ICS5200 Home

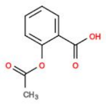
Virtual Screening toolbox

Enter your query data here.

Ligand SMILES: O=C(C)Oc1ccccc1C(=O)O Go!

Protein Sequence: Go!

Ligand



Relative weight: 180.15851999999998 g/mol
Absolute weight: 180.04226 g/mol
Formula: $C_9H_8O_4$
logP: 1.131399977952242
logS: -1.9289999306201935
Polar surface area: 63.599998474121094 Å^2
Checking ChEMBL data...

Use Morgan fingerprints (ECFP4)

Use MACCS fingerprints

Tanimoto similarity coefficient: 0.8

Screen ligands and find possible target proteins | Back to last results

ICS5200 Home

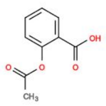
Virtual Screening toolbox

Enter your query data here.

Ligand SMILES: O=C(C)Oc1ccccc1C(=O)O Go!

Protein Sequence: Go!

Ligand



Relative weight: 180.15851999999998 g/mol
Absolute weight: 180.04226 g/mol
Formula: $C_9H_8O_4$
logP: 1.131399977952242
logS: -1.9289999306201935
Polar surface area: 63.599998474121094 Å^2
ChEMBL (molRegNo): 1280
Binds to proteins: P05106, P08514, P35354, P23219

Use Morgan fingerprints (ECFP4)

Use MACCS fingerprints

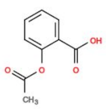
Tanimoto similarity coefficient: 0.8

Screen ligands and find possible target proteins | Back to last results

ICS5200 Home

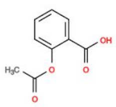
Virtual Screening Results

Query ligand:
ChEMBL (molRegNo): 1280



Unique Protein Accessions found:
(P08514 P05106 P23219 P35354)

Selected ligand:
1280



CC(=O)Oc1ccccc1C(=O)O

Copy | Excel | CSV | PDF

Search:

Mol Reg No	Molecule Pref Name	Similarity	Std Relation	Std Value	Std Units	Std Type	pChEMBL Value	Protein Desc	MolRegNo
1280	ASPIRIN	1	=	5000	nM	IC50	5	Integrin alpha-IIb	1280
1280	ASPIRIN	1	=	5000	nM	IC50	5	Integrin beta-3	1280
1280	ASPIRIN	1	=	80	%	Inhibition		Prostaglandin G/H synthase 1	1280
1280	ASPIRIN	1	=	4192	nM	IC50	5	Prostaglandin G/H synthase 1	1280
1280	ASPIRIN	1	>=	90	%	Inhibition		Prostaglandin G/H synthase 1	1280
1280	ASPIRIN	1	=	2400	nM	IC50	6	Prostaglandin G/H synthase 2	1280
1280	ASPIRIN	1	=	2400	nM	IC50	6	Prostaglandin G/H synthase 2	1280
1280	ASPIRIN	1	=	2400	nM	IC50	6	Prostaglandin G/H synthase 2	1280
1280	ASPIRIN	1	=	2400	nM	IC50	6	Prostaglandin G/H synthase 2	1280
1280	ASPIRIN	1	=	2400	nM	IC50	6	Prostaglandin G/H synthase 2	1280

Showing 1 to 10 of 12 entries

Previous 1 2 Next

Date: 19 June 2017 (meeting 22)

Meeting postponed from 12 June to 19 June.

Experiment results:

1. Leave-one-out (find proteins from ligand similarity) (b04)

Hide all bindings for one ligand and try to find them using similar ligands.

Tanimoto Recall

Tc	Morgan	MACCS
0.6	0.914	0.991
0.7	0.870	0.982
0.8	0.764	0.966
0.9	0.614	0.929

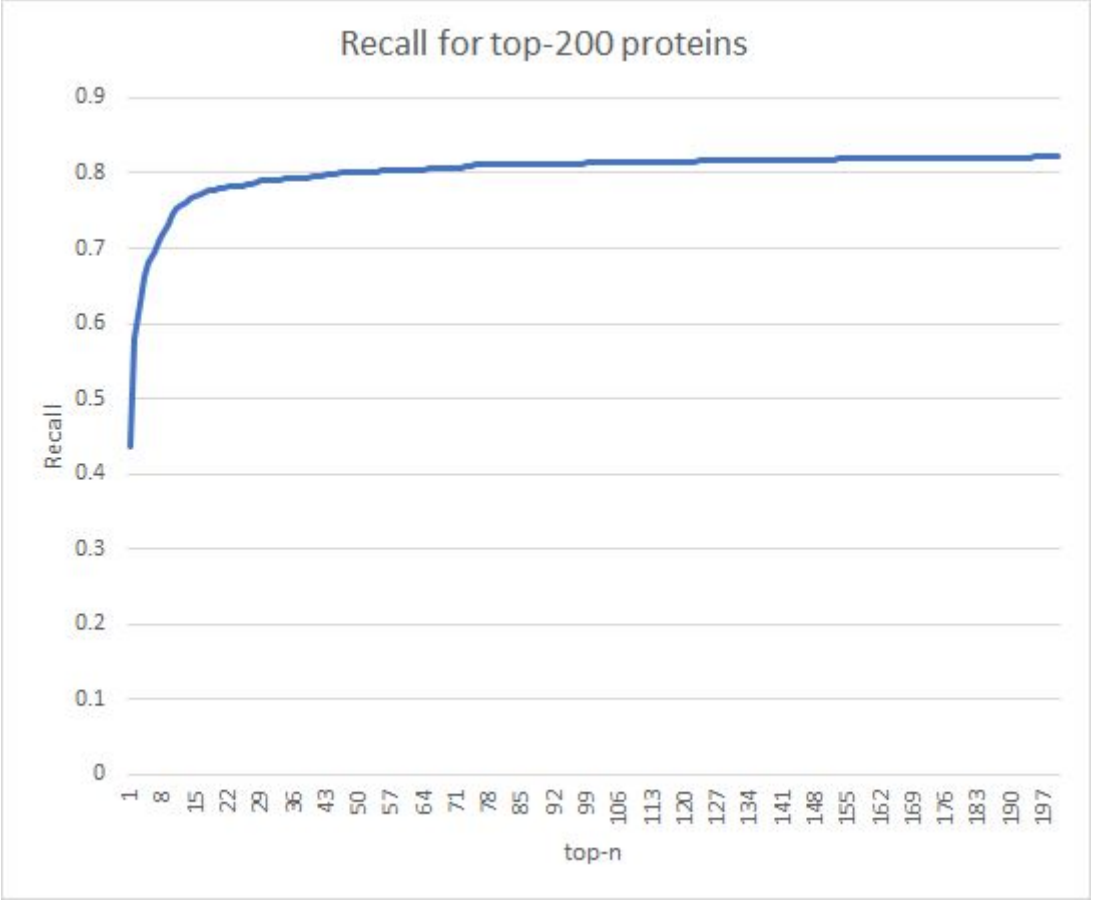
Dice Recall

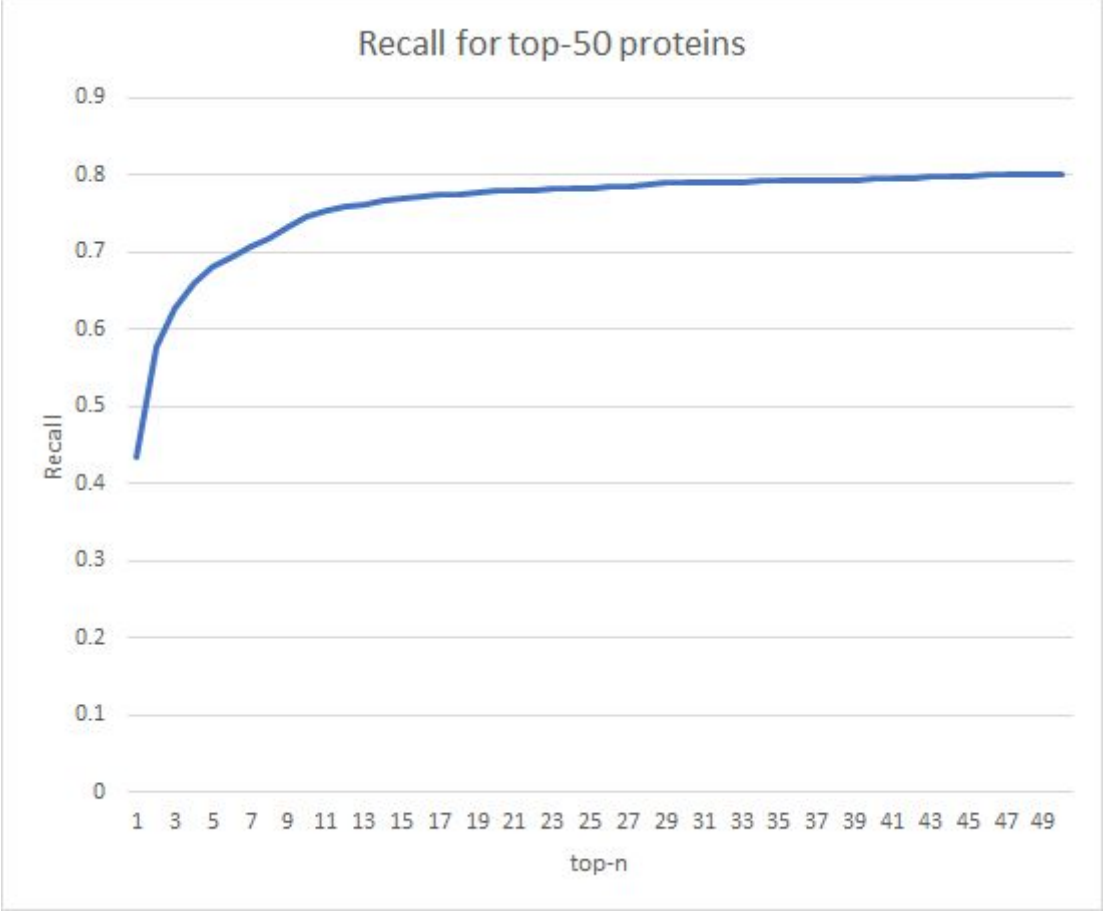
Dice	Morgan	MACCS
0.6	0.958	0.998
0.7	0.947	0.997
0.8	0.921	0.992
0.9	0.807	0.974

2. Protein *n*-top (b00)

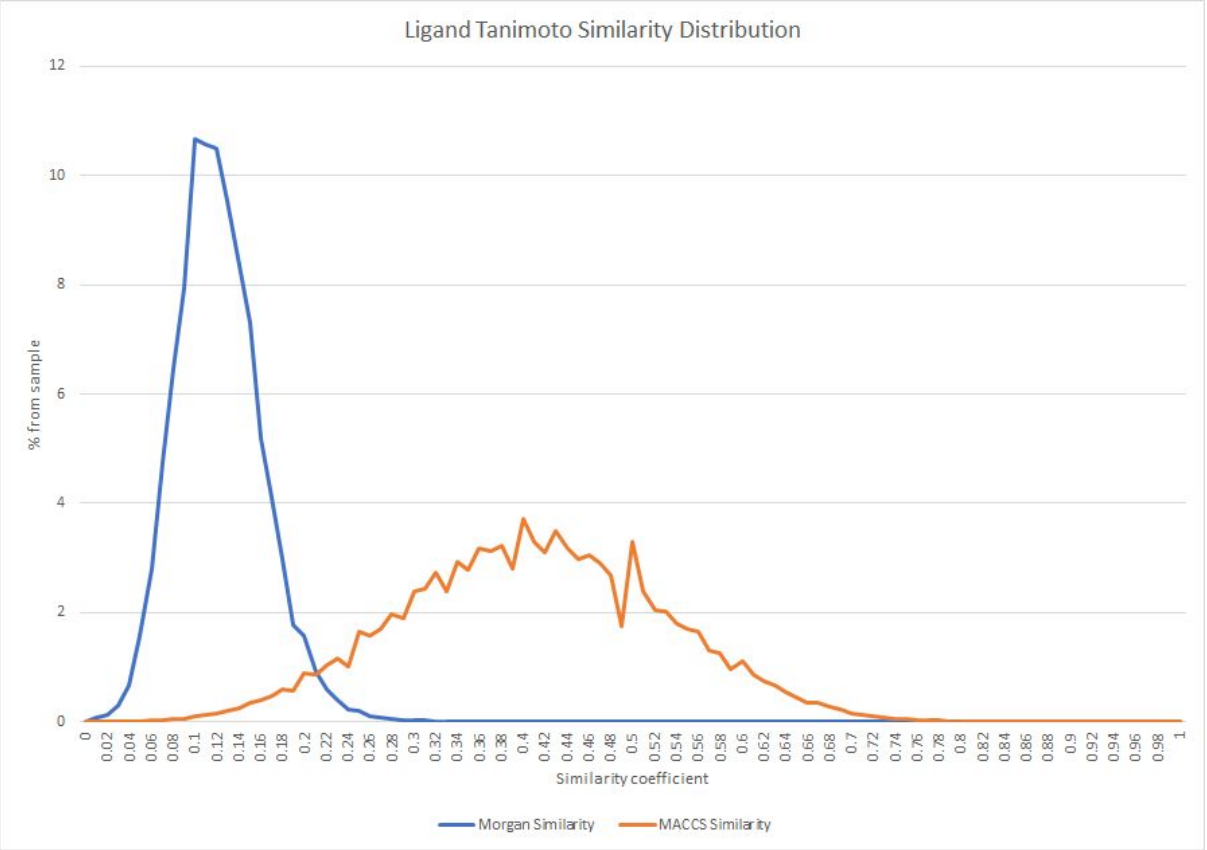
Hide 20% of the bindings and try to unhide them using top-n similar proteins. (experiment 1 from last session)

Bindings: 334236
Proteins: 1875
Ligands: 210090


Recall for top-200 proteins
Recall
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1 8 15 22 29 36 43 50 57 64 71 78 85 92 99 106 113 120 127 134 141 148 155 162 169 176 183 190 197
top-n


Recall for top-50 proteins
Recall
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
top-n

3. Ligand similarity distribution (b02)



Date: 29 May 2017 (meeting 21)
Meeting started at 3pm.

We started the meeting by going through the tasks completed by JD during the last two weeks. JD pointed out that the values obtained last time for experiment 1 are incorrect. During code review, JD discovered that he was not omitting recall and precision for those ligands for which there are no proteins in the known set (see note at the end of the method section of experiment 1 in the notes – link above). Thus the final average recall and precision scores are skewed.

The new results are as follows:

> Total hidden set (bindings that we hid) = 31636 (after removing the ones for which we do not have the same ligand in the known set, thus we cannot find known proteins)
>
> Results using Top 10 proteins from the predicted target proteins:
> • We discovered 93% of the 1533 unique hidden proteins (e-value = 0.04)
> • We discovered 82% of the 1533 unique hidden proteins (e-value = 0.0005)
> • Recall:
> o 0.746 (e-value = 0.04)

    - o 0.745 (e-value = 0.0005)
- Precision:
    - o 0.754 (e-value = 0.04)
    - o 0.753 (e-value = 0.0005)

Next we discussed the ligands similarity matrix. Finally, a half similarity matrix of 210064 ligands was computed and stored to disk. It tallies 88Gb of float numbers.

The next experiment discussed was about running experiment 2 from last meeting with varying Tanimoto Coefficient. The results are summarised in the table below.

Tc	**0.7**	**0.8**	**0.9**
Recall	0.788	0.504	0.151
Precision	0.791	0.506	0.152

It is interesting to note the steep gradient of score values.

JD pointed out the difference in using Morgan fingerprints (as in the previous experiment) and MACCS. For MACCS, when Tc = 0.7, recall and precision scores reads as 0.905 and 0.918 respectively.

We discussed that given the element of False Positives, we need to take precision with a pinch of salt, thus it is best to do some experiments using recall only to evaluate how well our system can uncover known ligand-protein bindings.

Until next meeting, JD is to work on:

- Run experiment 2 from last session using leave-one-out, and measure the recall.
- Run experiment 1 from last session and plot Recall-*n* graph.
- We also discussed another type of graph to produce, by having Tc on the y-axis and n on the x-axis.

Next meeting is set for 12 June at 4pm.

Date : 17 May 2017 (meeting 20)
Meeting started at 4pm, MSc hot-desk area (present for meeting, JP and JD)

JD presented the results obtained from the experiments that were agreed during the last meeting. Here is a summary:

Experiment 1:
Aim: System evaluation by using known target proteins to uncover other known target proteins for a given ligand that are purposely hidden.

Dataset : ChEMBL ligand-protein bindings (HomoSapiens and high affinity - as discussed in previous meetings)

Setup: Split dataset in two in the ratio 4:1. 80% is used referred as the known bindings set and 20% are hidden (referred as the hidden set).

Number of bindings: 334236
Number of unique proteins: 1875
Number of unique ligands: 210090
Number of known set bindings: 267388
Number of hidden set bindings: 66848

Number of unique ligands in hidden set: 61876

Method:

1. Let query ligand be a ligand that we hid bindings from (that is, a ligand from the hidden set)
2. Find the proteins that we know that a it binds to (using the known bindings set); call this the known protein binding set
3. Given the set of known proteins, find similar proteins from the full proteins set and remove any similar proteins that are in the known protein binding set.
4. Rank the similar proteins and take top *n*; call this the predicted protein set
5. Calculate recall and precision using the formulae presented during the last meeting.

Execute the above steps for all ligands in the hidden set and record recall and precision for each. Remove any scores for which there are no known proteins for a ligand. The latter case arise when in our dataset we only have one binding for a given ligand and this ligand is in the hidden set. Finally compute the average of all recall scores and the average of all precision scores.

Results:

	Prediction Top 1	Prediction Top 10	Average
recall	0.22	0.38	0.34
precision	0.25	0.39	0.35

Results are not promising, further investigation is required.

Experiment 2:
Aim: System evaluation by using similar ligands to a query ligand to uncover known target proteins that are purposely hidden.

Dataset : ChEMBL ligand-protein bindings (HomoSapiens and high affinity - as discussed in previous meetings)

Setup: Split dataset in two in the ratio 4:1. 80% is used referred as the known bindings set and 20% are hidden (referred as the hidden set).

Number of bindings: 334236
Number of unique proteins: 1875
Number of unique ligands: 210090
Number of known set bindings: 267388
Number of hidden set bindings: 66848

Number of unique ligands in hidden set: 61876

Method:

1. Let query ligand be a ligand that we hid bindings from (that is, a ligand from the hidden set).
2. Using Tanimoto Similarity and coefficient Tc, find all similar ligands in our full ligands set.
3. Get all proteins that we know that are targets for the ligands resulted in step 2 (using the known binding set).
4. Remove the proteins that we know they are targets for our query ligand (via the known binding set); call this the predicted protein set.
5. Calculate recall and precision using the formulae presented during the last meeting.

Execute the above steps for all ligands in the hidden set and record recall and precision for each. Remove any scores for which there are no known proteins for a ligand. The latter case arise when in our dataset we only have one binding for a given ligand and this ligand is in the hidden set. Finally compute the average of all recall scores and the average of all precision scores.

Results:

Using 1024-bit Morgan fingerprint and a Tanimoto coefficient of 0.7, the results over 61876 hidden ligands is:

Recall: 0.788
Precision: 0.791

Other:
JP and JD discussed the above experiments and set the following objectives until the next meeting:

- Verify the correctness of experiment 1.
- Compute the percentage of uncovered proteins in experiment 1 from the total hidden set.
- Run experiment 2 using different Tc coefficients
- Continue working on building a Tanimoto similarity half matrix for all ligands in our ChEMBL subset.
- Get the time required to compute 100,000 similarities on Spark.

Next meeting: Monday 29 at 4pm.

Date : 4 May 2017 (meeting 19)
Meeting started at 4pm, JP's office (CA sick)

JD presented his research how to evaluate the ligand/protein matching system proposed in this study to JP, discussed it, and agreed to implement it by next meeting. Here is a summary of the evaluation strategy.

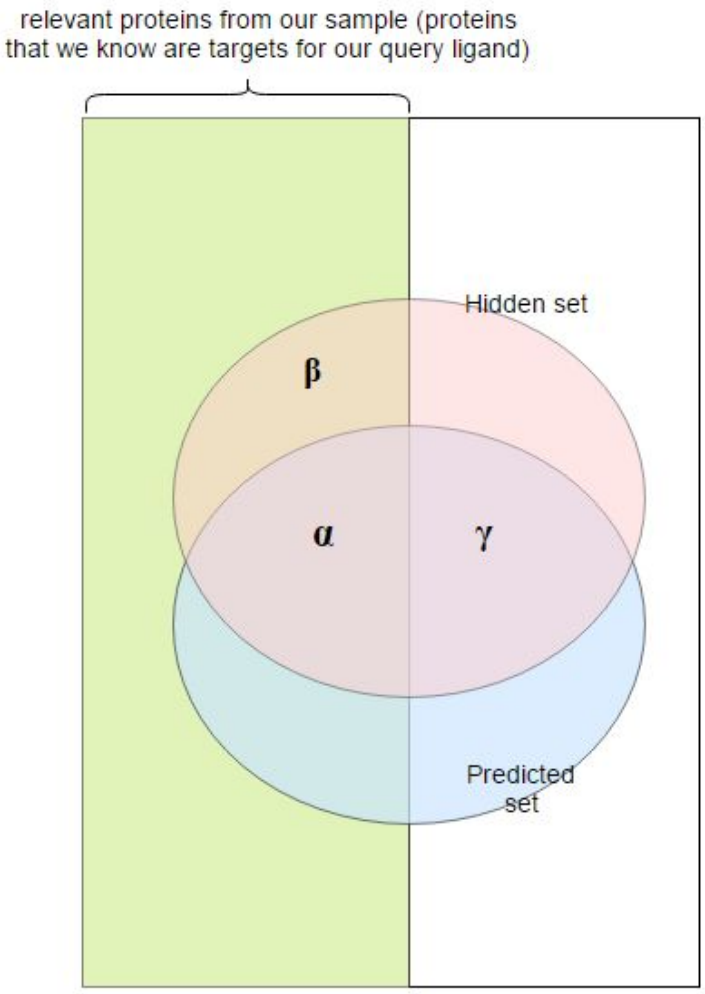
JD suggested an evaluation approach based on collaborative filtering recommender system and inspired by the paper "A New Cross-Validation Technique to Evaluate Quality of Recommender Systems" (https://link.springer.com/chapter/10.1007/978-3-642-27387-2_25). We are mostly interested in recommended some good items (ligands or proteins depending on what we are searching for), thus we prefer a system with high precision. The below describes the procedure to find new proteins that can be targets for our query ligand. Let the sparse matrix B be the matrix of all bindings between ligands and proteins (see figure below):

	Protein 1	Protein 2	...	Protein n
Ligand 1	b1	b2	...	
Ligand 2		b3	...	b4
...	b5		...	b6
Ligand m	b6		...	

Next we will hide 20% of our proteins (20% of the columns in the figure above).

Then, the result obtain by running the recommending algorithm to find putative protein target for our ligand can be depicted as the Venn Diagram below.

The predicted set is the set of proteins returned by our algorithm, the hidden set is the set of proteins that we hid at the start, and the green section is the set of all proteins from our sample that we know that bind to the query ligand.



Thus:

$$Recall = \frac{|PredictedProteins \cap RelevantProteins \cap HiddenProteins|}{|RerlevantProteins \cap HiddenProteins|} = \frac{|\alpha|}{|\alpha \cap \beta|}$$

$$Precision = \frac{|PredictedProteins \cap RelevantProteins \cap HiddenProteins|}{|PredictedProteins \cap HiddenProteins|} = \frac{|\alpha|}{|\alpha \cap \gamma|}$$

$$F1\text{-}measure = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

We can perform various experiments and compare the results using a Precision-Recall plot. The threshold will be the number of recommended proteins returned by the search algorithm.

JP and JD discussed that after this, JD shall look into the Subset algorithm to create ligand clusters.

Next meeting is on Wed 17 May at 4pm.

Date : 28 March 2017 (meeting 18)
Meeting started at 4pm in Charlie's office (JP sick)

Started meeting by discussing the difficulties encountered to generate the matrices on Azure HDI cluster. CA asked if there is the possibility to know a priori if some ligands generate a 0 or 1 similarity. In that case, we can skip them and perhaps we gain performance.
@JP: is there something that I can use to determine this, perhaps some ligand feature? I do not know if this gives us performance, as we still need to match one ligand to all other ligands in the set.
CA questioned if there are other papers that we can cite that did something similar. JD answered that he tried to find, but there are only a couple and do not give details, like Simap.

During the meeting we discussed the conference paper and the difficulties encountered. JD mentioned that the main issue is that he do not has an evaluation in place and he need to build this up from start. All he has is the protein matrix with a rough idea of the time taken to generate. He feels that this is not enough for a conference paper and more work need to be done. Of course, this work is also valid for the final master's report.

CA and JD agreed that we need to have a valid evaluation and solid methodology for the conference paper and agreed to continue working on the paper, perhaps not for the cibcb2017.
@JP do you agree with this decision? I feel that I need to work more to have something solid for cibcb so that I do not spoil it.
[JP replied to email that Quality must come first]

CA and JD discussed the cross-validation proposed in the paper "cross-validation for..." at https://link.springer.com/chapter/10.1007/978-3-642-27387-2_25. The idea revolves around the idea we mentioned once to hide part of the known bindings and evaluate how much the system find from these bindings. However, the paper is about movie recommendation not bindings.

CA asked JD to provide some metrics about the cluster requirement to generate the full ChEMBL ligands matrix as until now, JD only managed to generate 80,000^2 matrix.

Next meeting for two weeks' time, that is, 11 April at 4pm.

Date: 14 March 2017 (meeting 17)
Meeting at 4pm at Charlie's office

JD presented the data obtained for creating the Protein's similarity matrix:

- 156,850 Homo Sapiens fasta sequences
- Took about 5hr 30min to create a full matrix using BLASTP
- Created 45,332,538 matches using e-value of 10
- The number of matches is expected to be very very small when compared with the total number of possible combinations because proteins in the human body have different functions, thus different sequences.

CA suggested that JD shall document the design and Dr Shoemake's use case (even though we do not have any feedback yet).

During the meeting, we discussed if it makes sense to cluster the protein matrix and mentioned different cluster approaches. Of particular interest, CA mentioned density clustering and label propagation algorithm. Clustering can be an optimisation of what I did for Dr Shoemake to get a list of proteins. JP mentioned an algorithm named Subset, with O(n), which may be good for big data scenarios.

Next meeting is set for 28 March at 4pm.

Date: 2 March 2017 (meeting 16)
Meeting at 4:00pm

JD demonstrated the work done to suggest putative protein bindings to Dr Shoemake's ligand.

JD also started to work on creating a full protein similarity matrix.

JP asked JD to send the list of similar SMILES (those similar to Dr Shoemake's) together with a list of protein description.

For next meeting, JD shall continue working on the proteins similarity matrix, search on cross-validation, and recommendation system confidence level.

Next meeting is set for PI day at 4pm.

Date: 14 February 2017
Meeting with Dr Claire Shoemake from the Faculty of Medicine and Surgery at 9:00am at JP's office. For this meeting, JP, CA, Dr Shoemake, and myself were present. It lasted about 30 minutes.

The meeting is about a use case where Dr Shoemake has a ligand and would like to know binding proteins.

Dr Shoemake is going to send us a ligand in SMILES format so that we can find similar ligands in ChEMBL and using these similar ligands we can suggest proteins that this new ligand is probable to bind with.

Date: 7 February 2017 (meeting 15)
Meeting at 4:00pm

During today's meeting we discussed the challenges in the evaluation process for this project. The main blocker is that a "recommended" ligand/protein binding by the project is not necessarily in ChEMBL and thus we cannot prove that this recommendation is correct. Ultimately, we need to have such recommendations, as this is the purpose of the whole project. However, this can only be verified in a wet lab.

CA suggested to look if there are other domains with such evaluation issues and see how it is tackled.

We mentioned that JD shall try to think on the lines how we can hide the known connections and ensure that the system discovers them. At least, a good percentage (which must be measured and evaluated) must be discovered by the system.

JP mentioned that although it is difficult to evaluate and get proves for the suggestions, it must be a scientific process and sound.

CA suggested that we can look at the confidence level of the project's suggested binding. Given, a high confidentiality, we can later take the next step (not in the time and budget allocated for this project) to suggest a binding to be tried in a wet lab.

JP asked JD to work on the protein similarity matrix and get some values of how much time this takes to complete. We shall see how feasible is this. This is similar to SIMAP, but the latter is out-dated and the software architecture is not publicly available. If the full matrix proves to be unfeasible, we may try to cluster proteins on similarity to limit our set.

JP is to send binding filters to JD so that the latter can clean the data based on these filters.

Next meeting: Thursday 2 March at 4:00pm

Date: 27 January 2017 (meeting 14)
Meeting at 4:00pm

JD met JP, as CA was sick. JD showed the progress done and demonstrated experiments to find putative bindings given an unseen ligand or given an unseen protein.

JD and JP reviewed the evaluation process and JD had to continue to work on it.

For the next meeting:

- Do ligands experiment given an unseen SMILES, rather than using the 1% from validation set.
- Start implementing graphs
- Work on evaluation plan

Next meeting: 7 February 2017 at 4:00pm.

Date: 10 January 2017 (meeting 13)
Meeting at 4:00pm

Since last meeting, JD focused on building web UI that given a ligand, the program finds similar ligands, gets their known binding, and suggests these bindings as putative binding for the given ligand. For this prototype, I take 100,000 bindings from ChEMBL. 1% of this sample is taken as unseen bindings, while the remaining 99% are used to suggest putative bindings.

For the next meeting, JD has to do the same for Proteins.

Next meeting: Tue 24 January 2017

Date: 20 December 2016
Postponed: JD sick

Date: 12 December 2016 (meeting 12)
JD met JP at 3pm to go through the progress report.

Date: 6 December 2016 (meeting 11)
Meeting at 4:00pm

Before this meeting, JD sent the first progress report draft. JP and CA provided some points about the write-up:

- Aims and Objectives shall be in the form of bullet points, so that the reader can immediately pin-point them.
- Use different symbols for proteins and ligands (in diagrams)
- Make clear the novel parts
- Ensure to show the challenges

During the meeting JP, CA, and JD discussed the challenges for evaluation. We agreed that there is no need to go into full details for the progress report. But we definitely need to discuss how we are going to evaluate the project. The questions are:

- Are we going to take the value of different similarity algorithms separately, or we are going to combine them somehow?
- How shall evaluation be done, if we cannot verify any recommendation that the system propose, if this is not in ChEMBL?

Until next time, JD should have finished the progress report and research into:

- Virtual Screening machine learning features
    - I carried out some research about this and skimmed a couple of papers. My understanding is that when ML techniques are used, the research targets a couple of proteins or more - normally attributes to a particular disease. Further, there are medicinal chemists that complain that these techniques do not give them visibility to why their tool predict a binding.
- Virtual Screening consensus ranking.
    - I researched this and read this paper (https://www.ncbi.nlm.nih.gov/pubmed/16045308). There's also a paper, Combination of Fingerprint-based Similarity Coefficients Using Data Fusion (http://pubs.acs.org/doi/abs/10.1021/ci025596j) which I need to read in detail. The first conclusion I am deducing from these papers and others that I went through their abstract and conclusion only, I can say that there is no silver bullet combination, but medical chemists need to experiment with multiple similarity coefficients depending on the target protein. An interesting observation from one of the papers (I need to find this again as I forgot to take note of it) is that ranking by summation of the various similarity coefficients gave a quite good binding prediction.

Next meeting 20 December 2016 @ 9:30am

Date: 15 November 2016 (meeting 10)
Meeting at 4:00pm

Given that chemical data cannot be manipulated and computed in Neo4j, JD researched other alternatives. Hadoop + Spark proved to be a good alternative. Why not MapReduce, but Spark? Spark is more flexible and is easier to integrate with other systems. Later, I can use dataframes and graphframes; structures that are integrated in Spark 2.0. Unfortunately, this was not an easy ride. I encountered a lot of difficulties. The "read-made" VMs from different BigData organisations, such as Cloudera and mapR, are standalone installations and are modified so that they promote their infrastructures. Thus, I had to go the long way and install my own 3-node cluster from ground up. This took me some days to complete, but managed to create something that I can scale out and use as I need to.

Having the cluster ready, next was time to implement a UI prototype.

JP and CA asked for an executive summary. This shall be no longer than 1 page and shall include a summary and architecture of what I am trying to achieve. They also emphasised, that JD shall work on the progress report, given that the date is getting closer.

Next meeting is scheduled for 6 December at 4:00pm.

Date: 1 November 2016 (meeting 9)
Meeting at 4:00pm

Since last meeting, JD continued to study the ChEMBL schema and extracted binding relationships. Sample data was uploaded to Neo4j to get better understanding of ligand/protein relationship. Some statistics:

- Bindings in ChEMBL: 2,930,127; of which, some need to be filtered out due to no binding affinity data
- Molecules: 710767
- Targets: 2888

JD also carried research to draw molecule structure from SMILES. Unfortunately, there is no javascript library that does so. However, there are libraries, such as chemweb doodle, that can draw the molecule from mol file. This data is also available in ChEMBL.

Until next meeting:

- Work on progress report
- Research cross-validation techniques, such as jack-knifing and bootstrapping
- Look into better data visualization

Next meeting: Tuesday 15 November @ 4pm.

Date: 18 October 2016 (meeting 8)
Meeting at 4:00pm

During the last 2 weeks, JD started an introduction to D3.js. This proved to be harder than sigma.js. The latter had libraries to read data from NEO4j whereas in D3 there is none and one has to use REST API. Further, in D3.js there is an issue that backward compatibility is something much desired. One of the biggest issues is that the API changes from one version to the other. For example, in D3 v3 there is a namespace and functions for layout whereas in v4 this is replaced by something else. Thus, one need to pay attention for the version being used. Charlie suggested that we should pick one version and stick to it. Ideally we download the js files and reference them locally rather than using the ones available online. This safeguards us from any mishaps in the future in case that the files are not available anymore.

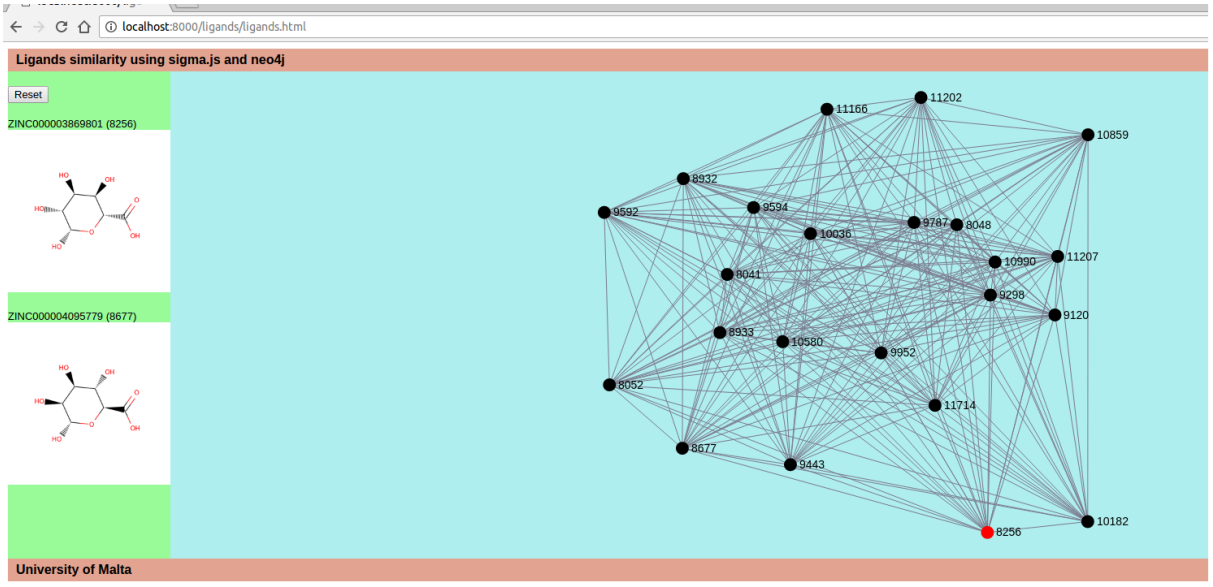
Further to D3 experimentation, JD installed ChEMBL MySQL database and myChEMBL VM. Started to look into the schema and also went through some Python scripts available with the VM.

Until the next meeting, Tuesday 1 November, JD is to continue learning about D3js, create a SQL query to display the relationship between Protein and Ligands in ChEMBL, and start drafting the "Progress Report" required to be handed in in December. This latter report has 10% of the final mark.

Next meeting: Tuesday 1 November @4pm.

Date: 3 October 2016 (meeting 7)
Meeting at 8:15am (this was not a good idea - a lot of traffic and we started at 9:00am)

Joseph demonstrated the work done so far. A sample screen shot is shown below. Code will be uploaded to github at http://www.github.com/jod75 shortly.



Until the next meeting, JD is to work on the following tasks:

- Try D3.js and do the same task above using D3.js
- Familiarise with Chembl - to link ligands with protein
- Look into jung (java library) - for graph analysis
- Create software design to support similarity layers (visualisation) and different similarity algorithms

Next meeting is to be held on Tuesday 18 October @ 4pm.

Date: 19 September 2016 (meeting 6)
Meeting held at JP's office at 4:00pm

JD demonstrated the prototype work done so far. This can be summarised as follows:

1. Download 100 random homo sapiens proteins and store them as a FASTA file
2. Create a local BLAST database for the downloaded proteins in (1)
3. Run BLAST query (use FASTA file from (1) and db in (2))
4. Create e-value matrix from the result obtained in (3)
5. Plot a directed weighted graph (in neo4j) for the matrix created in (4)

It has been observed that in some cases there is an alignment from seq1 to seq2 but not from seq2 to seq1. This is mainly due to the fact that the scoring function property considers the distribution of amino acids in the query. When the query and the subject are exchanged, the amino acid distribution is changed, thus a different score value is obtained. (citation needed, see also http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html)
Further the e-value is based on the database size, thus one can get different e-value for the same sequence alignment when this is queried against different databases.
(https://blast.ncbi.nlm.nih.gov/blast/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=FAQ)

Task to be carried out until next meeting:

1. Create similarity matrix on ligands and plot graph

Sample code and result:
1. Download 100 sample sequences (use http://www.uniprot.org/uniprot/?query=taxonomy%3A%22Homo+sapiens+%28Human%29+%5B9606%5D%22+AND+reviewed%3Ayes&sort=score website and save 100 homosapiense protein sequences in fasts file)

2. Create a local blast (Basic Local Alignment Search Tool) db:

```
$makeblastdb -in random100humanproteins.fasta -parse_seqids -dbtype prot
```

3. Create a python script to run blast queries across all sequences downloaded in (1) against the blast db created in (2)

```
# 0. localblast.ipynb
# This program takes a fasta file and finds sequence similarities from a
local database
```

```
#
from Bio.Blast.Applications import NcbiblastxCommandline

blast_in = "/home/joseph/Projects/Labs/homosapiens100Random.fasta" # all
set
db = "/home/joseph/Projects/Labs/homosapiens100Random.fasta"
blast_out = "/home/joseph/Projects/Labs/test.xml"

# prepare the blastx command
# note that we have to set the cmd = 'blastp' to run the protein
counterpart instead of nucleoid
blastx_cline = NcbiblastxCommandline(cmd='blastp',
                                     query=blast_in,
                                     db=db,
                                     evalue=0.05,
                                     outfmt=5,
                                     out=blast_out)


print(blastx_cline)
stdout, stderr = blastx_cline()
print(stdout)
print(stderr)
```

5. Parse XML using BioPython and create score (e-value) matrix. Code:

```
# 1.readBlastOutput
from Bio.Blast import NCBIXML
from Bio import SeqIO
import numpy

blast_out = "/home/joseph/Projects/Labs/test.xml"
fasta_file = "/home/joseph/Projects/Labs/homosapiens100Random.fasta"

result_handle = open(blast_out)

# use NCBIXML.read if there is only one record, otherwise use .parse()
# blast_record = NCBIXML.read(result_handle)

blast_records = NCBIXML.parse(result_handle)

# create matrix from scores

# rewind iterator
result_handle = open(blast_out)
blast_records = NCBIXML.parse(result_handle)
```

```
# keep a protein list as an index for matrix, and create a matrix of
num_proteins x num_proteins
# make matrix as 1 matrix, that is, no match

# not the best way to do this!
protein_index = []
fasta_records = list(SeqIO.parse(fasta_file, "fasta"))
for record in fasta_records:
    protein_index.append(record.id.split('|')[1])

num_proteins = len(protein_index)
score_matrix = numpy.ones((num_proteins, num_proteins))
print "There are {0} proteins in fasta file (used to build blast
output)".format(num_proteins)


def getProteinIndex(protein_id):
      "Returns the index of the given protein"
      pos = 0
      if protein_id in protein_index:
      pos = protein_index.index(protein_id)
      else:
      pos = len(protein_index)
        protein_index.append(protein_id)
      return pos

# iterate records
for blast_record in blast_records:
    print('-------------------------------------')
      # get second element from query - this contains the protein number
      query_protein_id = blast_record.query.split('|')[1]
      query_protein_index = getProteinIndex(query_protein_id)
      print(query_protein_id)

      print "num hits: {0}".format(blast_record.query_id)
      for alignment in blast_record.alignments:
        print(alignment.accession)
      hits = len(alignment.hsps)
      evalue = 1 # no match
      if hits > 1:
            # there may be multiple high-score pairs, for this exercise we
will take the average score
            # compute average score
            total_evalue = 0
            for hsp in alignment.hsps:
```

```
                  total_evalue = total_evalue + hsp.expect
            evalue = total_evalue / hits
            print " has {0} hits with an average score of
{1}".format(hits, evalue)

      else:
            evalue = alignment.hsps[0].expect
            print " has 1 hit of score {0}".format(evalue)

      # add score to score matrix, use query_protein_id as column
        score_matrix[getProteinIndex(alignment.accession),
query_protein_index] = evalue

print(score_matrix)
numpy.savetxt("/home/joseph/Projects/Labs/score_matrix.csv", score_matrix,
delimiter=",")
```

6. Create neo4j graph:

```
# 3.neo4jblast.ipybn

from py2neo import Graph, Node, Relationship
from Bio import SeqIO
import numpy

# initialise variables
fasta_file = '/home/joseph/Projects/Labs/homosapiens100Random.fasta'
score_matrix_file = '/home/joseph/Projects/Labs/score_matrix.csv'

graph = Graph()
graph.delete_all()

# load matrix file
score_matrix = numpy.loadtxt(score_matrix_file, delimiter=',')


# read fasta file and get proteins (nodes)
protein_index = []
fasta_records = list(SeqIO.parse(fasta_file, 'fasta'))
for record in fasta_records:
    protein_index.append(record.id.split('|')[1])


# create graph and nodes
protein_nodes = []
```

```
for protein in protein_index:
	node = Node('Protein', name=protein)
	protein_nodes.append(node)
	graph.create(node)


# get node from graph by protein name
def getNode(protein_id):
	"Returns the Node for the given protein"
	query = "MATCH (n:Protein {name:'" + protein_id + "'}) RETURN n"
	nodes = graph.run(query)
	return nodes.next()

# create a relation for each entry in matrix that in <1
for col in range(0, len(protein_index) - 1):
	for row in range(0, len(protein_index) - 1):
	# skip diagonal (self relationship)
	if (col != row) and (score_matrix[row, col] < 1):
		#node1 = getNode(protein_index[row])
		#node2 = getNode(protein_index[col])
		node1 = protein_nodes[col]
		node2 = protein_nodes[row]
		graph.create(Relationship(node1, ("match", {'evalue':
score_matrix[row, col]}), node2))
```

Sample graph:

Date: 12 August 2016 (meeting 5)
Meeting held at Charlie's office at 9am

During the meeting JD summarised the work done so far and did a small demo to CA. The work done so far can be summaries as follows:

- Installed BioPython (http://www.biopython.org)
- Ran BLASTP queries to find sequence alignment
    i. Using Internet BLAST database
    ii. Using a local custom BLAST database made up of 100 random human proteins
- Read and traverse BLAST result and interpret e-value

JD to continue working on the results obtained from BLAST alignment results to create e-value half matrix and edge weighted network.

Next meeting: tentatively August 22 or 29 morning (after 11am), or 12 Sept

Date: 1 August 2016 (meeting 4)
Meeting held at Charlie's office at 5pm

JD rounded up his work during the past 2 weeks. He mentioned the problems encountered with PyMol. Once a pdb file was loaded and rendered, the application crashed with error that it cannot find __init__. After several debugging, it turned out that it the Ubuntu VM required to have a considerable amount of video RAM (128MB). JD read the SIMAP paper and discussed that there is no infrastructure technical information/detail in it. JP and CA suggested to check about distributed BLAST (if it exists).

During our meeting, we also discussed Big Data resources, as the offline protein data bank is about 500GB. We need to be able to crunch this data using proper Big Data infrastructures. Charlie is taking care of this aspect, to try to set up University infrastructure or a cloud based solution.

Until next meeting, JD is to:

- Research about distributed BLAST
- Create half matrix of 100 random sequence alignments from the protein data bank
- Represent the half matrix created before into an edge weighted network (possibly using Big Data structures)

Next meeting is to be held on Friday 12 Aug at 9am (Charlie's office)

Date: 18 July 2016 (meeting 3)
Meeting held at JP's office at 6pm

JP explained briefly the notion of R/S stereochemistry. This is not part of the Morgan's Fingerprint sequence and that is why in my previous experiment (http://www.clounce.com/chemoinformatics/tanimoto-molecular-similarity-experiment) ZINC000000895032 scored 1, that is, similar to ZINC000000895128.

Last time, we introduced small molecules and the next step is to introduce macro molecules, proteins.
JP introduced proteins and how small molecules bind to them. He introduced PyMol, dud.e (http://dude.docking.org/), and Blastp.

Actions until next meeting:

JD to:

1. Finish off reading the perspective paper on Molecular Similarity in Medicinal Chemistry.
2. Familiarise with PDB (Protein databank) - download protein sequence and parse
3. Download and Install PyMol
4. Read Similarity Matrix of Proteins (SIMAP): http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1347468/pdf/gkj106.pdf
5. Install and familiarise with Blastp

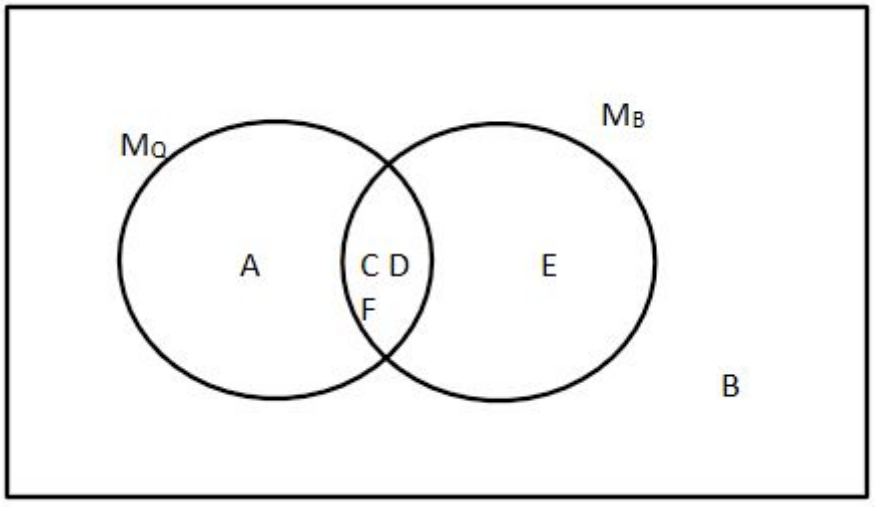
Next meeting to be held on 1 August @5pm.

Date: 4 July 2016 (meeting 2)
JP explained virtual screening and fingerprint concept. A fingerprint is a series of bits, where 1 signifies the presence of a feature while 0 its absence. For example, the binary number 1001 may mean that the particular molecular fingerprint has a positive charge (1XXX), no nitrogen atoms (X0XX), no feature C (XX0X), and feature D (XXX1).

In virtual screening, one of our aims is to find similar molecules from a database that are similar to our molecule query (identified by a fingerprint). For example, the maccs fingerprint can be up to 166 bits in length. There are various ways to find similar molecules, e.g. using cosine similarity or Jaccard.

$$Jaccard\ similarity = \frac{|A \cap B|}{|A \cup B|}$$

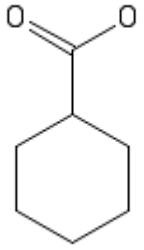
Let M_Q=101101 (ABCDEF), M_A= 000000, M_B = 001111, then the Jaccard Similarity = ⅔.



JP mentioned that the above is a simple way of finding similar molecules, but it has some disadvantages and it is important to choose the correct fingerprint features. For example, if a particular feature is present in almost all molecules, then the feature is not discriminant and almost ineffective. The same can be said if a feature is not present in many of the molecule dataset. For such, other similarity algorithms exist, such as the Morgan fingerprint that takes neighboring features into consideration.

Charlie mentioned that there are Graph algorithms that are suitable to discover neighbor nodes in incremental fashions. We may find this useful for network similarity, later on.

JP mentioned the notion of SMILES, a data structure used represent molecules. For example, OC(=O)C1CCCCC1 represents:



Where O is an Oxygen atom, C is a Carbon atom, = is double bond, () means a branch, and 1..1 is a loop. The same molecule can be represented as C7O2 but the latter does not give you the bonding information.

Actions for next meeting:

JD to:

1. Read the perspective paper, Molecular Similarity
2. Write a paragraph about virtual screening
3. Download and install RDKit, select a molecule from ZINC database and create its Morgan fingerprint, find the top 20 matches using tanimoto (Jaccard index) and draw them out.
4. Upload all source in github
5. Keep journal in Blog

Next meeting to be held on 18 July @5pm.

Date: 20 June 2016 (meeting 1)
Present: Charlie Abela, JP Ebejer, Joseph D'Emanuele

Discussion Items:

- Proposal
- Title: Big data approaches for the discovery of novel medical molecules
- Plan for Literature Review
- Filled in Dissertation Title and Supervision sheet (to be submitted by 23 June)

JD to look into:

- *Introduction to ChemoInformatics*, by Andrew Leach
- RDKit (software package)
- Virtual screening (to provide summary by next meeting)
- Send Google Calendar invite for meetings (agreed to keep Monday 5pm and meet every 2 weeks) JD will take minutes and share them

JP is to:

- Write dissertation proposal and share it (to be submitted by 23 June)
- Share LaTeX template

CA suggested to keep a blog and share it between the three of us to assess progress and help JD in his work. CA also suggested that once a paper is read, JD shall make a summary to start building up the Literature Review and Bibliography. (Even if we decide not to use the paper.)

Next meeting:

- 4 July 2016 @ 5pm
- By this meeting JD is to provide a summary about Virtual Screening