Week 3 (01.11 - 07.11.2019)

Part 1

Paper: Recent applications of deep learning and machine intelligence on in silico drug discover

This paper talks about Machine Learning and Deep Learning concepts in general for a bit and then moves on to how these concepts can be applied in the context of drug discovery and drug development, and discusses the advantages and disadvantages of these methods over more traditional methods.

1.1 Introduction:

Drug discovery and development in earlier days was mostly done through experiments between compounds and possible targets. This was not only time and labour intensive, but also very costly. Recent years saw the emergence of high throughput screening (HTS), which allowed made it possible to conduct experiments to scan through thousands of different compounds and detect their bioactivity levels on selected targets.

But HTS has its own disadvantages. These experiments can also be time consuming and expensive and requires extensive biological and chemical libraries.

Then came the Virtual Screening (VS) methods, which is the field of estimation of unknown drug-target pairs using statistical models. Apart from drug discovery, VS can also be used for drug repurposing.

There are mainly 3 types of VS methods:

- **Structure-based** VS employs 3D structure of targets and com- pounds to model the interactions
- **Ligand-based** VS uses the molecular properties of compounds (mostly non-structural) to model the interactions with targets
- **Proteochemometric modeling (PCM)** approach models the inter- actions by combining non-structural descriptors of both com- pounds and targets at the input level.

1.2 Descriptors and Features for VS:

- A molecular descriptor, which is used to build VS models, should reflect the chemical and physical properties of the molecule, so that the statistical model can learn and generalise well.
- A popular sub-group of molecular descriptors are fingerprints (i.e. binary vectors), where each dimension of the vector represents presence (1) or absence (0) of a particular property.
- Target protein descriptors are employed along with compound molecular descriptors in PCM.

- Sequence-based target descriptors use the amino acid sequence of proteins, whereas structure-based descriptors use 3D atomic coordinates of proteins.

1.3 Databases and Gold-Standard Datasets:

- **Compound, bioactivity and target protein databases:** PubChem, ChEMBL, Drugbank. Each database has its own advantages and disadvantages and they usually complement each other in terms of data.

- **Gold standard datasets for VS:** In machine learning, the term 'gold-standard data sets' refers to reliable sets of information created to address a particular problem. In VS, gold-standard data sets generally comprise manually curated compound-target pairs and their bioactivity values.

1.4 Machine Learning Applications in VS:

Although there is no distinction in ML literature, VS methods can be divided into similarity-based and feature-based methods.

a) Similarity-based:
  - These methods rely on the assumption that biologically, topologically and chemically similar compounds have similar functions and bioactivities and, therefore, they have similar targets.
  - In chemical space, similarities are calculated by searching molecular substructure and isomorphism based on the representations of molecules such as SMILES and InChI. In target space, similarities are mainly calculated by sequence alignment methods.
  - Advantages:
    - When problems involve heterogenous data, different types of similarity matrices can be combined in the same model.
    - Sophisticated kernel methods can be applied.
    - Relatively simple to model.
  - Disadvantages:
    - Computationally not practical to apply on large datasets as it involves a high number of similarity calculations.

b) Feature-based:
  - In these methods, each instance (i.e. compound and/or target) is represented by a numerical feature vector, which reflects various types of physico-chemical and molecular properties of the corresponding molecules.
  - The constructed feature vectors are fed to a machine learning algorithm to create a predictive model for the interaction with the corresponding target. When a new query

compound’s feature vector is given to the trained model as input, the output of the predictive model is either active or inactive against the corresponding target protein.

- There are other flavours too, where fingerprints, sequence-based profiles and graph networks are used.
- Advantages:
    - They can reveal intrinsic properties of compounds and targets that play a critical role in DTI’s.
    - A problem specific feature-selection can be applied to obtain more accurate predictions.
- Disadvantages:
    - Selection of negative samples for the construction of training sets, which in turn leads to class imbalance problems.
    - High dimensionality of feature vectors.

1.5 Deep Learning Applications in VS:

DNN (Deep Neural Networks) yield successful learning of the representations of the input data through multiple levels of abstraction. There are many DNN architectures that have their advantages and disadvantages based on the nature of data being analysed and the type of features.

Eg: Deep convolution neural networks are extensively used in the field of Computer Vision

DNN-based techniques are also divided into two according to the number of prediction tasks in a model, such as the single- task and multi-task DNNs. All of these DNN architectures can be considered under the title of feature-based machine learning methods. The performance of a single-task DNN was said to increase with increase in size of training datasets.

1.6 Evaluation metrics and performance comparison of VS methods:

There are different metrics like precision and recall to measure the quality of VS methods and they each have their own properties. Using only precision as the evaluation metric would results in overlooking the high number of FN predictions, since precision does not take FNs into account. The same case is applied for the recall and the FPs. To overcome this issue, F1-score is employed, which is a harmonic mean of precision and recall, to consider both the FPs and FNs.

MCC is another measure which also is a balanced performance calculation metric similar to the F1-score. It was reported that MCC can very well be used for performance evaluation when classes are imbalanced [258]. The main difference between MCC and F1-score is that F1-score does not take TNs into account, whereas MCC does. Therefore, using MCC for performance evaluation can be more convenient, especially when one has a reliable negative training dataset.

1.7 Discussion and conclusion:

- A significant issue in predictive model development in VS is the training data set construction.

- Datapoint scarcity is even more pronounced in negative training set selection, which is also called the class imbalance problem.
- One of the solutions proposed for this problem is employing decoys, which are compounds that have similar physico-chemical properties but different topologies compared with the known active compounds for the selected targets. These decoy molecules are inactive against the corresponding targets; as a result, they can be used in negative test sets to accurately assess the performance of the models regarding the FPs.
- Deep learning methods have been reported to be robust against the noise in the training data, not only for negatives but also for positives.
- In the literature, it was indicated that the prediction performance of computational methods was highly dependent on the targets. Therefore, target-specific machine learning and feature selection methods can be investigated more to enhance accuracy of prediction
- For some of the traditional ML methods, such as the SVM, low amount of training instances is often sufficient; however, the training data should be error-free to generate a high-performance predictive model. It is generally the opposite for DNNs, as successful applications of DNN models are usually trained with a large number of instances even though they contain high error rates in some cases.

Part 2

Last week there were some problems creating the model for the data provided (eri-data.smi). I realized the issue was with nan and inf values in the dataset as well as the resulting molecular descriptor file (which was generated by the rdkit tool). I wasn't so sure about what to do with them, so I set all the nan and inf values to 0. I don't know if this is the correct behaviour or not. There are other methods like forward_fill too, which takes the column value from the previous row. I think another solution could be to impute the values.

Also, last week the galaxy server was extremely slow, so I couldn't use the 'Ensemble methods tool' to create a model for the data provided. The speed is better this week and I was successful in creating a model. Below is the link to the history:

https://usegalaxy.eu/u/lorraine_coelho/h/model-to-determine-if-the-compound-is-active-against-the-estrogen-nuclear-receptor

I also executed this in the script, using RandomForestClassifier and GradientBoostClassifier for comparison. RF had a prediction score of about 0.92 (with the following settings: min_samples_leaf=15, n_estimators=250, max_depth=20).

GB also gave a perdiction score of about 0.92 with the following settings: n_estimators=150, learning_rate=1, max_features=30, max_depth=50.

Part 3

Feature Selection:

For this task, I had to generate the molecular descriptor file using the **Mordred tool**. This gives about 1600 features, compared to the rdkit tool, which gives about 200 features. The task was to perform feature selection and weed out uncorrelated and unimportant features. I have used the Random Forest Classifier here. Corresponding script is featureSelection-mordred.py

I first tried out the feature selection tools provided by sklearn. The **SelectKBest** method. This is Univariate method that uses statistical models to select the features that best correlate with the output. I have used the f_classif statistical test for this. Below given is a table of performance scores with different number of selected features.

Number of features	**Performance**
40	0.90
30	0.88
20	0.88
15	0.90
10	0.89

Below given are the top 20 features selected by this method:

GI9

piPC10

piPC9

n6Ring

MDEC-23

piPC8

JGI8

TpiPC10

AATS7d

GATS8i

piPC7

ATSC1v

SpMAD_Dt

nG12FRing

AATS8d

C2SP2

PEOE_VSA7

AATS7v

fMF

AATS6d

The **ExtraTreesClassifier** tool ranks all the features in descending order with respect to a score that gives the feature importance of each feature against the output variable. We can select n best out of it.

Number of features	**Performance**
40	0.90
30	0.90
20	0.89
15	0.91
10	0.90

Below given are the 20 best features selected by this method:

AETA_beta_ns

NaaCH

AETA_beta

ATS2p

JGI9

GATS8i

AATS6d

StsC

nG12FRing

MWC03

GATS4d

C3SP3

ATSC4v

GGI6

ATS3dv

AATS6dv

AATSC1v

SpMAD_Dt

NsOH

C2SP2

There is some intersection from both the lists, but not a lot. And the performance improvement with feature selection is not that significant either.

According to literature, these methods are not very accurate. So, I tried a technique called **Backward Elimination**, which belongs to a set of algorithms called **Wrapper Methods,** which iteratively removes features that do not contribute to improvement in the model performance. The feature importance is estimated with the pvalue. If the pvalue is more than 0.05, the feature is removed.

But this method is computationally very intensive (albeit better than the filter methods) and it is not suited to our dataset as we have a lot of features and the algorithm takes a very long time to run. I couldn’t get a result after waiting for an hour, but probably it can be accomplished on a system with better resources. You can find the code to this in the fs-mordred2.py file.