This is the results document for the github.com/pcypw2/**AqSolMol** repository. For any questions regarding the code or results, please contact myself (Phil Wroe) or Joe Redshaw. The contact details for us both are found below:

* [pcypw2@nottingham.ac.uk](mailto:pcypw2@nottingham.ac.uk)
* [psxjr4@nottingham.ac.uk](mailto:psxjr4@nottingham.ac.uk)

**Molecular Descriptors**

The graph representation was compared against the MACCS key fingerprint and ECFP structural bit descriptors as these are the popular choice for their high predictive performance (figure 1). An additional benchmark model that replicates the ESOL model by Delaney et al. will be used which utilises physiochemical descriptors. Specifically, molecular weight (*MolWt*), lipophilicity (*logP*),rotatable bonds (*RB*) and aromatic proportion (*AP*). The descriptors were calculated using the SMILE strings provided in the *AqSolDB* and RDkit. For the graph representation, additional steps were required through Grakel to create the adjacency matrix for each graph

.

**Modelling Methods**

1. *Multiple Linear Regression*

The *AqSolDB* was passed through the ESOL multiple linear regression method as benchmark for predictive success. This was achieved using scikit-learn.

1. *Support Vector Regression*

SVMs implemented by scikit-learn where used to assess kernel performance. Table 1 highlights the kernel methods used. The vectorial kernel matrices were calculated in-house using scikit-learn, whilst the graphical kernel matrices were computed externally through Grakel before being inputted into the SVM.

**Model Building and Evaluation**

The SVM hyperparameters were optimised using a grid search 5-fold cross validation method with an exponentially growing sequence. E.g., c=2-5, 2-4, …, 25, g=2-15,2-13…, 25, e=1-3,1-2, 12. Prior to model fitting, the data underwent a linear scaler transformation. Both the scaling and grid search optimisation methods were performed using sci-kit learn. The n\_iter parameter for Weisfeiler Lehman and Weisfeiler Lehman Optimal Assignment kernel was optimised by testing a range of n\_iter from 0 to 15 and the model with best performing results was chosen. Each SVM model was built by with an 80:20 split the *AqSolDB* to form a training and test set. Model performance was assessed through calculation of the coefficient of determination (*r2*) and the error persistent in the model was represented through calculated of the root mean squared error (*RMSE*). All analysis was performed with scikit-lear

**Preliminary Analysis**

*Aims of the preliminary analysis:*

* To finetune the AqSolDB, ensuring the correct balance between allowing experimental noise into the model and being as representative of the vast potential of chemical space
* Create a benchmark model to compare support vector regression performance against

The first investigation involved splitting the AqSolDB into three groups to understand if the removal of data with high error had any significant improvement on prediction quality. **AqSolDB-A** contained only group 1 where there was only 1 reported compound and thus no standard deviation. **AqSolDB-B** which had groups 1,2, and 4 which introduced an error of <0.5 SD where there were more than one reported compound across the 9 datasets used to build the AqSolDB. Finally, **AqSolDB-C** which contained all 5 groups which included a standard deviation >0.5 where there are multiple occurrences of a compound.

The ESOL method by Delaney et al. was used to build a normalized MLR model for each of the three datasets (table 1). Whilst AqSolDB-A achieved slight improvement in prediction performance, overall, the three models were largely inaccurate. Given that the AqSolDB was in its raw state and had yet to undergo any pre-processing, this result does not come as a surprise. Figure 1 highlights the extreme ranges of datapoints of lipophilicity (logP) and molecular weight (MolWt) that are included in the AqSolDB. In the context of aqueous solubility modelling for a drug discovery project, inclusion of datapoint points over 1000 Da are not relevant and only hinders prediction performance as anything over this MolWt would not be considered in the pharmaceutical industry. A ‘tight’ and ‘broad’ range for MolWt and logP were proposed to focus the dataset and make it relevant for drug-like molecules.

**Table 1.** MLR Regression performance of AqSolDB datasets with increasing amount of experimental error. r2 = coefficient of determination and RMSE = root mean squared error.

|  |  |  |
| --- | --- | --- |
|  | *r2* | *RMSE* |
| AqSolDB-A | 0.46 | 1.74 |
| AqSolDB-B | 0.44 | 1.79 |
| AqSolDB-C | 0.44 | 1.77 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Dataset*** | ***MolWt*** | ***n*** | ***LogP*** | ***Lower r2 CI boundary*** | ***r2*** | ***Upper CI boundary*** | ***RMSE*** |
| *1* | *≤500* | *9076* | *-7 ≤ x ≤ 7* | *0.55* | *0.57* | *0.58* | *1.42* |
| *2* | *≤500* | *9329* | *-11 ≤ x ≤ 11* | *0.58* | *0.59* | *0.60* | *1.46* |
| *3* | *≤1000* | *9391* | *-7 ≤ x ≤ 7* | *0.53* | *0.54* | *0.55* | *1.48* |
| *4* | *≤1000* | *9783* | *-11 ≤ x ≤ 11* | *0.55* | *0.56* | *0.57* | *1.55* |
| *5* | *≤5299* | *9983* | *-41 ≤ x ≤ 68* | *0.43* | *0.44* | *0.46* | *1.77* |

**Table 2.** Results of MLR regression on tight and broad bands of MolWt and logP (tests 1-4) compared against the raw dataset (test 5). n = number of datapoints, CI = confidence internal. Green highlight shows the benchmark model and dataset used for the kernel investigation

**Figure 1.** Histogram plots showing the spread of solubility, molecular weight and lipophilicity for the AqSolDB

When compared to the raw dataset, each test performed significantly better, but there is little difference when comparing between tests. A 95% confidence interval (CI) limit therefore was calculated to determine which dataset, if any, had significantly improvements on prediction quality. Findings from this, show that the dataset 2 had a significant improvement in prediction performance over both dataset 3 and 4, which had a higher cut off for MolWt. However, we decided that inclusion of as many datapoints as possible was the route to go down to increase the flexibility of our models to unseen molecular structures. This increased inclusion of structures in the extreme ranges of MolWt and logP better represented the chemical space of drug-like molecules better which leads to higher accuracy predictions. If however, you were predicting a class of small MolWt compounds, using the tighter dataset will yield better results as it is unnecessary to include larger molecular weight compounds. Given the binomial-like distribution of solubility values in figure 1, retaining the extreme solubility datapoints was an import to creating a more generalised model. To confirm this, the MLR equations generated from both test 2 and 4 (equations 1 and 2) were applied to the validation dataset (table 4). Whilst this showed no significant difference between the two as the CI boundaries for each test overlap less error in broader dataset, thereby showing the need of a large representation of different molecules. Therefore we used the models highlighted in green in table 2 and table 3 as our benchmark for test and validation results.

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |

**Table 3.** Applied MLR equation of dataset 2 (equation 1) and dataset 4 (equation 2) on an external validation set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Equation** | **Lower *r2* CI boundary** | ***r2*** | **Upper *r2* CI boundary** | **RMSE** |
| (1) | 0.47 | 0.51 | 0.54 | 1.41 |
| (2) | 0.48 | 0.52 | 0.56 | 1.33 |

**Kernel Investigation Results**

See READ.me file and methods folder for more information on the precise methodology on how the parameters C, gamma, epsilon and n\_iter were optimised. Vectorial kernels usesxsd were linear, radial basis function (RBF), polynomial and sigmoid. As for graph kernels, we explored neighbourhood hashed (NH), random walk (RW), propagation (PG), ordered dag decomposition (ODD), pyramid match (PM), shortest path (SP), graphlet sampling (GS), vertex histogram (VH), Weisfeiler Lehman subtress (WLS) and Weisfeiler-Lehman optimal assignment (WLOA).

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Descriptor** | **Kernel** | **C** | **gamma** | **Epsilon** | **Degree** | **n\_iter** | **Training set** | | **Test set** | | **Validation set** | |
| r2 | RMSE | r2 | RMSE | r2 | RMSE |
| ESOL | Linear | 32 | - | 1 | - | - | 0.55 | 0.67 | 0.54 | 0.62 | 0.62 | 0.62 |
| RBF | 20 | 2-1 | 0.1 | - | - | 0.71 | 0.54 | 0.67 | 0.59 | 0.54 | 0.68 |
| Poly | 2-5 | 2-5 | 0.1 | 1 | - | 0.55 | 0.67 | 0.55 | 0.69 | 0.54 | 0.68 |
| Poly | 2-4 | 0.5 | 1 | 2 | - | 0.15 | 0.92 | 0.19 | 0.92 | 0.15 | 0.92 |
| Poly | 2-5 | 2-3 | 1 | 3 | - | 0.21 | 0.88 | 0.23 | 0.90 | 0.19 | 0.90 |
| Sigmoid | 25 | 2-15 | 0.1 | - | - | 0.55 | 0.67 | 0.54 | 0.69 | 0.54 | 0.68 |
| MACCS | Linear | 2-4 | - | 0.0001 | - | - | 0.6 | 0.63 | 0.56 | 0.68 | 0.47 | 0.72 |
| RBF | 2 | 2-7 | 0.1 | - | - | 0.89 | 0.33 | 0.73 | 0.53 | 0.62 | 0.62 |
| Poly | 2-4 | 2-2 | 0.1 | 1 | - | 0.6 | 0.63 | 0.56 | 0.67 | 0.5 | 0.7 |
| Poly | 2-5 | 2-4 | 0.1 | 2 | - | 0.83 | 0.41 | 0.68 | 0.58 | 0.71 | 0.51 |
| Poly | 2 | 2-7 | 0.1 | 3 | - | 0.88 | 0.35 | 0.69 | 0.56 | 0.18 | 0.9 |
| Sigmoid | 25 | 2-13 | 0.1 | - | - | 0.6 | 0.63 | 0.56 | 0.67 | 0.51 | 0.7 |
| ECFP | Linear | 2-5 | - | 0.1 | - | - | 0.72 | 0.53 | 0.52 | 0.71 | 0.45 | 0.74 |
| RBF | 2 | 2-9 | 0.0001 | - | - | 0.93 | 0.27 | 0.62 | 0.63 | 0.54 | 0.62 |
| Poly | 2-5 | 2-7 | 0.1 | 1 | - | 0.69 | 0.56 | 0.56 | 0.67 | 0.53 | 0.69 |
| Poly | 215 | 2-6 | 0.1 | 2 | - | 0.96 | 0.21 | 0.56 | 0.67 | 0.46 | 0.74 |
| Poly | 2-5 | 2-6 | 0.1 | 3 | - | 0.97 | 0.16 | 0.4 | 0.79 | 0.2 | 0.9 |
| Sigmoid | 23 | 2-15 | 0.1 | - | - | 0.68 | 0.56 | 0.56 | 0.67 | 0.53 | 0.69 |
| Graphical | NH | 8 | 2-15 | 0.1 | - | - | 0.97 | 0.37 | 0.77 | 1.13 | 0.74 | 0.92 |
| RW | 256 | 2-15 |  | - | - | 0.04 | 2.28 | 0.01 | 1.82 | 0 | 1.82 |
| PG | 23 | 2-15 | 1 | - | - | 0.56 | 1.55 | 0.45 | 1.77 | 0.35 | 1.47 |
| ODD | 2 | 2-15 | 0.1 | - | - | 0.73 | 1.20 | 0.51 | 1.66 | 0.52 | 1.25 |
| PM | 22 | 2-15 | 1 | - | - | 0.71 | 1.25 | 0.64 | 1.43 | 0.5 | 1.28 |
| SP | 2-2 | 2-15 | 1 | - | - | 0.25 | 2.02 | 0.23 | 2.09 | 0.02 | 1.80 |
| GS | 2 | 2-15 | 1 | - | - | 0.06 | 2.25 | 0.02 | 2.36 | -0.01 | 1.82 |
| VH | 2 | 2-15 | 1 | - | - | 0.25 | 2.02 | 0.25 | 2.15 | 0.18 | 1.68 |
| WLS | 23 | 2-15 | 1 | - | 6 | 0.93 | 0.64 | 0.7 | 1.29 | 0.69 | 1 |
| WLOA | 2 | 2-15 | 0.1 | - | 5 | 0.92 | 1.12 | 0.78 | 1.12 | 0.75 | 0.91 |

**Predicted vs Experimental Plots**

Below are the performance plots of each kernel method. The dark blue plot (left) was built from the training dataset, the light blue plot (middle) is from the test set and the green plot (right) is the validation dataset. For polynomial kernel, only the best performing degree is shown (ESOL = 1, MACCS =2, ECFP =2)

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

**Conclusions & Further Work**

This research focussed on application graph kernel to aqueous solubility modelling and presented an array of different kernel types. Whilst there was a varied success throughout, the stand-out performing graph kernels, neighbourhood hashed, Weisfeiler-Lehman subtree and Weisfeiler Lehman optimal assignment had equivocal or improved prediction quality compared against the best performing vectorial kernels. Computationally, WLOA kernel took significantly longer and would not recommend using before trying the NH or WLS first.

Any contributions to this work are invited and encouraged! Particular work of interest could involve application into different dataset that of high experimental accuracy or focusing on a particular class of compounds. Additionally, many more graph kernel types exist than just the ones mentioned in this research have yet to be applied to predicting solubility. I believe the AqSolMol has acted as a good proof of concept and should spark more interest in the field. Particular work could involve using indefinite kernels that are inertially non-positive definite.