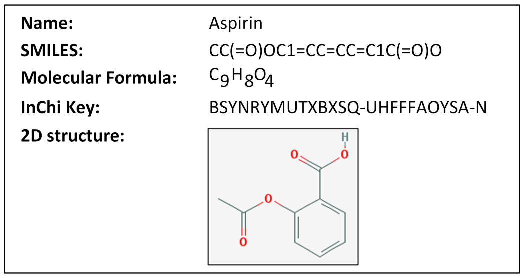
In 1998, Brown coined the term *chemoinformatics* to define "the mixing of information resources to transform data into information, and information into knowledge, for the intended purpose of making better decisions faster in the arena of drug lead identification and optimisation" [8]. Today, chemoinformatics has a much broader sense, and is defined as "the application of informatics methods to solve chemical problems" [12].

At the heart of computational drug design and discovery is virtual screening. Data for virtual screening comes from several sources. Of particular interest to this research is the ChEMBL Open Data database [13]. The data in this database is regularly updated by manually abstracting binding, functional, and ADMET (absorption, distribution, metabolism, excretion, and toxicity properties as assessed in *in vivo* reports) information from primary published literature. This data is then further curated and standardized. Standardization allows data from different databases to be compared and used together. Wilton *et al* have suggested that there are three classes under ligand-based virtual screening methods (machine learning techniques, pharmacophore-based design, and similarity searching) and one class under structure-based methods (protein-ligand docking) [34]. All these methods depend on the amount of structural and bioactivity data available.

Structure-based drug design is about identifying a compound for *in vitro* testing based on the knowledge of the drug's 3D structure. The process involves docking of candidate ligands to a target and through scoring functions estimate the likelihood that this binding trigger or block a biological response [19, 20].

Pharmacophore-based techniques involve the creation of a model that contains the molecular features required for structurally diverse ligands to likely bind to a common receptor site on the target protein [29]. Chemical similarity searching approach offers a complementary alternative to pharmacophore-based technique. Here, a query compound is used to search a database of compounds to find similar ones [33]. The result is then sorted in decreasing similarity order and the top compounds are said to be likely to exhibit the same activity by Patterson's neighbourhood behaviour principle [25]. Machine learning techniques' objective is to construct a model that can identify relationships between the chemical structure and the observed activity. Leach and Gillet [22] presents a survey how these algorithms are used in chemoinformatics applications. In these techniques, data from High-Throughput Screening (a mass automated system for *in vitro* screening) is often classified as 'active' or 'inactive'. The aim is to derive a mathematical model that predicts the activity class of new structures. Additionally, methods for non-specific targets have been developed. An example of such a technique is the prediction of the likelihood that a molecule has "drug-like" characteristics and possesses desired physiochemical properties. Amongst these methods are substructure filters to eliminate molecules that are known to be inappropriate starting points for drug discovery and Lipinski's rule of five which describe the molecular properties important to drug like compounds [23]. Several efforts have been made to combine ligand and structure based virtual screening in order to exploit the benefits of both techniques, example [11, 30].

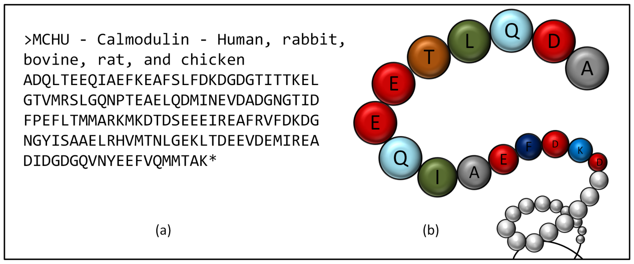
As ligand databases became more popular, there was the need to standardise virtual representation of small molecules. In 1985, Wiswesser proposed an improved system over the 1954 Wisesser Line Notation named Simplified Molecular Input Line System (SMILES) [32]. SMILES is nowadays a standard line notation for representing small molecule structures using ASCII strings (see Figure 1 for an example). It must be noted that there are usually different but equally valid SMILES descriptors for the same structure, thus making SMILES not ideal for indexing the molecules. For example, the structure of ethanol can be represented as C(O)C, CCO, and OCC. Canonicalization algorithms have been developed aiming to create a unique SMILES string. However, the canonical SMILE representation depends on the canonicalization algorithm used. In 2005, the International Chemical Identifier (InChi) algorithm was released as open source [15]. The purpose of InChi and the hash-key version InChiKey is to provide an unambiguous way to index and search all chemical structures.



Name: Aspirin
SMILES: CC(=O)OC1=CC=CC=C1C(=O)O
Molecular Formula: $C_9H_8O_4$
InChi Key: BSYNRYMUTXBXSQ-UHFFFAOYSA-N
2D structure:

Figure 1: Molecular representation for Aspirin

As SMILES describe ligands, the FASTA format represents a linear sequence of amino acids in a protein, see Figure 2. Amino acids, the building blocks of proteins, are represented using single ASCII characters. A sequence is divided into two parts, a description line and the sequence representation. The Protein Data Bank (PDB), established in 1971 is one of the most commonly used database for protein structures.



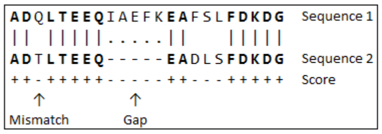
```
>MCHU - Calmodulin - Human, rabbit,
bovine, rat, and chicken
ADQLTEEQIAEFKEAFSLFDKDGDGTITTKEL
GTVMRSLGQNPTEAELQDMINEVDADGNGTID
FPEFLTMMARKMKDTDSEEEIREAFRVFDKDG
NGYISAAELRHVMTNLGEKLTDEEVDEMIREA
DIDGDGQVNYEEFVQMMTAK*
```

Figure 2: (a) FASTA and (b) Amino Acid sequence

Similarity searching is central to medicinal chemistry [24], and as in almost any other field, it is very subjective. As such, many computational similarity methods have been introduced. Maggiora *et al.* point out that there are three basic components required for a suitable molecular similarity measure: molecular or chemical feature representation, a potential weighing of features, and a similarity function [24]. Further, in their perspective paper, Maggiora and his team identify seven views of similarity: chemical, molecular, 2D, 3D, biological, global, and local similarities. They discuss at length the difference in similarity perspective between a trained medicinal chemist and that obtained by computational means. Kutchukian *et al.* argue that medicinal chemists intuition is critical for the success of modern drug discovery as interpreting in chemical terms the result of a "black-box approach" of many machine learning techniques is many-a-times impossible to do [21]. To address the subjectivity issue, algorithms that yield a numerical readout that can quantify similarity are used. One popular, simple, and fast function is the Tanimoto coefficient (Tc) [31]. It compares the features of two molecules and returns a score between 0 (no similarity) and 1 (similar). A Tc value of 0.85 is a commonly used threshold that reflects a high probability that two molecules share the same activity [24]. Other functions, such as torsion fingerprint deviation compares the molecular shape [27]. Molecular fingerprints are used as an input to similarity functions. A fingerprint is usually a bit-vector, sometimes as large as 4kbits, that represent the presence or absence of particular features in a molecule. For example, if 101101_2 is the molecular fingerprint for m_1 and 001111_2 of m_2, where each bit represents the presence (1) or absence (0) of a specific molecular feature, using equation 1 produces a Tanimoto coefficient of $\frac{2}{3}$.

$$Tc(X,Y) = \frac{\sum_i (X_i \wedge Y_i)}{\sum_i (X_i \vee Y_i)} \tag{1}$$

Proteins in FASTA format, represented by a long string of ASCII characters are matched using sequence alignment algorithms, such as the Basic Local Alignment Search Tool (BLAST) [3]. At the core of a sequence alignment algorithm is the scoring system. A simple and basic approach is to increase the score when two sequence parts match and deduct the score when a mismatch or a gap in sequence is encountered. Figure 3 demonstrates a simple alignment. Scores are typically stored in a scoring matrix and can be calculated both heuristically and probabilistically. Substitution matrices, such as point acceptance mutation (PAM) [10] and blocks substitution matrix (BLOSUM) series [16], are normally used to add weights to matching score in terms of gene mutation.



```
ADQLTEEQIAEFKEAFSLFDKDG   Sequence 1
|| |||||.....||   |||||
ADTLTEEQ-----EADLSFDKDG   Sequence 2
++-+++++-----++---+++++   Score
  ↑          ↑
Mismatch     Gap
```

Figure 3: Simple sequence alignment

The next section discusses the work carried out so far and the proposed solution.

4. PROPOSED SOLUTION

This section discusses the practical part of the study. It outlines the tests and evaluations done so far, the proposed solution, the dataset, and the hardware and software needed. A plan to complete the research is given in Appendix A.

4.1 Practical research carried out

After researching the main topics discussed in the "Background Research and Literature Review" section, several practical tasks and evaluations were carried out.

The first practical task was to create a ligand similarity graph. For this task, one hundred random compounds in SMILES format from the ZINC database were downloaded and a Morgan fingerprint for each was computed. Then, an upper triangular similarity matrix using Tanimoto function was created and was used to build a graph in Neo4j, a highly scalable native graph database. Similarly, a sample of protein FASTA files were used to create a protein similarity matrix using BLAST algorithm and imported into Neo4j to graph the relationship between proteins.

Next, the ChEMBL database was used to extract ligand-protein bindings. This allowed the bridging between the ligands and the proteins graphs. Figure 4 shows these components and their expected similarity matrix dimensions using ChEMBL data. A simple web interface, similar to Figure 8, was created using D3 javascript library to represent graphs.

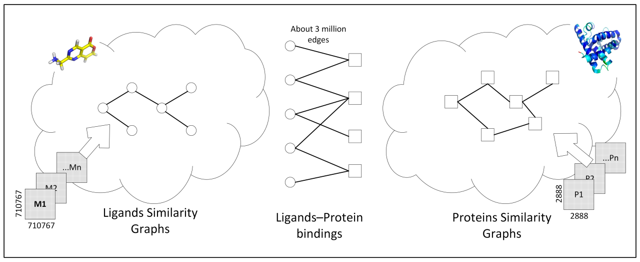


Figure 4: Ligands and Protein Graphs

The large volume of data, for example 24GB of compressed ASCII sequences of non-redundant protein sequence dataset, and the massive amount of similarity computation tasks requires a scalable solution. Apache Spark proved to be a good fit for this purpose. An environment with two Spark worker nodes and one master node was created. Each node ran a number of similarity functions written as PySpark jobs and the result was consolidated at the master node. The prototype cluster ran on a local machine and no benchmarks were taken; the purpose of this test was of a proof of concept nature. At this point, GraphFrames, a graph API that runs on top of Apache Spark was evaluated. A test similar to the one done using Neo4j was carried out and proved to be a satisfactory alternative with the benefit that there is no need for the data to be duplicated on two systems; one on Neo4j for graph processing and analysis, and another on Apache Hadoop's HFDS for use by Spark jobs.

4.2 Solution outline

The proposed solution evolves around the known ligand-protein bindings and the similarity concept; the more features that one can match, the higher the probability that the query ligand can bind with the target protein.

Two datasets, one for ligands L and another for proteins P are prepared. For set L, a number of similarity matrices M with dimension $|L| \times |L|$ are created. Each matrix $m \in M$ represents the similarity coefficients of each ligand $l \in L$ with all other ligands in $L - l$ for a particular feature f. For example, a matrix m_1 can represent the similarity coefficients of all ligands in the dataset using Morgan fingerprints and Tanimoto function, while another matrix, m_2 can represent the 3D-shape similarities using torsion fingerprint deviation. Given a query ligand q and a threshold ξ, one can find similar ligands in m_f such that $sim_q = \{\forall\ l, (q \sim l) \leq \xi\}$, where '$\sim$' can be any similarity function of choice. A typical process to create a similarity matrix is outlined next:

```
1. For $n$:  1 to $|L|$ do:
    1.1. $f_{l_n} = fingerprint(l_n)$
2. For $m$:  1 to $|L|$ do:
    2.1. For $n$:  1 to $|L|$ do:
        2.1.1. $sim(m,n) = f_{l_m} \sim f_{l_n}$
```

Figure 5: Simple similarity matrix algorithm

In Figure 5, f_l is a Morgan or other molecular fingerprint and *sim* is a similarity function such as Tversky. If *sim* is a symmetrical function such as Tanimoto, the inner loop at step 2.1. can be optimised to start from $m+1$ and create an upper triangular matrix instead a full one.

Graph G is a weighted bipartite graph joining sets L and P, where the weights represent the binding affinity between ligands and proteins. Given that $sim_q \subset L$, using G, one can walk from ligands in set sim_q to P, see Figure 6. Using the similarity knowledge about sim_q and the links in G, one can suggest that the missing links in G can possibly be putative bindings. For example, in Figure 6, given that q is similar to $l1, l2, l3$, and $l4$, using Petterson's similarity principle [25] one can say that there is a possibility that q binds to $p1$, $p2$, and $p3$. Further, doing the above test using different similarity functions, one can say that the more frequent a particular edge is present across different runs, the higher the probability that this binding can be successful in *in vitro* testing. Likewise, this hypothesis is valid to start from looking for similar $p \in P$ using similarity algorithms such as BLAST and walk back to L to find possible ligand binders.

Using Figure 6 and the above description, the proposed search solution can algorithmically be listed as in Figure 7.

The final solution shall provide a web application to allow similarity algorithms parameter configuration, user input for querying data, and result visualisation. The main features of this user interface are summarised in Table 1 and a sample UI mock is shown in Figure 8.

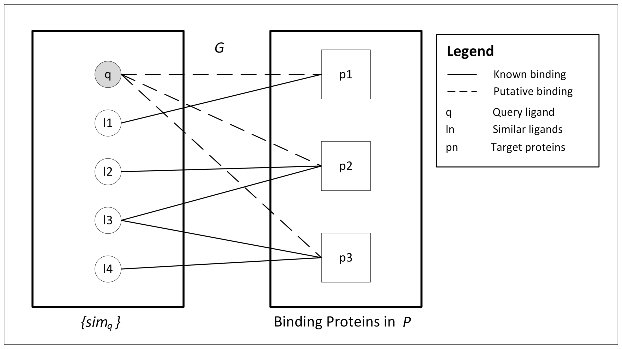


Figure 6: Putative Ligands-Protein binding

```
1. Let $q$ be the query ligand.
2. Find all ligands similar to $q$ in $L$ using a particular simi-
   larity algorithm and threshold $\xi$.
3. Let $\{sim_q\}$ be the resultant set from step 2.
4. Using the neighbourhood similarity principle, one can say
   that $q$ can potentially binds with all proteins that ligands
   in $\{sim_q\}$ bind to.
5. Repeat steps 2. to 4. for different similarity algorithms.
   The more frequent a binding edge appears in different sim-
   ilarity algorithms, the higher the probability that the
   binding is successful in in vitro testing.
```

Figure 7: Search algorithm

4.3 Dataset

Throughout this research, ChEMBL data (version 22) is used [13]. First, data related to Homo Sapiens (identified by taxonomy id 9606) was extracted and categorised into two sets, namely the ligands set L and the proteins set P. The number of samples extracted and cleaned from ChEMBL are described in Table 2. These two sets form the disjoint sets of a bipartite graph G, whose edges correspond to the ligand-protein binding, each weighted by the complex formation affinity recorded in ChEMBL. The edges are represented by set E in Table 2.

With the ligands and proteins datasets at hand different similarity matrices will be created. This is a long process that creates a lot of data and requires a lot of resources. The data structures are estimated to be a $710,767 \times 710,767$ matrix for each ligand similarity matrix, a little less than 3 million weighted edge bipartite graph, and a directed weighted graph of 2,888 nodes per protein similarity graph.

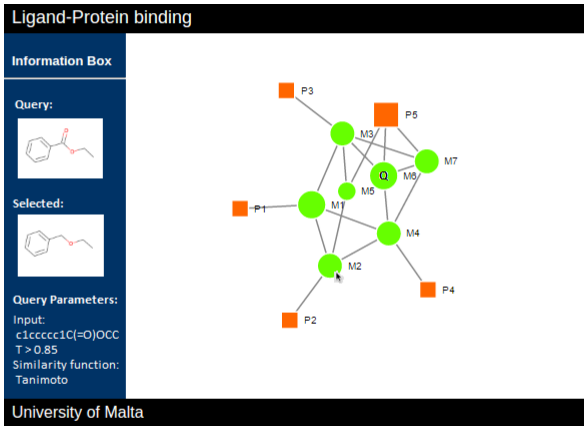
4.4 Hardware and Software

The prototypes can be carried out on normal i7 desktop machines. However, to handle the full dataset, a larger infrastructure is required. The plan is to run the full solution on an Apache Hadoop cluster [5] and the similarity functions, see Figure 5 step 2.1.1., run as Apache Spark jobs [6]. The latter is chosen over MapReduce for the following reasons:

- Spark does more processing in memory and uses less disk IO, making it faster. An interesting function of Spark is that it does lazy evaluation. That is, Spark only loads the data when it is asked to be returned from a function. Thus, Spark is able to optimise multiple map transformations and reduce operations by analysing if it can skip or merge certain tasks.
- Spark's real-time characteristic facilitates the similar-

Table 1: Visual tool features

Ligands	Bindings	Target Proteins
Similarity matrices for ligands: • Different similarity measures, such as, Tanimoto, Dice, and Tvesky. Graph Visualisation: • Node size = node degree • Node shape = circle • Filter by similarity factor, e.g. Tc >0.85 • Group by ligand's compound properties • On node hover show molecule data, such as structure and properties	Ligands-Proteins binding is a weighted bipartite graph built from ChEMBL data. Visualisation tool features: • Filter by binding strength • Filter by node (ligand or protein) degree Graph: • Edge width = binding strength	Similarity matrix for proteins using BLAST+ and/or other algorithms Visualisation: • Group by protein classification • Filter by protein classification Graph: • Node size = node degree • Node shape = square • Node color = protein classification • On node hover show protein data


Ligand-Protein binding

Information Box

Query:

Selected:

Query Parameters:
Input:
c1ccccc1C(=O)OCC
T > 0.85
Similarity function:
Tanimoto

University of Malta

Figure 8: Visualization mock

Table 2: ChEMBL data (Homo Sapiens)

Set	Number of items in set
Ligands L	710,767
Proteins P	2,888
Bindings E	2,930,127

ity matrix creation jobs execution.

- A big advantage of Spark is that it unifies a lot of interfaces like SQL and Graph frameworks into a single abstraction of Resilient Distributed Data. Thus it removes the need of multiple independent solutions and the overhead to integrate the results.

All machines will run Ubuntu open source operating system. Cluster automation scripts will be written in Bash, the visual tool in HTML5 and leveraging D3 potential to display interactive graphs, back-end programming will be carried out in Python, and documentation in LaTeX. Other packages that will be used during the research include:

- Spark's package GraphFrames [4] will be used to process graph data on Hadoop.
- RDKit library [26] to create ligand fingerprints and run similarity functions.
- BIOPython [9] is used to process FASTA files and run BLAST queries.

5. EVALUATION PLAN

Evaluation of the proposed solution is not straight forward. The aim of the research is to help medicinal chemists to discover new possible ligand-protein bindings, and measuring novelty is very difficult [7]. This is because the research is recommending a binding that the user does not know about.

At this juncture, it is important to note that the absence of a binding between a ligand and a protein in ChEMBL database does not mean that the pair do not bind, but it can mean that this binding was never recorded. This is analogous to the challenge imposed by offline recommender systems prediction evaluation [14]. To assume that the absence of a ligand-protein link in ChEMBL means that the pair do not form a complex inflate the number of false positives. Thus, evaluation metrics that are based on false positive outcome do not fully apply to this type of solution [14].

The evaluation is to be done by filtering out known bindings from the dataset and confirm that the solution returns these bindings in the top recommended results. The metric will be to count the number of putative bindings discovered out of the total number of bindings in the testing dataset. For this, the k-fold cross validation technique will be used. For each k experiments, k-1 folds will be used as the known ligand-protein binding set and the remaining fold will be used for testing.

Two types of experiments will be carried out with the aim to estimate the accuracy of recommending molecules that are likely to bind with a given target and vice-versa. The first type tests the proposed solution's putative target protein recommendation accuracy. Here, the ligands set will be k-folded and the unseen set will be fed to the system (as described in Figure 7) and confirm the recommended bindings. A successful recommendation is considered to be one that there is a known binding in ChEMBL database. The second type of experiments is a mirror image of the first. That is, the proteins set will be k-folded. As in the first type of experiments, a successful recommendation is considered to be one for which a binding exist in the full dataset.

The ultimate evaluation can be the testing of a newly unseen small-molecule and find putative targets and confirm the result via a competitive assay binding experiment [17]. This latter set of experiments requires wet laboratory resources which are not available at this time. If the possibility arises, a great evaluation will be to ask a group of medicinal researchers to use the software tool being developed and pass their feedback about the tool as a whole, its performance, and if it helped them in their research.

6. REFERENCES

[1] Session 3 (R & D): industry perspective on PPPs and the link between new business models and the regulatory framework.

[2] W. A, A. N, S. L, A. A, H. H, and S. S. In-silico drug design: An approach which revolutionarised the drug discovery process. *OA Drug Design and Delivery*, Sep 2013.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, Oct 1990.

[4] Apache Software Foundation. GraphFrames, Nov 2016.

[5] Apache Software Foundation. Hadoop, Aug 2016.

[6] Apache Software Foundation. Spark, Nov 2016.

[7] I. Avazpour, T. Pitakrat, L. Grunske, and J. Grundy. *Dimensions and Metrics for Evaluating Recommendation Systems*, pages 245–273. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[8] F. K. Brown. Chapter 35 - chemoinformatics: What is it and how does it impact drug discovery. volume 33 of *Annual Reports in Medicinal Chemistry*, pages 375 – 384. Academic Press, 1998.

[9] P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, Mar 2009.

[10] M. O. Dayhoff and R. M. Schwartz. Chapter 22: A model of evolutionary change in proteins. In *in Atlas of Protein Sequence and Structure*, 1978.

[11] M. N. Drwal and R. Griffith. Combination of ligand- and structure-based methods in virtual screening. *Drug Discovery Today: Technologies*, 10(3):e395–e401, Sep 2013.

[12] J. Gasteiger. Chemoinformatics: a new field with a long tradition. *Analytical and Bioanalytical Chemistry*, 384(1):57–64, 2006.

[13] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, and J. P. Overington. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, Sep 2011.

[14] A. G. Guy Shani. Evaluating recommender systems. Technical report, Nov 2009.

[15] S. R. Heller, A. McNaught, I. Pletnev, S. Stein, and D. Tchekhovskoi. Inchi, the iupac international chemical identifier. *Journal of Cheminformatics*, 7(1):23, 2015.

[16] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U.S.A.*, 89(22):10915–10919, Nov 1992.

[17] J. Hughes, S. Rees, S. Kalindjian, and K. Philpott. Principles of early drug discovery. *British Journal of Pharmacology*, 162(6):1239–1249, Feb 2011.

[18] W. L. Jorgensen. The many roles of computation in drug discovery. *Science*, 303(5665):1813–1818, 2004.

[19] A. J. Kooistra, H. F. Vischer, D. McNaught-Flores, R. Leurs, I. J. P. de Esch, and C. de Graaf. Function-specific virtual screening for GPCR ligands using a combined scoring method. *Scientific Reports*, 6:28288, Jun 2016.

[20] B. S. P. R. T. Kroemer. Structure-based drug design: Docking and scoring. *Current Protein & Peptide Science*, 8(4):312–328, Aug 2007.

[21] P. S. Kutchukian, N. Y. Vasilyeva, J. Xu, M. K. Lindvall, M. P. Dillon, M. Glick, J. D. Coley, and N. Brooijmans. Inside the mind of a medicinal chemist: The role of human bias in compound prioritization during drug discovery. *PLoS ONE*, 7(11):e48476, Nov 2012.

[22] A. R. Leach and V. Gillet. *An Introduction to Chemoinformatics*. Springer, 2010.

[23] C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Deliv. Rev.*, 46(1-3):3–26, Mar 2001.

[24] G. Maggiora, M. Vogt, D. Stumpfe, and J. Bajorath. Molecular similarity in medicinal chemistry. *Journal of Medicinal Chemistry*, 57(8):3186–3204, 2014. PMID: 24151987.

[25] D. E. Patterson, R. D. Cramer, A. M. Ferguson, R. D. Clark, and L. E. Weinberger. Neighborhood behavior: a useful concept for validation of "molecular diversity" descriptors. *Journal of Medicinal Chemistry*, 39(16):3049–3059, Jan 1996.

[26] RDKit. Open-Source Cheminformatics and Machine Learning, Apr 2016.

[27] T. Schulz-Gasch, C. Schärfer, W. Guba, and M. Rarey. TFD: Torsion fingerprints as a new measure to compare small molecule conformations. *Journal of Chemical Information and Modeling*, 52(6):1499–1512, Jun 2012.

[28] K. Schwaber. *Agile project management with Scrum.* Microsoft press, 2004.

[29] H. Sun. Pharmacophore-based virtual screening. *Current Medicinal Chemistry*, 15(10):1018–1024, Apr 2008.

[30] F. Svensson, A. Karlén, and C. Sköld. Virtual screening data fusion using both structure- and ligand-based methods. *Journal of Chemical Information and Modeling*, 52(1):225–232, 2012.

[31] T. T. Tanimoto. IBM internal report. Technical report, IBM Corporation, Armonk, NY, Nov 1957.

[32] D. Weininger. SMILES, a chemical language and information system. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[33] P. Willett, J. M. Barnard, and G. M. Downs. Chemical similarity searching. *Journal of Chemical Information and Computer Sciences*, 38(6):983–996, Nov 1998.

[34] D. Wilton, P. Willett, K. Lawson, and G. Mullier. Comparison of ranking methods for virtual screening in lead-discovery programs. *Journal of Chemical Information and Computer Sciences*, 43(2):469–474, Mar 2003.

APPENDIX

A. PROJECT PLAN

This research is being carried out on part-time basis over a period of 1 year. The final research will be presented as the dissertation for a masters level in Artificial Intelligence. The methodology framework of choice is Scrum with a two week Sprint and a Velocity of four points, confer [28]. Table 3 outlines the User Stories (main tasks) identified, their respective Scrum Points, and Sprint Number when these tasks are going to be carried out. The purpose of using Scrum is to have a deliverable by the end of each Sprint, thus effort can be measured and the Supervisor and Co-Supervisor are kept up-to-date.

The plan proposed has some contingency to compensate impediments that may hinder the completion of a particular User Story in time. This contingency is added in two forms, (a) all tasks are to be completed a couple of weeks prior the deadline, and (b) some repetitive User Story may take less time as the framework needed will be already implemented in previous ones.

Table 3: Project Plan

Start	End	Sprint Number	Story Number	Story Points	Description
20-06-2016	03-07-2016	1	0	Done	Discussion and research about proposal
04-07-2016	17-07-2016	2	10	Done	Reading about Chemoinformatics, virtual screening, and molecule fingerprinting
			11	Done	Researching and creating a similarity matrix for 100 random molecules (ligands)
18-07-2016	31-07-2016	3	20	Done	Introduction to proteins, Protein Data Bank, and reading of related literature
01-08-2016	14-08-2016	4	21	Done	Introduction to BLAST and exeperimenting with local BLAST
15-08-2016	28-08-2016	5	22	Done	Implementation of protein similarity matrix for random 100 proteins
29-08-2016	11-09-2016	-	-	-	-
12-09-2016	25-09-2016	6	23	Done	Continue protein similarity matrix and uploading data to Neo4j
26-09-2016	09-10-2016	7	30	Done	Introduction to D3
10-10-2016	23-10-2016	8	40	Done	Introduction to ChEMBL and create a D3 graph showing molecule and ligand bindings
24-10-2016	06-11-2016	9	50	Done	Introduction and building a Spark cluster
07-11-2016	20-11-2016	10	51	Done	Researching GraphFrames on Spark
21-11-2016	04-12-2016	11	60	Done	Implement distributed BLAST and Tanimoto similarity using PySpark
05-12-2016	11-12-2016	12	61	Done	Working on progress report
12-12-2016	25-12-2016	13	100	1	Dataset preparation
			200	3	Automate Apache Hadoop and SPARK cluster installation
26-12-2016	08-01-2017	14	300	4	Implementation and documentation of ligands graph creation and analysis framework
09-01-2017	22-01-2017	15	400	4	Design, implement, and document ligands similarity matrices creation
23-01-2017	05-02-2017	16	500	4	Design and setup of web application framework
06-02-2017	19-02-2017	17	501	3	Documentation of web application framework
			600	1	Design search for similar ligands
20-02-2017	05-03-2017	-	-	-	-
06-03-2017	19-03-2017	18	601	4	Implement and document search for similar ligands
20-03-2017	02-04-2017	19	700	4	Design, implement, and document Protein similarity matrices creation
03-04-2017	16-04-2017	20	800	4	Design and implement search for similar proteins
17-04-2017	30-04-2017	21	801	2	Document search for similar proteins
01-05-2017	14-05-2017	22	900	4	Design, implement, and document bipartite graph
15-05-2017	28-05-2017	23	1000	4	Implement highlight of graph walks from query ligand to target protein (Part I)
29-05-2017	11-06-2017	24	1001	1	Implement highlight of graph walks from query ligand to target protein (Part II)
			1002	3	Document highlight of graph walks from query ligand to target protein
12-06-2017	25-06-2017	25	1100	4	Implement highlight of graph walks from query protein to ligand (Part I)
26-06-2017	09-07-2017	26	1101	1	Implement highlight of graph walks from query protein to ligand (Part II)
			1102	1	Document highlight of graph walks from query protein to ligand
10-07-2017	23-07-2017	27	1200	4	Evaluation and analysis
24-07-2017	06-08-2017	28	1201	4	Documentation related to User Story 1200
07-08-2017	20-08-2017	29	1300	4	Review write-up
21-08-2017	03-09-2017	30	1301	4	Finalize write-up