Usage guidelines

contact sure@sunderland.ac.uk.

Development of Functional Computational Workflow Methods for Quality Automated Molecular Modelling-based Drug Design

Fiona McMahon[a], *Peter Dawson*[a], *Roz Anderson*[a], *Ben Sattelle*[b], *Rob Warrender*[a], *Mark Gray*[a]
[a] Sunderland University, [c] Molplex LTD

Introduction

Drug discovery is costly and time consuming. The overwhelming majority of compounds tested experimentally are ineffective or actively harmful. *In silico* methods are popular in industry as they can eliminate compounds doomed to failure from a library of drug candidates. However, users spend a significant amount of their day checking calculations, and moving files between programmes. This limits the number of drug candidates that can be assessed in any given time. We are creating an automated workflow to handle calculations so that user time and computer resources are utilised with high efficiency.
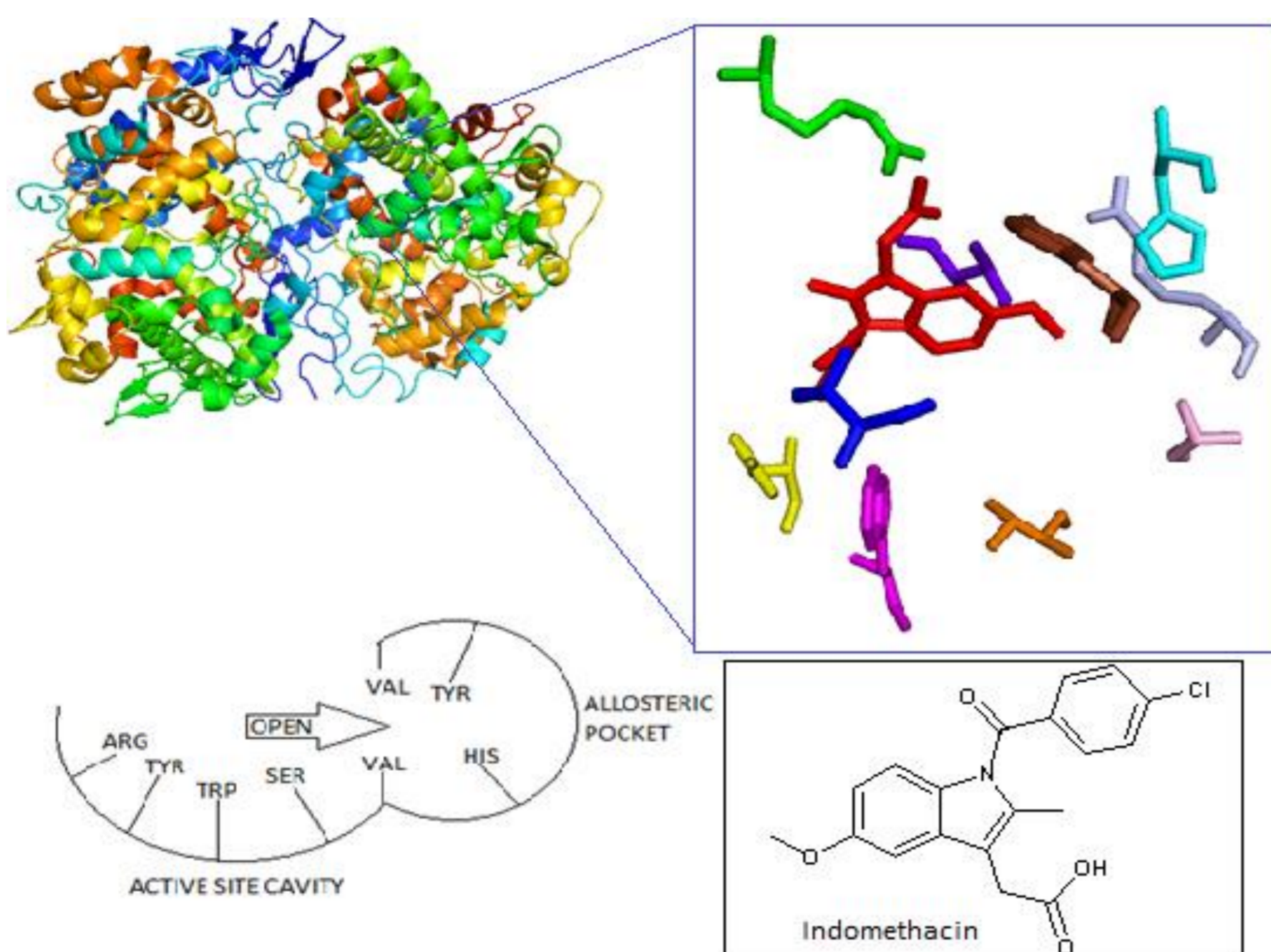
A Functional Workflow for ab initio Drug Discovery

A full evaluation of a compound would assess key physicochemical properties such as $\log P$, pK_a and solubility, estimate toxicity, and predict binding to a given drug target. A workflow enabling a computer to determine if a drug candidate within a compound library is promising in one aspect, and advancing it automatically to the next step in its overall evaluation, would significantly reduce user input and make calculations more efficient. Although programmes exist that estimate physicochemical properties and binding empirically, they fail with structures that resemble the compounds used to create their parameter set. *ab initio* calculations are computationally expensive but can succeed where empirical methods struggle. The increase in computing power in line with Moore's Law means that an *ab initio* approach is now feasible.

Cyclooxygenase: A "live" drug discovery problem

Validation of the workflow under construction uses the COX (cyclooxygenase) enzyme as a drug target, and known COX inhibitors to test the system. COX inhibitors are used clinically for the treatment of inflammation and pain, while rofecoxib (Vioxx) was taken off the market in 2004 after it was found to increase the risk of heart attack and stroke. As this is a well researched area, the experimental data we require to evaluate all aspects of the performance of our system is already in the public realm. If we can show that we can produce a computational system that reproduces what is seen experimentally and in the clinic for known COX inhibitors, we will be in a position to evaluate untested compounds against COX and other known drug targets.

COX-2, its active site and interactions with an NSAID

The Workflow

This is a preliminary look at the overall workflow. It shows 2-Way communication tools between Gaussian09 (G09) and AutoDock, leading to an overall database that will hold any relevant data:
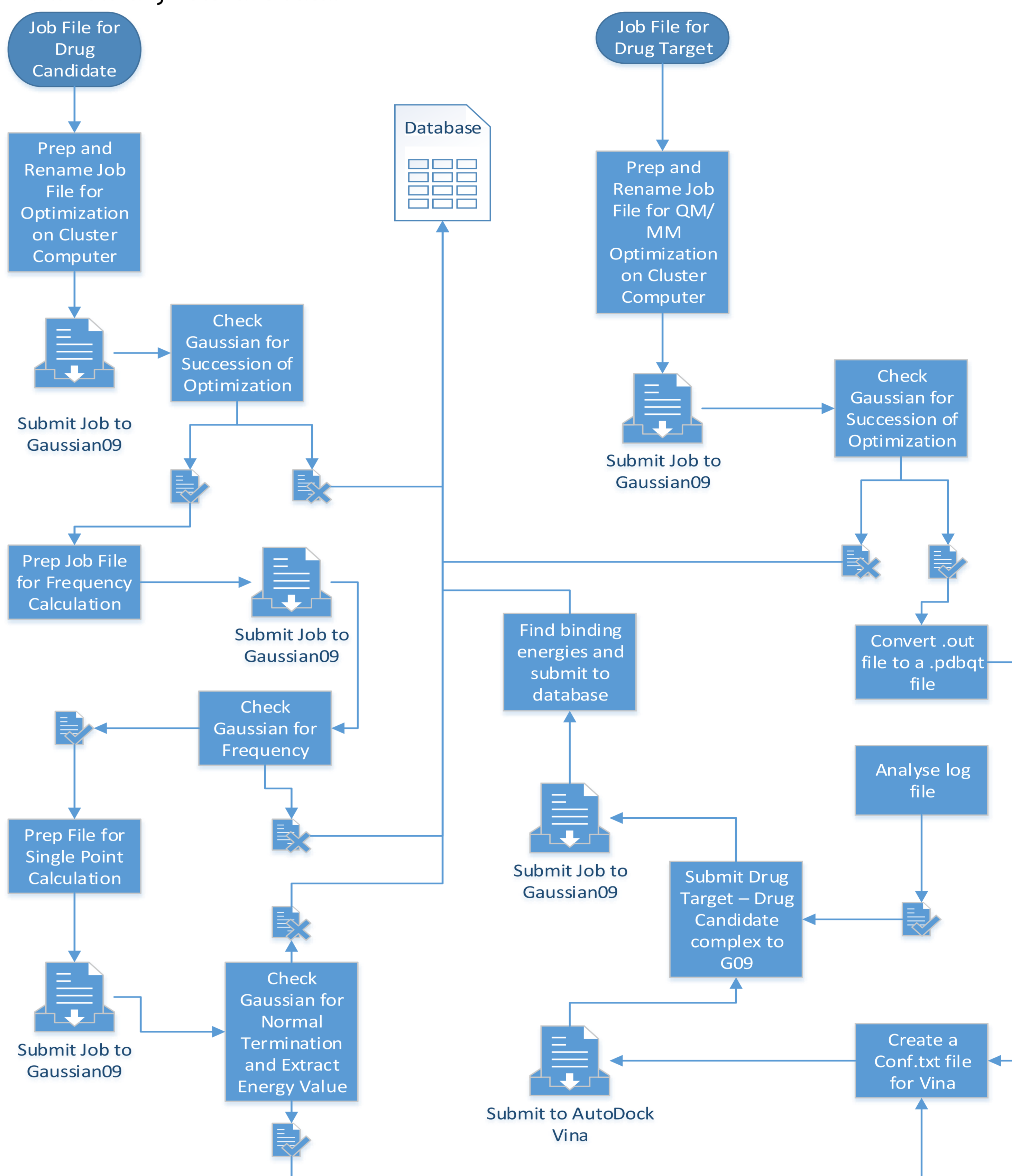
Results

To date, we have built programs that submit jobs to G09, analyse the output files and create new jobs files for the next calculations. The current workflow can be seen below:

Compound is submitted to workflow for investigation

After a geometry optimization calculation is finished, a program will extract data from the output file, determine whether it was successful, and if it is, create a new job file and submit a frequency calculation to G09

After frequency calculation is completed, a program will extract data from the output file and depending on the results, create a new job file for a high level single point calculation

Methods

Programs able to handle data and make decisions are being written in Python. These scripts are called into a C# environment within a novel "passport" system developed at the University of Sunderland.[1] The passport is a bespoke computational platform used to show where files have been, and where they will need to go next to undergo further calculations. It is utilized on the cluster computer available at the Department of Computing, Engineering and Technology, which has 40 nodes, each with two dual-core 2.6GHz chips and 8GB of RAM, meaning up to 160 cores are available for calculations.

Conclusion

We have written scripts to analyse output files, create new job files, and submit jobs to Gaussian for calculations using a python/C# interface. This amalgamation of the two languages allows us to use different techniques which are optimized for their specific duties. C# works well within the computing environment, allowing for high speed and consistency during computer tasks. In contrast, Python allows for dynamic handling of variables, and is widely established in the scientific community.

References

1 Tindle, J., Gray, M., Warrender, R. L., Ginty, K. & Dawson, P. Application Framework for Computational Chemistry (AFCC) Applied to New Drug Discovery. *Int. J. Grid High Perform. Comput.* **4,** 46–62 (2012).