Developing a workflow management system for fragment-based virtual screening

Simon Bray

September 25, 2023

Albert-Ludwigs-Universität Freiburg im Breisgau

Technische Fakultät
Institut für Informatik
Lehrstuhl für Bioinformatik

Dissertation zur Erlangung des akademischen Grades Doctor rerum naturalium (Dr. rer. nat.) der Technischen Fakultät der Albert-Ludwigs-Universität Freiburg im Breisgau

Developing a workflow management system for fragment-based virtual screening

Simon Bray

1. Reviewer	Prof. Dr. Rolf Backofen Lehrstuhl für Bioinformatik Albert-Ludwigs-Universität Freiburg im Breisgau
2. Reviewer	Dr. Steffen Wolf Biomolecular Dynamics Group Albert-Ludwigs-Universität Freiburg im Breisgau
Supervisors	Prof. Dr. Rolf Backofen and Dr. Steffen Wolf

September 25, 2023

Simon Bray
Developing a workflow management system for fragment-based virtual screening
Defence: September 25, 2023
Dean of the Faculty of Engineering: Prof. Dr. Roland Zengerle
First reviewer: Prof. Dr. Rolf Backofen
Second reviewer: Dr. Steffen Wolf

Albert-Ludwigs-Universität Freiburg im Breisgau
Lehrstuhl für Bioinformatik
Institut für Informatik
Technische Fakultät
Georg-Köhler-Allee 79, Freiburg im Breisgau, Germany
and Freiburg im Breisgau

Publications

Peer-reviewed journal articles

- **Simon Bray**, John Chilton, Matthias Bernt, Nicola Soranzo, Marius van den Beek, Bérénice Batut, Helena Rasche, Martin Čech, Peter Cock, Björn Grüning, Anton Nekrutenko. Planemo: a command-line toolkit for developing, deploying, and executing scientific data analyses. *Genome Research* (under review), https://doi.org/10.1101/2022.03.13.483965

- **Simon Bray**, Victor Tänzel, Steffen Wolf. Ligand Unbinding Pathway and Mechanism Analysis Assisted by Machine Learning and Graph Methods. *Journal of Chemical Information and Modelling*, Volume 62, Issue 19, 29 September 2022, https://doi.org/10.1021/acs.jcim.2c00634

- **Simon Bray**, Tim Dudgeon, Rachael Skyner, Rolf Backofen, Björn Grüning , Frank von Delft. Galaxy workflows for fragment-based virtual screening: a case study on the SARS-CoV-2 main protease. *Journal of Cheminformatics*, Volume 14, Article number: 22, 12 April 2022, https://doi.org/10.1186/s13321-022-00588-6

- Wolfgang Maier, **Simon Bray**, Marius van den Beek, Dave Bouvier, Nathan Coraor, Milad Miladi, Babita Singh, Jordi Rambla De Argila, Dannon Baker, Nathan Roach, Simon Gladman, Frederik Coppens, Darren P. Martin, Andrew Lonie, Björn Grüning, Sergei L. Kosakovsky Pond, Anton Nekrutenko. Ready-to-use public infrastructure for global SARS-CoV-2 monitoring. *Nature Biotechnology*, Volume 39, pages 1178–1179, 29 September 2021, https://doi.org/10.1038/s41587-021-01069-1

- Qiang Gu, Anup Kumar, **Simon Bray**, Allison Creason, Alireza Khanteymoori, Vahid Jalili, Björn Grüning, Jeremy Goecks. Galaxy-ML: An accessible, reproducible, and scalable machine learning toolkit for biomedicine. *PLOS Computational Biology*. Volume 17, Issue 6, Article number: e1009014, 1 June 2021, https://doi.org/10.1371/journal.pcbi.1009014

- **Simon A. Bray**, Tharindu Senapathi, Christopher B. Barnett, Björn A. Grüning. Intuitive, reproducible high-throughput molecular dynamics in Galaxy: a tutorial. *Journal of Cheminformatics*, Volume 12, Article number: 54, 10 September 2020, https://doi.org/10.1186/s13321-020-00451-6

- Steffen Wolf, Benjamin Lickert, **Simon Bray**, Gerhard Stock. Multisecond ligand dissociation dynamics from atomistic simulations. *Nature Communications*, Volume 11, Article number: 2918, 10 June 2020, https://doi.org/10.1038/s41467-020-16655-1
- **Simon A. Bray**, Xavier Lucas, Anup Kumar, Björn A. Grüning. The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform. *Journal of Cheminformatics*, Volume 12, Article number: 40, 01 June 2020, https://doi.org/10.1186/s13321-020-00442-7
- Tharindu Senapathi, **Simon Bray**, Christopher B. Barnett, Björn Grüning, Kevin J. Naidoo. Biomolecular Reaction and Interaction Dynamics Global Environment (BRIDGE). *Bioinformatics*, Volume 35, Issue 18, pages 3508–3509, 15 September 2019, https://doi.org/10.1093/bioinformatics/btz107.

Oral presentations

- **Simon Bray**. Automatic updates of Galaxy tools and workflows. Galaxy Community Call, online (worldwide). 9 June 2022.
- **Simon Bray**. The Galaxy platform as a solution for reproducible molecular dynamics workflows. Research-Data Management in Biophysics workshop, European Biophysics Conference, online (Vienna, Austria). 24 July 2021.
- **Simon Bray**. Computational drug screening against the SARS-CoV-2 main protease. Virtual Freiburg COVID-19 workshop, Nationales Forschungsnetzwerk COVID-19, online (Universitätsklinikum Freiburg, Germany). 24 July 2021.
- **Simon Bray**. Designing and executing workflows for virtual screening of the SARS-CoV-2 main protease. Bioinformatics Community Conference, online (worldwide). 21 July 2020.
- Tim Dudgeon, **Simon Bray**. Cheminformatics: Screening of the main protease. ELIXIR Webinar series, online (worldwide). 14 May 2020.
- **Simon Bray**. The ChemicalToolbox: Computational chemistry in Galaxy. Galaxy Community Conference, Freiburg, Germany. 4 July 2019.

Poster presentations

- **Simon Bray**, Lorraine Coelho, Björn Grüning. Automating Galaxy tool requirement updates with Planemo. Galaxy Community Conference, online (worldwide). 6 July 2021.
- **Simon Bray**, Alireza Khanteymoori, Björn Grüning. The ChemicalToolbox: Computational chemistry in Galaxy. Galaxy Community Conference, Freiburg. 4 July 2019.
- **Simon Bray**, Steffen Wolf, Gerhard Stock. Analyzing Protein-Ligand Dissociation using Dissipation-Corrected Targeted Molecular Dynamics. Workshop on Computer Simulation and Theory of Macromolecules, Hünfeld. 23 March 2019.

Workshops

- **Simon Bray**. Training: Protein-ligand docking and cheminformatics with the Galaxy platform. Indian Conference of Bioinformatics pre-conference workshop, online (India). 9 November 2021.
- **Simon Bray**, Wolfgang Maier. Training: Automating Galaxy workflows using the command line. Galaxy Community Conference, online (worldwide). 28 June - 2 July 2021.
- **Simon Bray**, Chris Barnett. Training: High-throughput molecular dynamics with Galaxy. Bioinformatics Community Conference, online (worldwide). 28 June - 2 July 2021.
- **Simon Bray**. Demonstration: Computational chemistry in Galaxy. ELIXIR-CZ workshop, online (Czech Republic). 7 December 2020.
- **Simon Bray**, Chris Barnett, Tharindu Senapathi. Training: High-throughput molecular dynamics with Galaxy. Bioinformatics Community Conference, online (worldwide). 18 July 2020.

Student supervision

The Bioinformatics Group has an internal requirement that each doctoral student supervise at minimum 2 master's theses or equivalent, where a master's project or B.Sc. thesis counts as half a master's thesis. This requirement was fulfilled by supervising the following students:

Master's thesis:

- **Lorraine Coelho**. Multi Protein-Ligand Interaction Prediction using Machine Learning Models. Completed: 19 July 2021.

Master's projects:

- **Öner Aydogan**. Tool Resource Prediction for Genomic Datasets. Completed: 12 October 2022.
- **Abdus Salam Khazi**. Binding Affinity Prediction of Protein-Ligand Complexes. Completed: 18 October 2021.
- **Erik Schill**. BioBlend to Galaxy API extension and OpenAPI specification. Completed: 17 May 2021.
- **Lorraine Coelho**. A Robust Machine Learning Pipeline for Cheminformatic Tox21 Datasets. Completed: 22 July 2020.

Abstract

Drug development is a long, complex and expensive process. In particular, the first step of obtaining an initial list of drug candidates is challenging. Experimental screening, for example using protein-ligand binding assays, is fundamentally limited, and as a result, the concept of virtual screening comes into play. Virtual screening involves the use of *in silico* experiments such as statistical analyses, protein-ligand docking, and free energy calculations based on molecular dynamics (MD) simulation, in order to predict whether a particular compound is likely to bind to a particular target protein. Often, an initial list of candidates is generated by a fragment-approach, where a fragment is a small organic compound that can serve as a substructure for a putative drug candidate. Fragments can be found in either an experimental or theoretical manner, and can then be combined, or amended by the addition of other functional groups, in order to produce a list of candidate molecules.

There is then a need to determine the likelihood that these candidates bind to the target protein. There are several computer-based methods that can be of service in this task; these methods are not mutually exclusive, but on the contrary are typically used sequentially as well as in parallel. However, they require different amounts and types of computational resources, and careful planning is therefore required to manage resources, organise the software tools as complete workflows, and then to deploy them. To organise and perform the analysis, the scientist can use a workflow management system. Such systems allow multiple tools to be concatenated into a single pipeline, which can then be can be executed via the command line or a graphical interface. This has the advantage of being more convenient than the tedious execution of individual tools one after the other and helps avoid any manual errors. For highly complex analyses that require several different software tools with stepwise repetition, such as MD simulations for hundreds of ligands against a single target protein, the use of a workflow management system is the only viable option. Another challenge in virtual screening is reproducibility. In a reproducible scientific work, other scientists must be able to critically evaluate the work by performing the same experiments or simulations themselves and thus verifying the results. The issue of reproducibility has received much attention recently, including in the field of computational chemistry and virtual screening. The use of a workflow management

system helps to increase the reproducibility of a study, because the details of all tools run, with parameters and all versions of the tool software, are recorded to make the analyses repeatable for other scientists who want to verify their work.

The focus of this work was to develop a platform for fragment-based virtual screening based on the Galaxy workflow management system. This platform can be used either through a graphical web-based interface or through the command-line - the latter is a useful alternative for complex simulations or analyses that may require additional scripting. In order to make the use of the command line easier, significant contributions were made to Planemo and BioBlend, two Python libraries that allow direct access to Galaxy via the Application programming Interface (API). In order to demonstrate the utility of the platform developed, two projects were carried out using the developed tools and workflows.

First, a study was performed on the T4 lysozyme mutant L99A in complex with benzene using the dcTMD technique as a model system for fragment-protein binding. T4L-L99A is a commonly used model system for free energy calculations, and is especially useful as a model for fragment binding, due to the small size of the pocket and the benzene ligand, which is typical for the compounds and pockets generally used in fragment-based screening studies, and the fact that benzene binds rather weakly. Like many MD methods, dcTMD requires the execution of a large number of steps in sequence, and requires the creation of an ensemble of simulations, both features which benefit from the use of a workflow management system. The analysis was able to uncover multiple unbinding pathways, an essential feature of the dcTMD method, and to characterise the thermodynamics and kinetics of several of these. The final results were comparable to experimental benchmarks.

Second, a virtual screening was performed with the aim of identifying effective inhibitors of the major protease of the SARS-CoV virus; 53,000 compounds were generated based on 22 non-covalent crystallographic fragments, and their binding ability was analysed sequentially by protein-ligand docking, MMGBSA calculations and dcTMD simulations. Several million docking poses were generated, and scored by experimental validation against the crystallographic fragment structures. Over 200 compounds were then assessed by MMGBSA, followed by a further filtering and execution of a dcTMD workflow for 50 compounds. One fragment, which enforces a conformational change on the protein binding site, was found to confer particularly

strong binding ability on derived compounds, and it was shown that particular interactions correlated especially strongly with both MMGBSA and dcTMD scores.

Zusammenfassung

Die Entwicklung von Medikamenten ist ein langer, komplexer und teurer Prozess. Vor allem der erste Schritt, eine Liste von Wirkstoffkandidaten zu erstellen, ist eine Herausforderung. Das experimentelle Screening, z. B. mit Protein-Ligand-Bindungsassays, ist grundsätzlich begrenzt, sodass das Konzept des virtuellen Screenings ins Spiel kommt. Das virtuelle Screening umfasst *in silico* Experimente wie statistische Analysen, Protein-Ligand-Docking und Berechnungen der freien Energie auf Grundlage von Molekulardynamiksimulationen (MD-Simulationen), um die Bindungswahrscheinlichkeit eines bestimmten Protein-Liganden-Systems zu vorherzusagen. Häufig wird eine initiale Liste von Kandidaten mit Hilfe eines Fragment-Ansatzes erstellt, wobei ein Fragment eine kleine organische Verbindung ist, die als Substruktur für einen mutmaßlichen Wirkstoffkandidaten dienen kann. Fragmente können entweder auf experimentelle oder theoretische Weise gefunden werden und dann kombiniert oder durch andere funktionellen Gruppen ergänzt werden, um eine Liste von Kandidatenmolekülen zu erstellen.

Anschließend muss ermittelt werden, inwieweit diese Kandidaten geeignet sind, an das Zielprotein zu binden. Es gibt mehrere computergestützte Methoden, die bei dieser Aufgabe hilfreich sein können; diese Methoden schließen einander nicht aus, sondern werden vielmehr in der Regel sowohl aufeinanderfolgend als auch parallel eingesetzt. Sie erfordern jedoch unterschiedliche Mengen und Arten von Rechenressourcen, so dass eine sorgfältige Planung erforderlich ist, um die Ressourcen zu verwalten, die Software-Tools als vollständige Workflows zu organisieren und sie dann einzusetzen. Um die Analyse zu organisieren und durchzuführen, kann der Wissenschaftler ein Workflow-Management-System verwenden. Solche Systeme ermöglichen es, mehrere Tools zu einer einzigen Pipeline zu verketten, die dann über die Befehlszeile oder eine grafische Schnittstelle ausgeführt werden kann. Dies hat den Vorteil, dass es bequemer ist als die mühsame Ausführung der einzelnen Tools nacheinander und hilft, manuelle Fehler zu vermeiden. Für hochkomplexe Analysen, die mehrere verschiedene Softwaretools mit schrittweiser Wiederholung erfordern, wie z.B. MD-Simulationen für Hunderte von Liganden gegen ein einziges Zielprotein, ist der Einsatz eines Workflow-Management-Systems die einzige praktikable Option. Eine weitere Herausforderung beim virtuellen Screening ist die Reproduzierbarkeit.

Bei einer reproduzierbaren wissenschaftlichen Arbeit müssen andere Wissenschaftler in der Lage sein, die Arbeit kritisch zu bewerten, indem sie dieselben Experimente oder Simulationen selbst durchführen und somit die Ergebnisse verifizieren. Die Frage der Reproduzierbarkeit hat in letzter Zeit viel Aufmerksamkeit erhalten, auch im Bereich der computergestützten Chemie und des virtuellen Screenings. Der Einsatz eines Workflow-Management-Systems trägt dazu bei, die Reproduzierbarkeit einer Studie zu erhöhen, da die Details aller ausgeführten Tools mit Parametern und allen Versionen der Software aufgezeichnet werden, um die Analysen für andere Wissenschaftler, die ihre Arbeit überprüfen wollen, wiederholbar zu machen.

Der Schwerpunkt dieser Arbeit lag auf der Entwicklung einer Plattform für fragmentbasiertes virtuelles Screening auf der Grundlage des Workflow-Management-Systems Galaxy. Diese Plattform kann entweder über eine grafische, webbasierte Oberfläche oder über die Kommandozeile verwendet werden - das letzteres ist eine nützliche Alternative für komplexe Simulationen oder Analysen, die möglicherweise zusätzliche Skripte erfordern. Um die Nutzung der Kommandozeile zu erleichtern, wurden wesentliche Beiträge zu Planemo und BioBlend geleistet, zwei Python-Bibliotheken, die einen direkten Zugriff auf Galaxy über die *Application Programming Interface* (API) ermöglichen. Um die Nützlichkeit der entwickelten Plattform zu demonstrieren, wurden zwei Projekte mithilfe der entwickelten Werkzeuge und Workflows durchgeführt.

Erstens wurde eine Studie über die T4-Lysozym-Mutante L99A im Komplex mit Benzol unter Verwendung der dcTMD-Technik als Modellsystem für die Fragment-Protein-Bindung durchgeführt. T4L-L99A ist ein häufig verwendetes Modellsystem für Berechnungen der freien Energie und eignet sich besonders gut als Modell für die Bindung von Fragmenten, da die Tasche und der Benzol-Ligand klein sind, was typisch für die Verbindungen und Taschen ist, die im Allgemeinen in fragmentbasierten Screening-Studien verwendet werden, und da Benzol eher schwach bindet. Wie viele MD-Methoden erfordert dcTMD die Ausführung einer großen Anzahl von Schritten in Folge und die Erstellung eines Ensembles von Simulationen, beides Eigenschaften, die von der Verwendung eines Workflow-Management-Systems profitieren. Die Analyse konnte mehrere Abbindungspfade aufdecken, ein wesentliches Merkmal der dcTMD-Methode, und die Thermodynamik und Kinetik mehrerer dieser Pfade charakterisieren. Endergebnisse waren mit experimentellen Daten vergleichbar.

Zweitens wurde ein virtuelles Screening mit dem Ziel durchgeführt, wirksame Inhibitoren der Hauptprotease (*main protease*) des SARS-CoV-Virus zu identifizieren. 53.000 Verbindungen wurden auf der Grundlage von 22 nichtkovalenten kristallographischen Fragmenten generiert, und ihre Bindungsfähigkeit wurde reihenweise durch Protein-Ligand-Docking, MMGBSA-Berechnungen und dcTMD-Simulationen analysiert. Es wurden mehrere Millionen Docking-Posen generiert und durch experimentelle Validierung anhand der kristallographischen Fragmentstrukturen bewertet. Über 200 der Verbindungen wurden anschließend mit MMGBSA bewertet, gefolgt von einer weiteren Filterung und der Durchführung eines dcTMD-Workflows für 50 Verbindungen. Es wurde festgestellt, dass ein Fragment, das eine Konformationsänderung an der Proteinbindungsstelle erzwingt, eine besonders starke Bindungsfähigkeit an abgeleitete Verbindungen verleiht, und es wurde gezeigt, dass bestimmte Wechselwirkungen besonders stark mit den MMGBSA- und dcTMD-Bewertungen korrelierten.

Acknowledgement

Whatever you do, in word or deed, do everything in the name of the Lord Jesus, giving thanks to God the Father through him. (Colossians 3:17)

I would like to thank the following people for their friendly and constructive collaborations during my thesis: Tharindu Senapathi and Chris Barnett from the Scientific Computing Unit, University of Cape Town; Steffen Wolf from the Institute of Physics, University of Freiburg; Tim Dudgeon from Informatics Matters; Rachael Skyner and Frank von Delft from the XChem group, Diamond Light Source. I would like to thank the Galaxy community as a whole for being welcoming and providing numerous learning opportunities, and in particular Marius van den Beek, Nicola Soranzo, and John Chilton, for the time spent reviewing and improving my pull requests.

I would like to thank all my colleagues in the Galaxy team at the University of Freiburg for their work together on many interesting projects and the various different ways they have supported and encouraged me during the preparation of this doctoral thesis. In particular, I would like to thank Gianmauro Cuccuru for his assistance deploying some unusual Galaxy tools. I would like to thank Anup Kumar, Mehmet Tekman and Joachim Wolff, for all their useful advice and support, not to mention many funny discussions about science, the process of completing a PhD and various other topics. Thanks to Cristóbal Gallardo for his book recommendations (even though I have ignored them all). I would like to thank Rolf Backofen for providing the opportunity to carry out this doctoral thesis in his group. Last but not least, I would like to thank Björn Grüning for the many times he provided help and support over the last four years, not only to me but to all those around the world involved in or making use of the Galaxy project.

Contents

1 Introduction

Proteins are the workhorses of the cell. They are responsible for the broad range of functionality which is essential for life, from proteins which play a structural role, to those which catalyse biochemical reactions, to signalling, to mediating biological processes such as transcription and post-translational modifications [1]. The classical view of a protein is a stable globular three-dimensional structure, though this is limited in many respects. The function of the protein is highly dependent on its structure. For example, for proteins with a catalytic function, the structure generally contains a pocket in which amino acids with a catalytic role are exposed. Globular proteins generally contain multiple small pockets, some of which are suitable for small molecules (so-called ligands) to bind. These ligands may have a biological role, for example as co-factors or as inhibitors which modulate the functionality of the protein. The binding pockets are also of interest to drug designers, who may be able to design a small compound capable of binding to a protein and altering its activity and thus the biology of the organism.

Drug development is a long, complex and expensive process. In particular, the initial step of obtaining a list of candidate compounds is challenging. The chemical space is estimated to consist of 10^{60} compounds with potential pharmacological activity [2]; drug designers need to locate a tiny subspace which is capable of interacting with the target protein. Experimental screening, for example by means of protein-ligand binding assays or X-ray crystallography experiments, is fundamentally limited in scope; thus, the concept of virtual screening comes into play. Virtual screening entails the application of *in silico* experiments such as molecular dynamics simulation, or statistical approaches, to predict whether a given compound is likely to bind to a given target protein.

Another concept that has gained importance in the last couple of decades is fragment-based screening [3]. Here, a fragment refers to a small organic compound, which is generally not itself a feasible drug candidate, but can act as a substructure for a putative drug molecule. The process of elaborating fragment structures and combining them to form candidate compounds has been approached in several ways. The so-called “fragment network” approach treats compounds as nodes in a graph [4]; compounds which share common substructures or fragments are connected together. Recent publications also make use of generative neural networks to produce a list of compounds based on one or more input fragment molecules [5].

Once a list of compounds has been generated, either by fragment screening or some other method, the need arises to score, rank and filter them on some measure relating to their binding affinity and likely usefulness as potential drug molecules. There are various approaches which differ in computational complexity and resource demands, including both simple heuristics like Lipinski's rule of five [6] and more complex methods which make use of machine learning. Among the most accurate methods, though also the most computationally expensive, are free energy calculations based on molecular dynamics (MD) simulations. As a first step, if the system consists of a protein and a ligand, but the position of the ligand in the binding site is not already known, docking software is used in order to find a physically reasonable hypothesis for the complex structure. This structure can then be used as a basis for MD simulation. Molecular dynamics is a physical simulation method which is often applied to biochemical systems. In its simplest form, MD is based on classical Newtonian mechanics, treating atoms as masses and bonds as springs which connect them [7]. An initial set of forces are applied to the component atoms of the system, the changes in position and velocity over a short time step (typically 1-2 femtoseconds) is calculated, and the process is iterated to produce a so-called trajectory which describes the motion of the system on the atomic level. While the resulting trajectory is inherently of interest, as it can reveal important aspects of biomolecular mechanisms, a key motivation when using MD to study protein-ligand binding affinity is to calculate a value for the free energy of binding. This value relates to the proportion of time a molecule is likely to spend in the bound compared to the unbound state. There are several methods for deriving free energy methods, each with a different physical basis. One of the most common is the MMGBSA or MMPBSA method, which also requires relatively little simulation (and thus compute) time, although it is acknowledged to be less accurate than other methods [8]. Others include alchemical calculations and thermodynamic integration. Some methods make use of equilibrium MD, in which the component molecules of the system are allowed to move undisturbed, analogous to their *in vivo* behaviour, whereas other use non-equilibrium MD, in which an artificial force is introduced to perturb the system and provoke some interesting response which cannot be observed in an equilibrium simulation.

All the methods mentioned so far (fragment network, docking, MD, machine learning based approaches) are not mutually exclusive, but indeed often need to be combined and then applied in sequence or in parallel. Creating and executing such workflows creates new difficulties for the scientist; the methods require widely differing amounts and types of computational resources, and careful design is required to ensure that workflows make optimal use of the available resources and that the large amount of data is organised in a manner which facilitates later analysis. The more computationally demanding methods generally provide more accurate insight into the binding between the protein and the ligand. For example, an initial list of target compounds might be first filtered based on some statistical methods, followed by docking of the shortlist and further filtering based on some methodology for scoring the docked poses. The remaining compounds might then be filtered further via several MD-based free energy methods, each with increasing computational demands as well as increasing accuracy. Finally, the compounds which pass all the stages successfully might be purchased for experimental testing. At workflow complexity grows, requiring the execution of numerous of parallel and consecutive steps the scientist may choose to make use of a workflow management system to organise and run the analysis.

Several workflow management systems already exist for organising scientific workflows [9]; among the most prominent are Nextflow, Snakemake and Galaxy in the field of bioinformatics, and KNIME in the field of cheminformatics. Smaller, more specialist systems are constantly under development, such as Icolos, a Python-based tool for virtual screening of newly generated molecular structures [10]. These systems allow chaining multiple tools together into a single pipeline, which can then be executed via the command line (or in some cases, via a graphical interface). This has the advantage of convenience over the laborious execution of individual tools in sequence, and ensures no manual errors can creep in. For highly complex analyses employing tens of different software tools, which need to be repeated in a stepwise manner (for example, running MD simulations for hundreds of ligands against a single target protein), employing a workflow management system becomes the only feasible option, compared to the classic approach of a shell script.

Another challenge in molecular simulation is reproducibility. A fundamental principle for any scientific work, regardless whether experimental or theoretical, is that it should be possible for other scientists to critically assess the work by performing the same experiments, simulations and analyses themselves and thus verifying the results. The issue of reproducibility has received increased attention in recent years among scientists in general and there have been claims of a “crisis of reproducibility” [11]. Molecular simulation and computational chemistry is no exception, since small changes in starting coordinates, user choices during parameterisation and the choice of MD engine (including even the particular software version used) can have a substantial effect on the outcome of the simulation. Employing a workflow management system has the additional benefit of helping to combat this issue, since it can act as a “lab notebook” which records the details of all tools and workflows executed, together with parameters and all tool software versions. Thus, full reproducibility of all analyses and simulations is ensured for either the scientists themselves or others who wish to verify their work.

1.1 Thesis outline

The main focus of this thesis has been the development of a platform based on the workflow management system Galaxy for fragment-based virtual screening. This development entailed the creation of multiple Galaxy tools and workflows, packages for the Conda package manager, and Docker and Singularity containers. This platform can be used either through a graphical web-based interface or via the command line; the latter is a useful alternative for complex simulations or analyses which may require additional scripting. In order to more easily enable command-line usage, substantial contributions were made to Planemo and BioBlend, two Python libraries which allow Galaxy to be accessed directly via the application programming interface (API).

Two projects making of use of the developed tools and workflows were carried out; these have a scientific value of their own, but demonstrate the utility of the platform developed. Firstly, a study of the T4 lysozyme L99A mutant in complex with benzene was performed, using the dcTMD technique, as a model system for fragment-protein interaction. Secondly, virtual screening was carried out with the aim of identifying effective inhibitors of the main protease of the SARS-CoV virus; 40000 compounds were generated, using an initial set of 22 non-covalent crystallographic fragments, and screened using protein-ligand docking, MMGBSA calculations, and dcTMD simulations.

1.2 Publication list

- **Simon Bray**, John Chilton, Matthias Bernt, Nicola Soranzo, Marius van den Beek, Bérénice Batut, Helena Rasche, Martin Čech, Peter Cock, Björn Grüning, Anton Nekrutenko. Planemo: a command-line toolkit for developing, deploying, and executing scientific data analyses. *Genome Research* (under review), https://doi.org/10.1101/2022.03.13.483965

- **Simon Bray**, Victor Tänzel, Steffen Wolf. Ligand Unbinding Pathway and Mechanism Analysis Assisted by Machine Learning and Graph Methods. *Journal of Chemical Information and Modelling*, Volume 62, Issue 19, 29 September 2022, https://doi.org/10.1021/acs.jcim.2c00634

- **Simon Bray**, Tim Dudgeon, Rachael Skyner, Rolf Backofen, Björn Grüning, Frank von Delft. Galaxy workflows for fragment-based virtual screening: a case study on the SARS-CoV-2 main protease. *Journal of Cheminformatics*, Volume 14, Article number: 22, 12 April 2022, https://doi.org/10.1186/s13321-022-00588-6

- Wolfgang Maier, **Simon Bray**, Marius van den Beek, Dave Bouvier, Nathan Coraor, Milad Miladi, Babita Singh, Jordi Rambla De Argila, Dannon Baker, Nathan Roach, Simon Gladman, Frederik Coppens, Darren P. Martin, Andrew Lonie, Björn Grüning, Sergei L. Kosakovsky Pond, Anton Nekrutenko. Ready-to-use public infrastructure for global SARS-CoV-2 monitoring. *Nature Biotechnology*, Volume 39, pages 1178–1179, 29 September 2021, https://doi.org/10.1038/s41587-021-01069-1

- Qiang Gu, Anup Kumar, **Simon Bray**, Allison Creason, Alireza Khanteymoori, Vahid Jalili, Björn Grüning, Jeremy Goecks. Galaxy-ML: An accessible, reproducible, and scalable machine learning toolkit for biomedicine. *PLOS Computational Biology*. Volume 17, Issue 6, Article number: e1009014, 1 June 2021, https://doi.org/10.1371/journal.pcbi.1009014

- **Simon A. Bray**, Tharindu Senapathi, Christopher B. Barnett, Björn A. Grüning. Intuitive, reproducible high-throughput molecular dynamics in Galaxy: a tutorial. *Journal of Cheminformatics*, Volume 12, Article number: 54, 10 September 2020, https://doi.org/10.1186/s13321-020-00451-6

- Steffen Wolf, Benjamin Lickert, **Simon Bray**, Gerhard Stock. Multisecond ligand dissociation dynamics from atomistic simulations. *Nature Communications*, Volume 11, Article number: 2918, 10 June 2020, https://doi.org/10.1038/s41467-020-16655-1

- **Simon A. Bray**, Xavier Lucas, Anup Kumar, Björn A. Grüning. The Chemical-Toolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform. *Journal of Cheminformatics*, Volume 12, Article number: 40, 01 June 2020, https://doi.org/10.1186/s13321-020-00442-7

- Tharindu Senapathi, **Simon Bray**, Christopher B. Barnett, Björn Grüning, Kevin J. Naidoo. Biomolecular Reaction and Interaction Dynamics Global Environment (BRIDGE). *Bioinformatics*, Volume 35, Issue 18, pages 3508–3509, 15 September 2019, https://doi.org/10.1093/bioinformatics/btz107.

Background

2

2.1 Proteins

Proteins are fundamentally, on the microscopic level, the biological machinery of life, with a functionality ranging from catalysis (enzymes), carrying biochemical signals, to structural roles [1]. Proteins are polymers of amino acids, i.e. chains of amino acids connected by covalent chemical bonds [1]. Unlike synthetic polymers, such as polystyrene or polypropylene, the monomer units which make up the polymer are not identical, or randomly distributed, but precisely determined by a complex biochemical machinery, following instructions encoded in DNA. The monomer units in question are amino acids; 20 of them are commonly used in protein synthesis and are thus referred to as proteogenic amino-acids (Figure 2.1). The precise composition and sequence of the protein chain endows it with unique properties.

The instructions for biochemical protein synthesis are stored in the genome of each organism, generally in the form of deoxyribonucleic acid (DNA). The DNA is itself a polymer, made up of nucleotide subunits. There are four different nucleotides commonly used in nature: adenine, thymidine, cytidine, and guanidine. The regions of DNA which encode protein can be split into codons of three nucleotides each; each codon corresponds to a single amino acid. There are $4^3 = 64$ possible codons, which is more than enough to encode the 20 constitutive amino acids [1].

Two processes are necessary to extract the information encoded in the DNA and to use it to synthesise the encoded protein. The first is transcription, in which DNA is used as a template to synthesise a related molecule, ribonucleic acid (RNA), with an identical but complementary sequence. Transcription is performed by an enzyme named RNA polymerase. RNA is considerably less stable than DNA and functions as a short-term information store, carrying the genetic information from the DNA, which in complex, eukaryotic organisms is generally sequestered in a cell nucleus, to the ribosome, the site of protein synthesis. The second process is translation, in which the ribosome reads the RNA molecule codon by codon and iteratively appends the corresponding amino acid to the nascent protein chain. When synthesis is complete, the chain is released and is subjected to various post-translational modifications before it can perform its function in the cell.

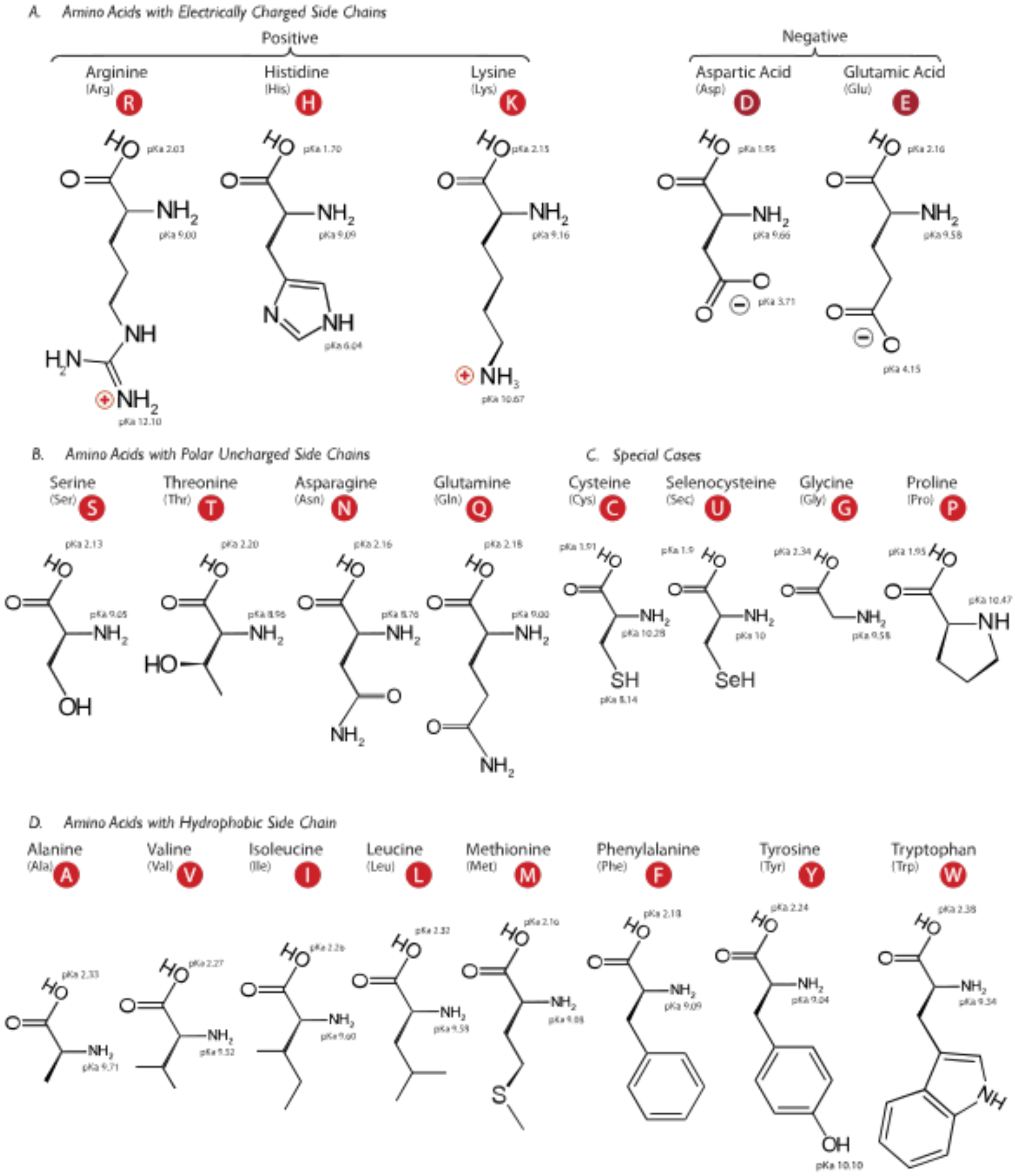


Fig. 2.1: Structures of the proteogenic amino acids. Information on the image source is provided in the List of Figures.

The essential factor which determines the function of a protein is the three-dimensional structure which it adopts during and after synthesis. This three-dimensional conformation (the tertiary structure) results deterministically from the protein sequence (the primary structure), with some influence from the cellular environment, such as post-translational modifications; despite this, prediction of 3D structure from 1D sequence is a highly challenging problem. The tertiary structure (Figure 2.2) results from the total sum of molecular interactions between the component amino acids, as well as the interactions between the amino acids and the surrounding solvent molecules. These interactions can vary in nature: they can be hydrogen bonds, hydrophobic interactions, van der Waals forces, salt bridges, or even covalent bonds, such as sulfide bonds [1].

Secondary structures occupy the level between the amino acid sequence and three-dimensional structure - they consist of helical, sheet-like and loop regions, which combine together to form the entire tertiary structure. In addition, a protein may be composed of more than one protein chain. In that case, the additional layer of protein structure is referred to as quaternary structure [1].

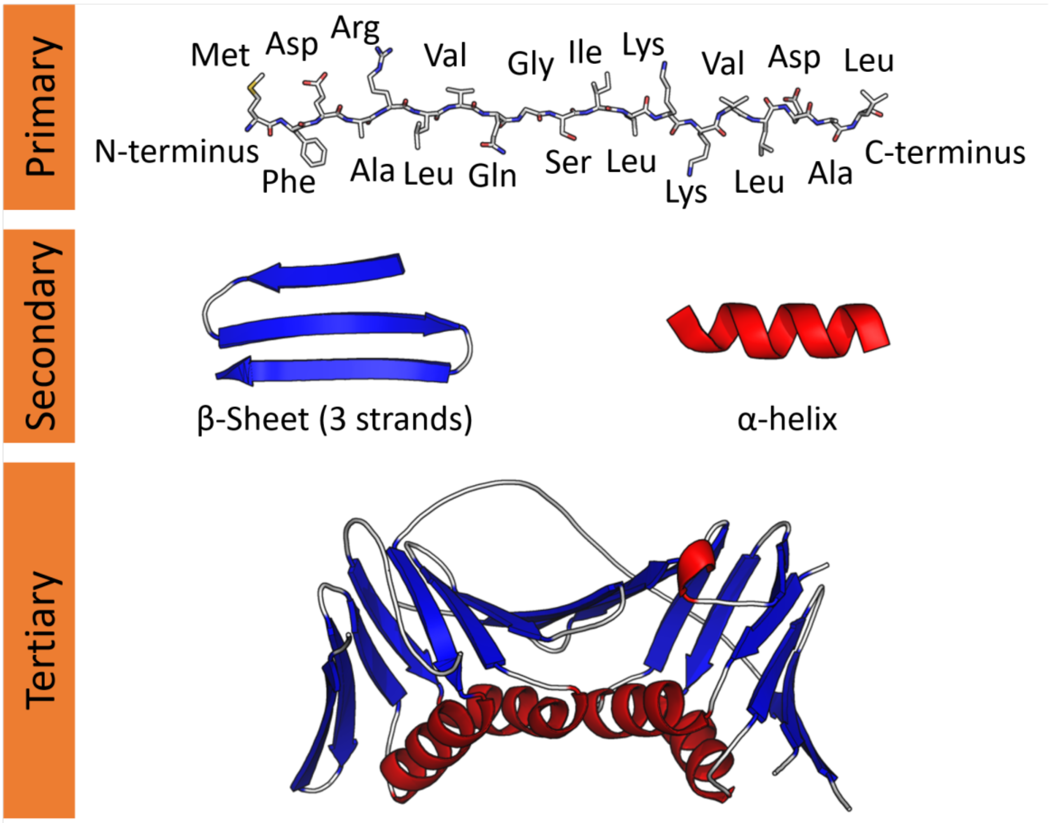


Fig. 2.2: Levels of protein structure. Information on the image source is provided in the List of Figures.

While the number of amino acids encoded directly in the genetic code is limited to twenty, they have diverse chemical properties. Some have aliphatic carbon chains, such as leucine or valine; some are aromatic, allowing pi-pi interactions, such as tryptophan; some are positively or negatively charged, depending on the pH. In addition, the properties of each amino acid are modulated by those surrounding it in the tertiary structure. For example, in certain hydrolytic enzymes, a serine residue, which is normally very difficult to deprotonate, can be rendered highly acidic due to the influence of neighbouring histidine and glutamine residues - the so-called catalytic triad [12].

Many, even a majority, of proteins do not have a defined three-dimensional structure; these are the so-called intrinsically disordered proteins [13]. Nonetheless, they possess important functionality; however, due to the difficulty of elucidating this functionality from the structure the attention of scientists has been heavily biassed towards globular proteins.

2.1.1 Experimental methods for protein structure determination

Due to the intrinsic difficulty of predicting three-dimensional structure from the amino acid sequence, experimental techniques for structure determination are essential. Several of these exist. The most important continues to be X-ray crystallography, which relies on overexpression and purification of the protein of interest, followed by crystallisation of the protein out of solution. The protein molecules in the crystal have a highly ordered position and orientation; as a result, when X-rays are fired at the crystal, they are deflected in a highly amplified manner to create signals which can be easily detected. The diffraction pattern thus created can be decoded to obtain the averaged three-dimensional structure [14].

Another method is nuclear magnetic resonance (NMR), a spectroscopic technique which allows measurement of local magnetic fields around atomic nuclei, from which interatomic distances and thus the entire 3D structure can be deduced [15]. NMR is the only technique which allows structural determination on the atomic level of proteins which do not possess a fixed three-dimensional structure, i.e. intrinsically disordered proteins. A third method is electron microscopy (EM), which makes use of the smaller frequency of electron radiation to achieve a much higher resolution than is possible with light microscopy, allowing imaging of the surface of individual protein molecules. However, while EM has gained much ground on X-ray crystallography in recent years, the achievable resolution still lags somewhat behind that of crystallography and NMR [14].

While purely computational prediction of the three-dimensional structure from the primary structure was previously considered an extremely difficult problem, indeed one of the major unsolved scientific problems [16], a major step forward was made in 2021 with the publication of AlphaFold2 [17], a machine learning method trained on experimental data and capable of predicting atomic coordinates from input protein sequences. The method consists of two main stages. Firstly, multiple sequence alignment from protein subsequences are used to provide structural information on evolutionarily related protein structures and to generate two matrices: one representing the multiple sequence alignment for those proteins, and one representing residue pairs. Secondly, these matrices are used to generate geometric (rotation and translation) attributes for each residue. The entire method is training in an end-to-end manner, rather than individually training network components as in previous versions of AlphaFold, and iterative refinement (dubbed recycling) using the whole network is also employed. The predicted structure is then relaxed by molecular dynamics simulation using the Amber force field [18] for gradient descent.

2.2 Drug design

Proteins are responsible for a huge range of biological functionality and thus their malfunctioning can quickly lead to disease. One of the aims of drug developers is to identify a protein involved in a particular disease and to design a small molecule which is capable of interacting chemically with it. On binding, this ligand alters the behaviour of the protein, for example by blocking its active site, or by forcing a change in conformation, so the protein activity is lowered (or alternatively, increased). The binding consists of interatomic interactions, similar in type to those responsible for the maintenance of the protein tertiary structure: hydrogen bonding, hydrophobic interactions, π-stacking, etc. In principle, a ligand which binds more quickly and tightly will be more effective at modifying the target protein's behaviour, and hence a more potent drug - hence the motivation behind computational investigations of ligand binding.

Experimentally, protein-ligand binding affinity is measured by means of binding assays. Commonly, fluorescent labelling of the ligands is employed, such that the behaviour of the fluorophore is modified depending on whether the ligand is in the bound or unbound state. In fluorescence polarisation assays, the extent to which the emitted light is depolarised correlates with the amount of time spent in the free unbound state [19]. A method which does not require fluorescent labelling of the ligand is surface plasmon resonance [20]; here, polarised light is reflected from a surface, on the reverse side of which protein molecules are immobilised, with free ligand able to bind. The extent to which binding occurs affects the angle at which the light is reflected - the surface plasmon resonance angle.

Drug development is a highly costly process with a very high failure rate in clinical trials [21]. Computational methods can contribute with the task of identifying and optimizing chemical structures with high binding probability or affinity. While the data obtained from binding assays and other experimental methods is much more valuable than computational studies, binding assays are limited due to the expense of synthesizing and testing a large number of compounds. Virtual screening - especially the kind of high-throughput screening with thousands or even millions of compounds which this work aims to facilitate - allows filtering out a subset of promising compounds which can then be subjected to experimental testing.

Computational approaches for drug design can be divided into two main categories. Ligand-based drug design involves extracting features (molecular descriptors) from the candidate compounds to make predictions about their binding affinity, based on knowledge of binding affinity obtained from other compounds [22]. Structure-based drug design makes use of three-dimensional structural information, including the protein structure and conformation [23]; methods include protein-ligand docking and molecular dynamics, which will be described in more detail below.

2.3 Fragment-based screening

The essence of fragment-based screening is that drug candidates are based on one or multiple substructures or “fragments”. These are typically responsible for a particular set of intermolecular interactions with the protein binding site, or confer a particular molecular property on the ligand. Drug candidates can thus be generated by taking a list of fragments as a basis and combining multiple fragments, or modifying a single fragment by appending various functional groups (Figure 2.3).

Fragments may be experimental entities; for example, in crystallographic fragment screening, protein crystals are soaked in solutions of various candidate compounds before performing diffraction experiments [3]. The resulting crystal structures show the positions of any bound fragments within the protein binding site. From this information, important binding interactions can be deduced, which is useful in designing lead compounds based on the structures of one or more crystallographic fragments. Another option is to deduce fragments by comparing known structures of known protein-ligand complexes and identifying common substructures and interactions.

A major advantage of fragment screening compared to high-throughput screening is that smaller compounds bind more indiscriminately and subpockets can be sampled more thoroughly. Thus, the overall amount of chemical information about the site gathered is higher. A disadvantage is the required sensitivity of the experimental methods; the binding of smaller compounds tends to be weaker and thus screening by a crystallographic or another method is significantly more challenging. Once a set of confirmed fragments has been assembled, however, it allows bottom-up design of drug candidates starting from the component functional groups, compared to the blinder approach adopted by high-throughput screening.

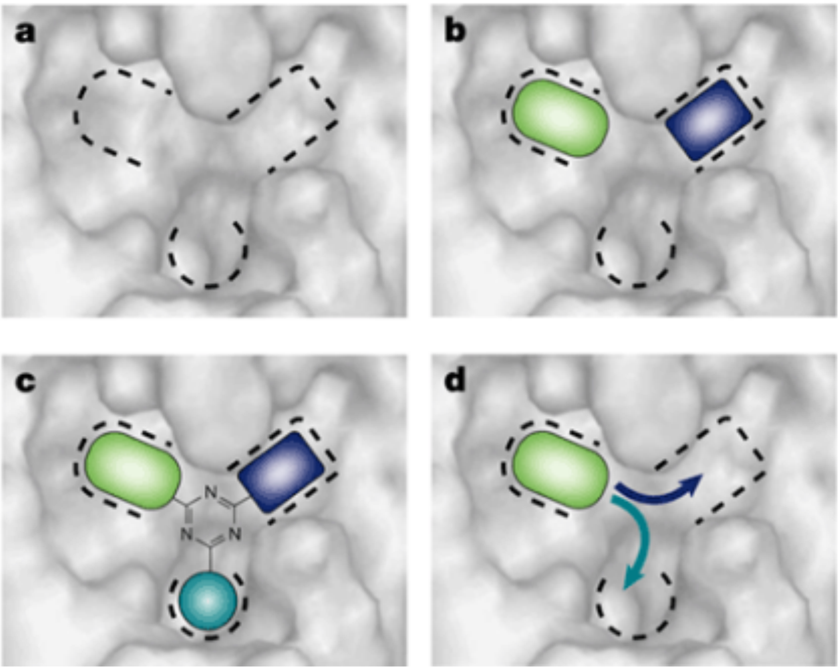


Fig. 2.3: An illustration how a ligand can be designed using a fragment-based approach. A number of subpockets are identified within the binding site, and fragment screening identifies molecular fragments that bind into each. A compound is then designed which either incorporates multiple fragments as substructures, or extends a single fragment into the neighboring subpockets. Information on the image source is provided in the List of Figures.

2.4 Free energy

Free energy (also known as Gibbs free energy) is an essential concept in molecular biophysics and computational chemistry. The change in free energy resulting from a chemical or physical process is defined as the maximum non-expansion work performed. A negative change in free energy ($\Delta G < 0$) indicates that a process is spontaneous at constant temperature and pressure. The free energy has two components, the enthalpy H and the entropy S:

$$G = H - TS \tag{2.1}$$

where T is the temperature. The enthalpy corresponds to the energy supplied to the system from the environment as heat, whereas the entropy is often informally related to the “disorder” associated with a process. Related to protein-ligand binding, the bound state generally has a lower entropy than the unbound state, where the two components diffuse in solution free from one another. The bound state also has a lower enthalpy, as heat has to be supplied to the system to break the intermolecular bonds binding the ligand to the protein binding site. Thus, the enthalpic and entropic components tend to counteract one another; which of the two components is stronger depends on the temperature of the system.

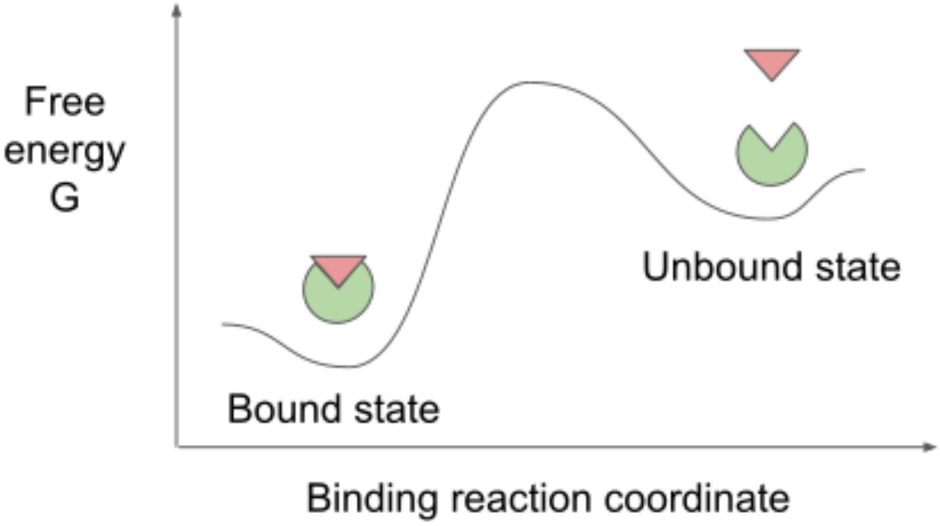


Fig. 2.4: A simple illustration of a free energy landscape for protein ligand association and dissociation. The bound state has in general a lower free energy than the unbound state (assuming that binding is energetically favoured) but there is also a kinetic barrier between the two states with a higher free energy.

The free energy can also be related to the equilibrium between the two states by the following equation:

$$\Delta G = -RT \ln K \tag{2.2}$$

where K is the equilibrium constant:

$$K = \frac{[PL]}{[P][L]} \tag{2.3}$$

where $[P]$, $[L]$ and $[PL]$ represent the concentrations of protein, ligand and complex respectively.

Thus, the free energy tells us something about the proportion between the number of ligand and protein molecules in the bound compared to the unbound state; the more negative the free energy, the more are in the bound state. However, the free energy tells us nothing about the rate at which binding and unbinding occurs; it only provides information about the proportion of time spent in each state and thus the same value for free energy is consistent with both rapid and slow transfer between the two states. In relation to Figure 2.4, ΔG is represented by the difference between the bound and unbound states, whereas the rate of transfer between the two is represented by the height of the barrier between the two states; there is not necessarily any relation between the two. This is the essential difference between thermodynamics and kinetics; both require consideration when designing potential drug molecules.

A concept which has attracted attention in recent years is drug residence time [24]. This posits that a key factor affecting the effectiveness of a putative drug is average length of time which the drug remains bound prior to dissociation, as the longer the drug remains bound, the longer the protein's behaviour will be modified and thus the more effective the drug is. The residence time is related to the kinetics of the drug, as it is inversely proportional to the rate of ligand dissociation:

$$\tau = \frac{1}{k_{diss}} \tag{2.4}$$

2.5 Computational methods

2.5.1 Cheminformatics

Cheminformatics refers to the use of techniques from information technology to solve problems related to chemical structures. An example of a problem in cheminformatics is a so-called similarity search - given two particular chemical structures, how can the level of similarity between them be evaluated? The main approach employed is fingerprinting, for which numerous implementations exist [25]. The chemical properties of a compound are encoded as a bitstring, with each bit representing the presence or absence of a particular property (Figure 2.5). The bitstrings, or their hashes, can then be compared.

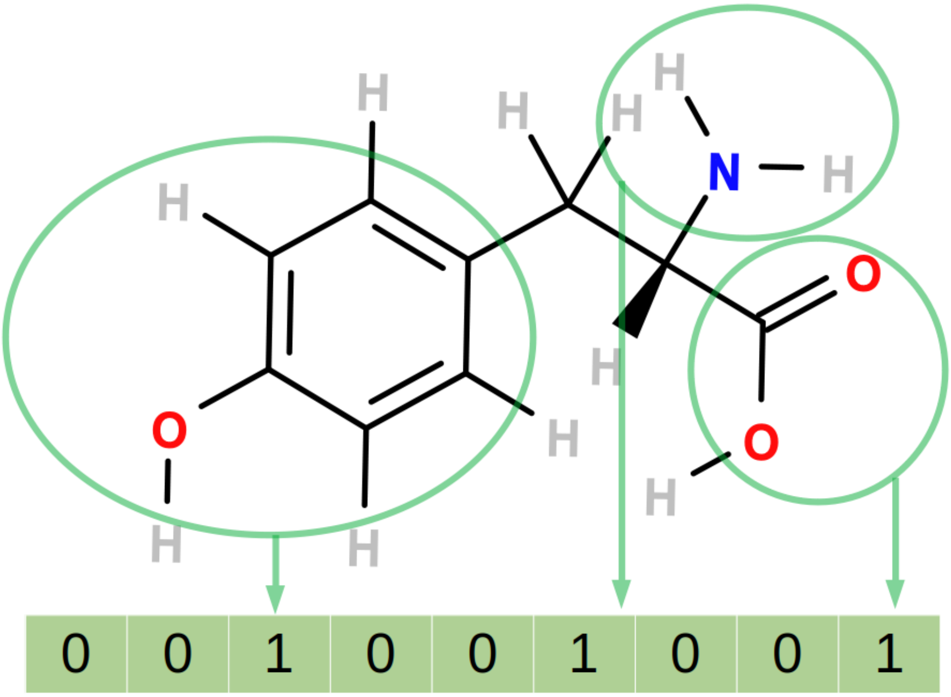


Fig. 2.5: An illustration of a simple fingerprinting system. Each digit of the bitstring represents the presence or absence of a feature in the molecule. In the example above, the presence of phenyl, amine and carboxylic acid groups in the compound are encoded. Information on the image source is provided in the List of Figures.

An important concept in cheminformatics is chemical space, closely related to the fingerprinting concept. This assumes that chemical compounds can be described by a number of properties, which form the dimensions of the chemical space - these dimensions could also be composite, e.g. derived by a dimensionality reduction technique such as principal component analysis from the raw properties. Molecules which are close in the chemical space are thus chemically similar. Despite the vastness of the total chemical space, the proportion of molecules which are actually biologically relevant is comparatively minute; the identification of these relevant molecules is the major challenge of cheminformatics [26]. As an example, a chemist might identify particular regions of the chemical space which are promising for a certain application and then search them in more detail for candidate compounds. Alternatively, the chemist might wish to create a compound library which samples the chemical space as evenly as possible; here, the task would be to identify sparsely occupied regions in the library and then to fill them with additional compounds.

Several ideas from the field of cheminformatics can be applied to the problem of predicting protein-ligand binding affinity. Experimental data (for example, from a binding assay) provides information about the kind of ligands which are likely to bind effectively to a particular protein. Statistical methods (e.g. machine learning [27]) can be used to extract this information as a statistical model which can then be applied to new, experimentally untested compounds to predict the likelihood or strength of their binding affinity. Such approaches are traditionally known as quantitative structure-activity or quantitative structure-property relationships (QSAR or QSPR) [28, 29].

Cheminformatics is characterised by a very wide range of file formats [30]. Some formats depict only the graph of the molecular structure, such as SMILES or InChI, whereas others, such as SDF, provide three-dimensional coordinates for all the atomic positions [31]. As a result, any flexible software which aims to be used by cheminformaticians must be able to deal with the range of datatypes used in the field. For example, a scientist might retrieve a list of SMILES strings from a chemical database to obtain a compound library for investigation, but need to generate three-dimensional structures (conformers) in order to assess the molecular interactions with a target protein, requiring a conversion to SDF or a similar format. One of the most commonly used command-line tools for converting between file formats is OpenBabel [31]. In general, there is more than a single possible three-dimensional structure producible from the molecular graph. Often, it is desirable to exhaustively enumerate all such structures for virtual screening projects. Gypsum-DL is a recently published software which focuses on generating small-molecule libraries for this purpose [32, 33]; this entails the generation of three-dimensional coordinates from flat molecular graphs such as SMILES strings, but also enumeration of various tautomers, isomers, and conformers.

2.5.2 Protein-ligand docking

Protein-ligand docking entails generation of hypothetical three-dimensional structures of the protein and ligand in complex. Generally, most docking software employs an iterative approach to optimise the coordinates of the ligand in the active site. For example, the rDock software employs a genetic algorithm, in which the chromosome consists of the ligand centre of mass, the orientation, the rotatable dihedral angles of the ligand, and the rotatable dihedral angles of the receptor. These are mutated by a random distance or angle at each step. The initial state is selected by placing the three-dimensional conformer of the ligand at a random grid point within the volume defined for docking. After each mutation, a “score” is calculated and if this ceases to decrease, convergence is considered to have been reached [34]. A standard scoring function, combining intermolecular interactions and intramolecular interactions for both protein and ligand, as well as user-defined restraint functions, is used as default; however, many docking programs give users the option to make use of custom scoring functions.

In general, protein-ligand docking is recognised to produce realistic poses, which correspond to experimental reality. Nonetheless, they are considerably weaker at evaluating the goodness of a particular pose, or ranking poses based on the likelihood of corresponding to experiment. Thus, a sensible view of docking is as a hypothesis generator, which produces multiple potential binding poses for a given protein-ligand pair; the process of verifying which of these is closest to the truth should be performed by a different method (for example, comparison with experimental data, if available). If docking is followed by MD simulations for free energy calculations, the identification of the most accurate docking pose is essential for obtaining high-quality free energy results [35], which indicates the importance of developing methods capable of confirming or rejecting proposed poses. One possibility is validation using experimental data (such as known crystallographic structures), or alternatively, MD simulations can be used to verify or refine hypothetical docking poses [36]. Recently, techniques based on deep learning, such as GNINA [37] and TransFS [38] have also been proposed to solve the problem of rescoring. A major challenge for these methods is obtaining good training data, in particular negative data.

2.5.3 Molecular dynamics

Molecular dynamics (MD) is a molecular simulation technique in which the system (protein, ligand and solvent) are modelled using Newtonian mechanics. While these systems are more accurately described by quantum mechanics, traditional molecular dynamics is a compromise solution to avoid the very high computational costs of quantum mechanics-based simulation; for example, compute time for Hartree-Fock simulations increases at a rate of k^4, where k is the number of electrons in the system [39], limiting its usefulness to very small molecules. At the beginning of an MD simulation, random forces are assigned to all atoms, which are modelled as masses connected by springs (covalent bonds). The movement of all atoms is calculated over the course of a time step and the process is then iterated to generate a trajectory which describes the evolution of system over time. Due to the high computational cost of MD simulations, MD simulations cannot be realistically used to model larger biological entities such as entire cells [40]; they are used primarily to focus on the behaviour or interaction of one or two molecules (not counting solvent), for example ligand binding or debinding to a protein, or changes in protein conformation [41].

A MD simulation proceeds by solving the Newtonian equations of motion for a system iteratively, which results in a trajectory describing the motion of the component atoms during the course of the simulation:

$$m_i\ddot{r}_i = f_i, f_i = \frac{dU_i}{dr_i} \tag{2.5}$$

where f_i are the forces acting on atom i, U the potential energy, and r the atomic coordinates. A force field is used to calculate U, which has both an intermolecular and intramolecular component:

$$U = U_{inter} + U_{intra} \tag{2.6}$$

The bonds, bond angles and torsion angles within the molecules which make up the system all exert an influence on U_{intra}. The force field determines how exactly U_{intra} is calculated from these variables. U_{inter} is determined by electrostatic and van der Waals interactions, which are modelled by Coulomb and Lennard-Jones potentials respectively [7] [42].

MD simulations require application of multiple tools in sequence. Firstly, both the protein and ligand require parameterisation for the particular force field which has been chosen for simulation. Secondly, a simulation box is defined and filled with solvent. Thirdly, equilibration of the system is performed, before the production simulation can begin. In addition, MD simulations are often performed as ensembles, i.e. the same simulation is repeated multiple times. The motivation here is often statistical robustness; carrying out a free energy calculation based on a single MD simulation will have a large error associated, which can be reduced by repeating the simulation multiple times and calculating an ensemble average [8]. Some kinds of free energy calculations inherently require multiple simulations, for example methods based on the Jarzynski equality, as will be discussed. Thus, the process of running MD calculations can be quite complex, requiring the execution of multiple steps both in series and in parallel.

From a technical point of view, MD has a heavy demand for computational resources, although it benefits greatly from the advent of general purpose GPU hardware; typically, the nonbonded force calculations, the most computationally demanding, are performed on the GPU. While the exact requirements vary massively based on the size of the system and the specific hardware used, GPU/CPU compute times on the order of hours may be required to calculate one simulation of 1 ns in length for a protein system. It should be noted that biochemical transitions are occurring on a timescale far longer than a nanosecond; the fastest enzymes have a turnover rate of 1 microsecond, while protein folding occurs on a millisecond or second timescale. Conformational changes, such as domain motion, may likewise last microseconds to milliseconds [43]. The computational cost means that the maximum simulation time for which MD can be performed is limited; the maximum time period that can be reached is on a level of milliseconds, and even then, specialist computing resources such as Anton [44] and Folding@home [45] are required.

Nonetheless, many biochemical and biophysical process take place on an even longer time scale than this; in particular, ligand residence times can be on the order of minutes or even hours. As a result, non-equilibrium simulations are often utilised, in which a bias is introduced to force the system artificially to cross an otherwise impassable kinetic barrier [46]. One of the currently most popular techniques is metadynamics; here, as the system moves over the free energy landscape, biassing potentials are added periodically to discourage the system from returning to previously visited states. Once all states have been sampled, the sum of the biassing potentials is inverted to reconstruct the free energy landscape [47]. Another example is random acceleration molecular dynamics (RAMD), in which an artificial constraint force is simply applied to the ligand with random direction, to encourage it to depart from the binding site [48]. Scaled (or "smooth potential") molecular dynamics (SMD) is another alternative [49]; here, the potential energy term is scaled down by a factor, which has the consequence of reducing the kinetic barriers for the system, at the cost of losing some detail of the free energy landscape [50]. Steered molecular dynamics involves the application of an external force at a constant velocity to one or more atoms in the system; the original study sought to mimic the effect of an atomic force microscope cantilever [51]. An example of a method which makes use of steered molecular dynamics is dynamic undocking (DUck), which uses the external force to break an identified key intermolecular bond and reach a "quasi-bound" state, where the work profile reaches a maximum. This work W_{QB} required to travel from the initial to the quasi-bound state was demonstrated experimentally to relate to binding affinity [52].

A technique which in some ways is conceptually similar to dynamic undocking is dissipation-corrected targeted molecular dynamics [53]. It makes use of targeted molecular dynamics, a technique similar to steered molecular dynamics, in that a force is used to enforce a change in a structure. Unlike steered molecular dynamics, however, in which the force applied obeys Hooke's Law, this constraint force is constant over the whole simulation, regardless of the size of the energy barrier; thus, the energy landscape is sampled completely evenly [54]. Like dynamic undocking, dcTMD uses the force to pull the ligand away from its initial position; in contrast, it requires not a single simulation but an ensemble of TMD simulations. The Jarzynski equality is then applied to this ensemble to derive equilibrium free energy and friction profiles. dcTMD thus depends on a theoretical physical justification, in contrast to dynamic undocking, which is rationalised in an empirical manner. The free energy and friction profiles thus generated represent a coarse graining of the system - they represent reduction of the system's dimensionality from $3N$ (where N is the number of atoms) to two. These profiles can then be used to run simulations based on the Langevin equation, which requires substantially fewer compute resources than MD simulations and thus can provide kinetic information based on a long simulation time, ranging even to multiple seconds.

An alternative method to circumvent the gap between the timescales of biochemical processes and MD simulations is the so-called supervised molecular dynamics (SuMD) [55], which avoids the use of a biasing force. Multiple short simulations are run in sequence; after each, progress towards a target (for example, ligand unbinding) is monitored, and if not sufficient, the simulation is restarted with random reassignment of atomic velocities. This allows a path through the energy landscape to be found efficiently, without employing an unphysical external force as used in SMD or TMD. Nonetheless, SuMD is more effective at uncovering molecular mechanisms of unbinding than providing quantitative kinetic measurements.

2.6 Galaxy

The wide range of tools which are applied during the course of virtual screening means that using a workflow management system is helpful to create complex workflows, execute them, and monitor the results. In this thesis, the workflow management system Galaxy was used extensively [56].

Galaxy provides a web-based graphical environment in which scientific software can be executed. Wrapper files map inputs, parameters and outputs between the web interface and the command line, so that after a user launches a tool in the graphical interface, a corresponding command is constructed by Galaxy. The command is then executed on a remote server; to ensure reproducibility, this is done in a separate environment, insulated from the rest of the server. This environment may be, for example, a Conda [57, 58] environment, or a Singularity [59] container. This ensures that the results of simulations and analyses remain the same, if analyses are repeated, as identical software builds are used, together with identical tool versions for all dependencies. Another advantage of Galaxy from the user's point of view is that it takes care that all executed software is allocated appropriate resources; for example, MD jobs will be assigned to a node with GPUs, cheminformatics tools which process a very large number of molecules are assigned a high amount of memory, and docking jobs which can be highly parallelised are assigned to nodes with multiple CPUs.

While the ability to execute jobs via a graphical, web-based browser can be convenient, especially as it allows easy sharing of simulation and data analyses [60], the main advantage of Galaxy for the computational chemist is the fact that they can compose complex workflows in the Galaxy interface and execute them. For example, a workflow can be assembled which docks compounds into a protein binding site, sorts and filters them, and runs the multiple steps required for an MD simulation. Galaxy workflows can also deal with a large amount of input data, using Galaxy's collection feature, which allows grouping of related datasets; Galaxy tools can be run on collections just like individual datasets, triggering a separate job for each component dataset of the collection. This makes a high level of parallelisation possible. Another use of collections is to run ensembles of simulations; the user can specify an integer value upon execution and a collection of that size will be created, with a separate MD simulation run for each component.

While Galaxy provides a graphical interface, it can often be useful to execute workflows via the command line, for example if a workflow needs to be executed a large number of times. Another use-case could be optimisation of workflow parameters; here, the same workflow could be executed multiple times, varying the parameters of the various component tools slightly each time. The outcomes of the runs can then be compared and a decision made which combination of parameters is preferred for the workflow. For these kinds of use-cases, Galaxy provides access directly to the backend via a REST API, allowing users to circumvent the graphical frontend and execute tools and workflows programmatically [61]. As the scale of analyses increases, the demand for such programmatic solutions also increases. Thus, there is a need for user-friendly, command-line interfaces which can be used to scale up complex analyses.

2.6.1 Conda

Conda is a package manager associated with the Python ecosystem, but capable of packaging software written in any language. It doubles as a manager of virtual environments, into which Conda packages can be installed. For scientific software, where reproducibility has a very high value, Conda is an extremely useful tool, as it allows managing of packages on the level of software versions and builds. If a user runs a Galaxy tool with the same version twice, or on two different servers, even several years apart, they can therefore be confident that the software being used, including all dependencies, is identical. Other advantages of Conda are its language-agnostic nature, the installation of binaries without compilation, and that installation does not require any special user permissions.

As part of this thesis, several widely-used computational chemistry software packages were integrated and maintained into the conda-forge [58] and Bioconda [57] channels, two open-source community-run repositories for the development of Conda packages. Conda-forge is general purpose and is the biggest channel by number of packages, while Bioconda has a more narrow focus on bioinformatics software.

2.6.2 BioContainers

The BioContainers project is developed in parallel to the Bioconda project, with the aim of increasing the reliability and reproducibility of research software even further [62]. This is achieved by containerisation, a technology first used in cloud computing; containerisation involves the creation of an isolated computational environment in which all required software is preinstalled and which cannot interact with the host system. While the level of isolation is not as complete as for a virtual machine, as the container still shares the underlying operating system with the host machine, it is considerably improved compared to a Conda environment. Containerisation is a necessary component of the software stack supporting workflow management systems such as Nextflow or Galaxy, as it allows straightforward installation of the numerous software dependencies and environments required for a complex scientific workflow [63]. The BioContainers project supports two different types of containers: Docker and Singularity. For every individual Bioconda package, as well as every combination of packages required by a Galaxy tool, Docker and Singularity containers are automatically built and stored, the latter by the Galaxy project in a shared Cern Virtual Machine File System (CVMFS) [64]. Once built, containers are distributed via the BioContainers Registry [65] with a RESTful API, allowing software developers to access the Registry programmatically.

3 The ChemicalToolbox: computational chemistry in Galaxy

This chapter summarises the work originally described in the following publications:

- **Simon A. Bray**, Tharindu Senapathi, Christopher B. Barnett, Björn A. Grüning. Intuitive, reproducible high-throughput molecular dynamics in Galaxy: a tutorial. *Journal of Cheminformatics*, Volume 12, Article number: 54, 10 September 2020, https://doi.org/10.1186/s13321-020-00451-6
- **Simon A. Bray**, Xavier Lucas, Anup Kumar, Björn A. Grüning. The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform. *Journal of Cheminformatics*, Volume 12, Article number: 40, 01 June 2020, https://doi.org/10.1186/s13321-020-00442-7
- Tharindu Senapathi, **Simon Bray**, Christopher B. Barnett, Björn Grüning, Kevin J. Naidoo. Biomolecular Reaction and Interaction Dynamics Global Environment (BRIDGE). *Bioinformatics*, Volume 35, Issue 18, pages 3508–3509, 15 September 2019, https://doi.org/10.1093/bioinformatics/btz107.

3.1 Introduction

Cheminformatics and computational chemistry are complex fields utilizing a wide variety of different tools, but data analysis and simulation is still done in a more or less *ad hoc* fashion, relying heavily on throwaway bash scripts or Python notebooks, which are often not reusable or reproducible, even by the original data analyst. Data analysis is often performed on both local environments and on a HPC cluster, and software installation itself is generally installed by a mixture of methods, including compilation by either the user or a system administrator, or via various package managers, with an emphasis on quick solutions rather than following best practices. As a result, much time is wasted dealing with dependencies and many analyses are, realistically speaking, impossible to reproduce. A few efforts have been made to improve the situation in recent years - examples include the Conda channel Omnia [66], which distributes binaries of the OpenMM simulation software [67] and related packages, analogous to the role of Bioconda in the bioinformatics community, and the BioExcel initiative, which unites several leading software packages in the field of molecular simulation, including GROMACS [68] for MD, HADDOCK [69] for docking, and PMX [70] for free energy calculation. One result has been the publication of the biobb package [71] for interoperable computational chemistry analysis. In particular, an emphasis has been laid on engaging with the Common Workflow Language community [72].

By contrast, the bioinformatics community has also faced similar challenges to those described above and made some progress towards solutions, including the Bioconda and BioContainers projects for distributing software and workflow management systems such as Snakemake, Nextflow and Galaxy. The motivation for the work described in this chapter is to propose some solutions to these problems for the field of computational chemistry, based on the solutions already applied by bioinformaticians, in particular the Galaxy platform. Galaxy is a data analysis platform widely used in bioinformatics, but less so in the fields of cheminformatics, computational chemistry and biophysics. In cheminformatics, the most prominent workflow management system employed is KNIME, whereas use of such a framework for organising and executing molecular simulations is currently rare. As part of this thesis, much work has gone into developing tools to make Galaxy a useful contender for developing and running cheminformatics and simulation workflows. Numerous contributions have also been made to neighboring ecosystems, for example Bioconda and BioContainers.

3.2 Methods

3.2.1 ChemicalToolbox

As a first step, work was done to create a comprehensive platform for cheminformatics and computational chemistry analysis based on the Galaxy system. A selection of tools were integrated, and existing tools were also updated and modified where necessary. The result was the creation of the ChemicalToolbox, an open, accessible webserver for cheminformatics. Unlike the typical webserver, however, for which the software is executed in an unknown environment and perhaps not even published for the user to use independently, the tools are completely reproducible; if desired, the user can install the tools onto a local Galaxy server and continue their work, or just install the Conda packages or containers if they prefer to work in a command line environment.

Firstly, tools for downloading chemical structures from public databases such as PubChem [73], ChEMBL [74], and ZINC [75] were added, and a workflow published which downloaded all structures from all of these databases, standardising and removing duplicates. A workflow was also created and published for the "hole filling" problem, which is encountered when a compound library is used in which compounds are unevenly distributed through the chemical space. In other words, there are "holes" in the chemical space where only very few compounds are located. The workflow proceeded by calculating fingerprints for all input molecules, identifying these sparse regions by clustering, and "topping-up" by selecting only small clusters and adding compounds from PubChem which are chemically similar to them.

Once a satisfactory initial compound library has been created, various tools provide a range of cheminformatics functionality for further analysis. Apart from the fingerprinting tools already mentioned, users may wish to interconvert between various chemical formats, generate three-dimensional conformers for a ligand based on a chemical graph, or generate charge forms at a particular pH value. To enable this, tools were written leveraging the open-source RDKit and OpenBabel packages. For the next stage in a typical virtual screening project, multiple tools for molecular modelling are also provided. In particular, tools for the open-source docking software AutoDock Vina and rDock have been incorporated into the ChemicalToolbox. These tools parallelise the command-line software, to allow rapid processing of compound libraries containing multiple chemical structures. Prior to docking, tools based on RDKit and fpocket can be used to identify potential protein pockets or binding sites. Typical workflows for such projects, based on both rDock and AutoDock Vina, have been published.

As part of this thesis, contributions were also made to Galaxy-ML, a Galaxy-based workbench for machine learning [76]. A workflow was created combining Galaxy-ML and ChemicalToolbox tools to perform QSAR-like analyses. Binding assay data can be downloaded from PubChem and submitted to this workflow; chemical descriptors are calculated for each compound using RDKit and a classification model is trained to assign the labels "active" or "inactive" based on these descriptors.

3.2.2 BRIDGE

In addition to the ChemicalToolbox, a more specialised toolkit for molecular dynamics simulation and analysis was developed in collaboration with partners at the University of Cape Town in South Africa. The basis of this toolkit is the GROMACS [68], CHARMM [77] and NAMD [78] molecular dynamics software, and analysis tools based on MDAnalysis [79] and Bio3d [80]. My contribution to the paper was the development of the GROMACS tools. GROMACS is a highly-performant MD code and is crucially open-source, allowing deployment of the Galaxy tools developed onto public servers.

GROMACS is a complex piece of software; its functionality is split over 99 subcommands, as of the 2022 release [81], and it has been under development for over thirty years (since 1991). Therefore, the process of wrapping it as for use in Galaxy required some design choices to be made. Only a subset of the functionality, sufficient for setting up and running MD simulations, was incorporated. Some of the subcommands were converted directly into Galaxy tools, e.g. `solvate` as a tool for solvation, `pdb2gmx` as a tool for parameterizing PDB files prior to simulation. In some cases, however, a direct mapping between GROMACS commands and Galaxy tools proved to be unintuitive. For example, running MD simulations in GROMACS, assuming parameterisation is complete and a valid set of GROMACS files is available, requires two commands: the `grompp` and `mdrun` subcommands. The former acts as a preprocessor which reads all input files, including topology, structure and the MDP file which specifies a configuration for the simulation, and records all data as a single binary file. This is then used as input for the second command, which actually runs the simulation. For the Galaxy implementation, the decision was made to combine both into a single tool. Thus, a user can run MD simulations on Galaxy without even being aware of this implementation detail chosen by the GROMACS developers.

For the original publication, tools for simulation setup, solvation, energy minimisation, NPT and NVT equilibration and production simulation were written. Development continued in the years after the original publication with the addition of more specialised tools: merging GROMACS topologies, calculating energy components from MD simulations, modifying the simulation box, generating GROMACS index files for molecular groups, generating restraints for simulations, and manipulating GROMACS trajectories.

3.3 Results

3.3.1 Workflows

A selection of workflows were developed and described in the ChemicalToolbox paper [82]. These are discussed in brief again below.

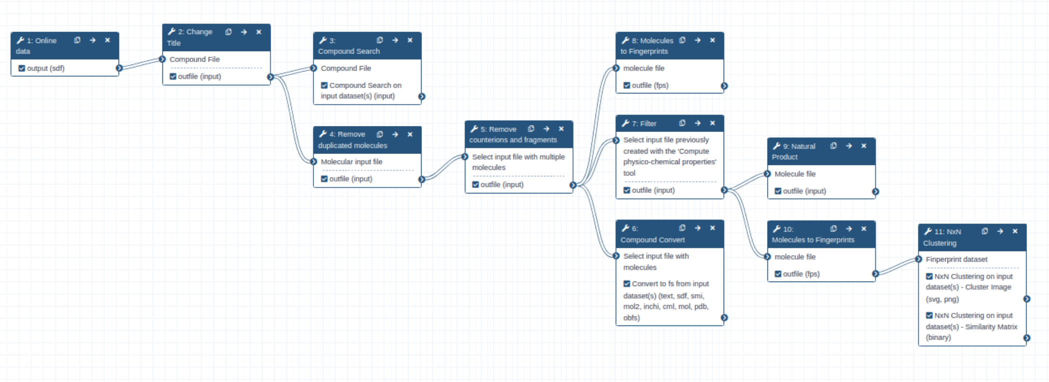

1: Online data
☑ output (sdf)
2: Change Title
Compound File
☑ outfile (input)
3: Compound Search
Compound File
☑ Compound Search on input dataset(s) (input)
4: Remove duplicated molecules
Molecular input file
☑ outfile (input)
5: Remove counterions and fragments
Select input file with multiple molecules
☑ outfile (input)
8: Molecules to Fingerprints
molecule file
☑ outfile (fps)
7: Filter
Select input file previously created with the 'Compute physico-chemical properties' tool
☑ outfile (input)
6: Compound Convert
Select input file with molecules
☑ Convert to fs from input dataset(s) (text, sdf, smi, mol2, inchi, cml, mol, pdb, obfs)
9: Natural Product
Molecule file
☑ outfile (input)
10: Molecules to Fingerprints
molecule file
☑ outfile (fps)
11: NxN Clustering
Finperprint dataset
☑ NxN Clustering on input dataset(s) - Cluster Image (svg, png)
☑ NxN Clustering on input dataset(s) - Similarity Matrix (binary)

Fig. 3.1: The protein-ligand docking workflow described below, as viewed in the Galaxy workflow editor. For clarity, simplified schematics of the workflows are provided here.

Data collection

There are a huge number of chemical structures stored in different locations online - some of the most well-known resources include PubChem, ChEMBL and ZINC, as well as many smaller, more specialised databases such as Drugbank [83]. A workflow was developed which accesses several of the largest of these resources and downloads all available compounds, before standardizing (e.g. at a constant pH, or to remove tautomers) and removing any duplicated structures (Figure 3.2). This provides an extensive list of compounds which users can make use of for further cheminformatics analysis, e.g. filtering for compounds containing a specific substructure.

Hole filling

Often, chemists and cheminformaticians face the issue of uneven occupation of the chemical space, a concept introduced in the Background section. This may be, for example, due to lack of synthetic routes to reaching certain chemical structures, leading to sparsely occupied areas in the chemical space. Nonetheless, it may well

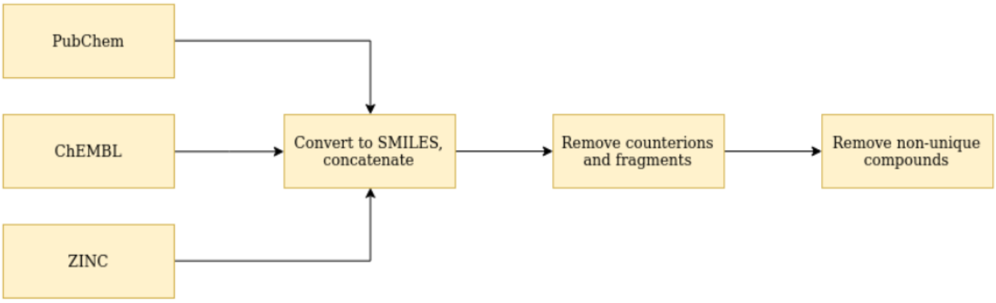

PubChem
ChEMBL
ZINC
Convert to SMILES, concatenate
Remove counterions and fragments
Remove non-unique compounds

Fig. 3.2: Simplified schematic of the data collection workflow. Information on the image source is provided in the List of Figures.

be undesirable that these regions are neglected in the data analysis. A workflow (Figure 3.3) is proposed for “hole filling”, entailing calculation of fingerprints for all the molecules in the Therapeutic Target Database (TTD) [84], followed by clustering according to the Taylor-Butina algorithm [85]. Singletons in the dataset are identified as candidates representing “holes” in the chemical space, which can be “filled” in the subsequent step; here, the PubChem database is searched for compounds which are close in the chemical space to each of the identified singletons and these compounds added to the TTD database.

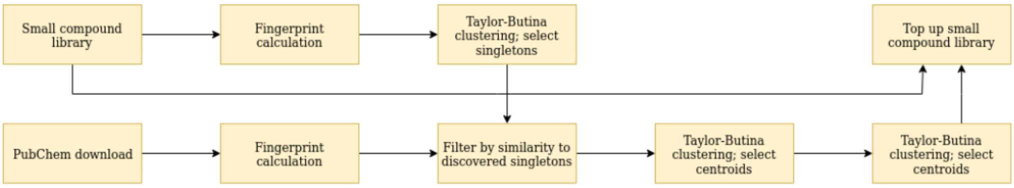

Small compound library
Fingerprint calculation
Taylor-Butina clustering; select singletons
Top up small compound library
PubChem download
Fingerprint calculation
Filter by similarity to discovered singletons
Taylor-Butina clustering; select centroids
Taylor-Butina clustering; select centroids

Fig. 3.3: Simplified schematic of the hole filling workflow. Information on the image source is provided in the List of Figures.

Protein-ligand docking

The third workflow (Figure 3.4) builds on Galaxy tools for protein-ligand docking, using the AutoDock Vina [86] software. The ChemicalToolbox also provides access to the rDock software [87] as an alternative, as well as fpocket [88], which can be used to locate protein pockets prior to docking. The workflow takes a PDB file, and optionally, a compound library in SMILES format; if the latter is not provided, a search is made of the ChEMBL database for structurally similar compounds to any ligand present in the PDB file. The files are then prepared for docking, generating

3D structures for the compounds to be docked, converting the receptor PDB file to PDBQT format, and locating a box surrounding the pocket into which the compounds should been docked. Docking itself is then performed, and the results presented in SDF as well as in tabular format.

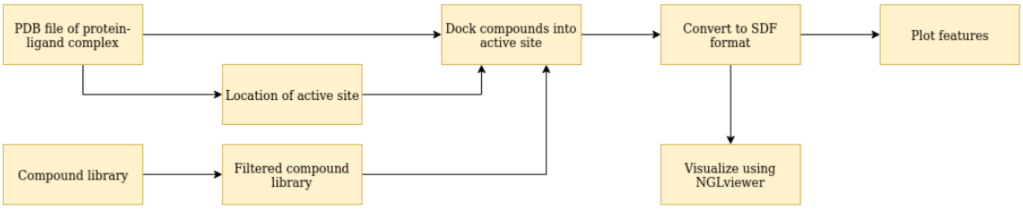

PDB file of protein-ligand complex
Dock compounds into active site
Convert to SDF format
Plot features
Location of active site
Compound library
Filtered compound library
Visualize using NGLviewer

Fig. 3.4: Simplified schematic of the protein-ligand docking workflow. Information on the image source is provided in the List of Figures.

QSAR

The fourth workflow (Figure 3.4) makes use of machine learning techniques to make predictions about unseen chemical compounds. Data from PubChem is used and cleaned using the OpenBabel toolkit to standardise formatting and remove duplicated molecules, before the Mordred library [89] is used to calculate 21 chemical descriptors, which are used as features for training a classification model. The dataset used as a case study is made up of around 8000 compounds which are classified as active or inactive as agonists of the estrogen receptor signalling (ERα) pathway [90]. A test-train split of 0.3:0.7 is used and a model based on the random forest classification algorithm, as implemented by the scikit-learn Python library [91], is trained. The trained model can be used directly within Galaxy to make predictions on unseen compounds, or its quality can be assessed using the figures and statistics automatically generated by the workflow (e.g. Figure 3.6).

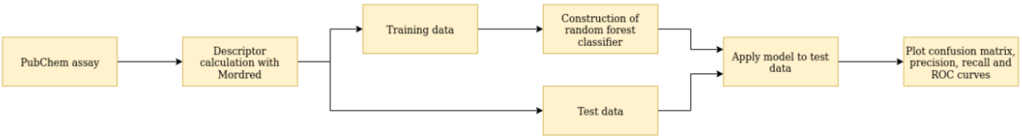

PubChem assay
Descriptor calculation with Mordred
Training data
Construction of random forest classifier
Test data
Apply model to test data
Plot confusion matrix, precision, recall and ROC curves

Fig. 3.5: Simplified schematic of the machine learning workflow. Information on the image source is provided in the List of Figures.

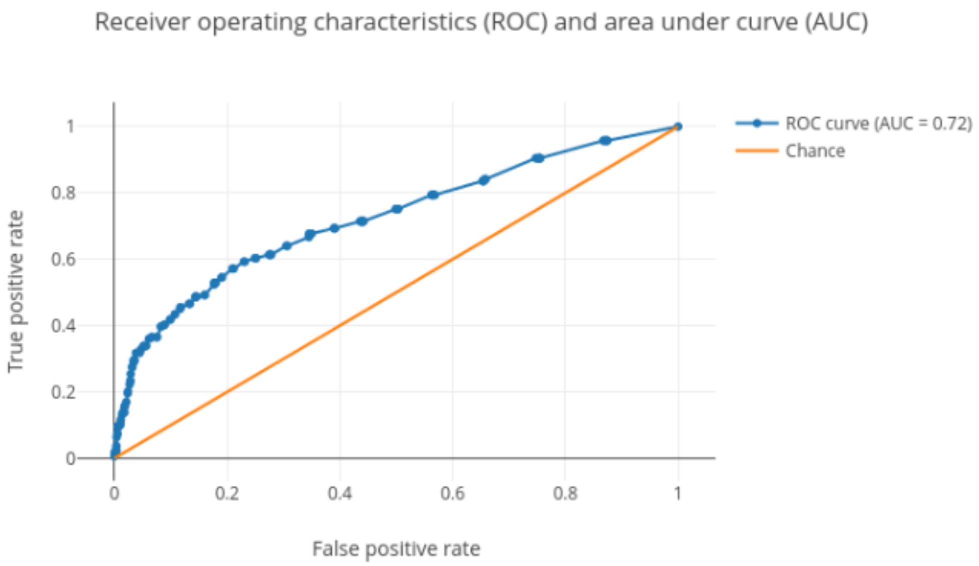


Fig. 3.6: Receiver operating curve generated by the QSAR workflow, as displayed in the Galaxy interface.

3.3.2 Capacity building

As mentioned, computational chemistry is a complex field, encompassing a multitude of concepts and software packages. High-quality online tutorials, which are accessible for newcomers, are missing for many of these - for example, protein-ligand docking. The Galaxy community has spearheaded a training initiative, the Galaxy Training Network [92], which provides tutorials for a range of topics, primarily in bioinformatics, but increasingly in other scientific fields, which make use of free, public Galaxy servers which give students access to a wide range of open-source tools via a graphical interface for users, ensuring that lack of knowledge of the command-line interface does not pose a barrier to learning scientific concepts. Eight different tutorials, covering protein-ligand docking, scoring, MD parameterisation, simulation, and analysis, QSAR-based prediction of biodegradation, and pharmacophore-based virtual screening were written. One of these, “High Throughput Molecular Dynamics and Analysis” was published as a standalone paper.

The aim of this paper was twofold: firstly, to provide a clear guide to the conduct of MD simulations on protein-ligand complexes using the Galaxy platform for the help of beginners to MD; and secondly, to demonstrate the use of Galaxy workflows, collections and command-line scripts for scaling up MD simulation, so that simulations of hundreds of ligands can be started simultaneously against a single protein at a single click. These ideas will be discussed more extensively in the subsequent chapter.

3.4 Conclusion

In conclusion, a complete web-based workbench for cheminformatics, computational chemistry, and molecular biophysics has been developed, together with training material to help introduce the topic to newcomers to the field. The ChemicalToolbox allows individual tools to be combined to form complex workflows, which can be executed either via the graphical interface or the command line. The latter case allows the scaling up of simulation and analysis to a massive scale. The remainder of this thesis describes some scientific use-cases in more detail, as well as the work done to enable command-line execution of the scientific workflows which have been developed.

The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform

The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform. **Simon A. Bray**, Xavier Lucas, Anup Kumar, Björn A. Grüning. *Journal of Cheminformatics, V*olume 12, Article number: 40, 1 June 2020, https://doi.org/10.1186/s13321-020-00442-7

Personal contribution:

I took over maintenance of an original version of the ChemicalToolbox in March 2019, which had been left unmaintained since 2014. I updated all tools to incorporate new functionality in the underlying software and fix bugs. In addition, I contributed many new tools, in particular for three-dimensional structural studies, such as for protein-ligand docking and molecular dynamics, I developed the four scientific workflows described in the paper, and wrote the submitted manuscript. Since publication, I have continued to maintain and develop the software further. In recognition of these major contributions to the paper, I am listed as first author for the publication.

Co-authors:

Xavier Lucas: co-planned the project and developed an initial version of the ChemicalToolbox prepared in 2014, and contributed to the paper.

Anup Kumar: integrated the visualizations and provided advice on use of Galaxy machine learning tools.

Björn Grüning: provided supervision of Simon Bray and code review, co-planned the project and developed an initial version of the ChemicalToolbox prepared in 2014, and contributed to the paper.

S. A. B

Simon Bray, 10.05.2022

Signatures:

The following co-authors confirm the above stated contribution:

Co-author	Date	Signature
Xavier Lucas	Basel, CH, 13.05.2022	
Anup Kumar	30.05.2022	Anup Kumar
Björn Grüning	31.05.2022	

Bray *et al. J Cheminform* (2020) 12:40
https://doi.org/10.1186/s13321-020-00442-7

Journal of Cheminformatics

SOFTWARE

Open Access

The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform

Simon A. Bray[1]*, Xavier Lucas[2], Anup Kumar[1] and Björn A. Grüning[1]

Abstract

Here, we introduce the ChemicalToolbox, a publicly available web server for performing cheminformatics analysis. The ChemicalToolbox provides an intuitive, graphical interface for common tools for downloading, filtering, visualizing and simulating small molecules and proteins. The ChemicalToolbox is based on Galaxy, an open-source web-based platform which enables accessible and reproducible data analysis. There is already an active Galaxy cheminformatics community using and developing tools. Based on their work, we provide four example workflows which illustrate the capabilities of the ChemicalToolbox, covering assembly of a compound library, hole filling, protein-ligand docking, and construction of a quantitative structure-activity relationship (QSAR) model. These workflows may be modified and combined flexibly, together with the many other tools available, to fit the needs of a particular project. The ChemicalToolbox is hosted on the European Galaxy server and may be accessed via https://cheminformatics.usegalaxy.eu.

Keywords: Cheminformatics, Protein-ligand docking, QSAR, Galaxy, Molecular dynamics

Introduction

Open-source software packages are now available for a wide range of cheminformatics applications, ranging from downloading [1, 2], manipulating, and processing small molecules [3–5], to protein-ligand docking calculations [6, 7], to quantum chemistry [8]. However, with the growth in the number of applications, the difficulty in combining these tools into easily usable, reproducible analysis workflows increases. Many tools require the user to possess some level of programming skill, or at least ability to use the command line; some also rely on unique file formats. Some tools require compilation of the source code for their use, which not only poses a challenge for computationally inexperienced scientists, but also muddies the waters if another user attempts to reproduce the analysis in another environment [9].

Use of technologies such as Conda [10] and containerization (most notably Docker and Singularity [11–13]) helps to mitigate some of these issues. Conda enables reproducible analyses and simplifies installation, while containerization technologies provide a common working environment across operating systems. However, knowledge of the command line is still required to run software, and the user is responsible for maintaining the thorough records (e.g. through use of a traditional lab book) that are required for full reproducibility of analyses.

Here, we present the ChemicalToolbox, a modular, intuitive platform for cheminformatics analysis, built within the Galaxy system [14, 15]. It combines numerous open-source cheminformatics tools, and integrates them into an intuitive, web-based user interface; requested jobs can then be sent to a high-performace computing (HPC) cluster for execution. Thus, the user has access to a range

*Correspondence: sbray@informatik.uni-freiburg.de
[1] Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, Freiburg, Germany
Full list of author information is available at the end of the article

© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

of useful tools and substantial compute resources, without being exposed directly to the HPC environment, or to the command-line interface used by much cheminformatics software. Tools can be run individually, or combined into workflows, which can then be shared with collaborators. All tools are made publicly available on the European Galaxy server, under the subdomain https://cheminformatics.usegalaxy.eu. As an alternative, the ChemicalToolbox can also be easily installed on personal computers, clusters, and cloud services; once installed, the system can be accessed simultaneously by multiple users, using current standard web browsers.

The ChemicalToolbox provides a range of tools for different applications, as depicted in Fig. 1. Chemical structures can be accessed from online databases such as PubChem [2] and ChEMBL [1]. Manipulation of chemical structures can be performed with OpenBabel [4] and RDKit [3], while calculation of molecular descriptors for QSAR studies may be done using Mordred [16] or PaDEL [17], which rely on RDKit and the Chemical Development Kit (CDK) [5] respectively. Protein-ligand docking may be performed using AutoDock Vina [6] and rDock [7]. Furthermore, the previously published BRIDGE platform [18] extends the core functionality of the ChemicalToolbox into molecular dynamics, providing a suite of tools which draws on the GROMACS [19], AmberTools [20], Parmed [21], and MDAnalysis [22] software. Apart from tools, the Galaxy codebase has been extended to provide features particularly useful for cheminformatics. These include support for a range of filetypes commonly used for reporting chemical structures, including PDB, SMILES, InChI, SMILES, SDF/MOL and MOL2, as well as tools for interconverting between these formats, based on OpenBabel. The most common GROMACS filetypes have also been made available. Another feature integrated directly into the Galaxy codebase is the NGLviewer [23], which may be used for visualization of compounds and macromolecules. Furthermore, apart from the features of the ChemicalToolbox itself, the inherent flexibility of the Galaxy system allows combination of the ChemicalToolbox with existing platforms developed by researchers working in other related areas, such as the Galaxy Genome Annotation project, metabolomics (Workflow4Metabolomics [24], Metaboloflow [25]), proteomics (Galaxy-P [26]), and machine learning—enabling the development of new, transdisciplinary workflows.

A number of other workflow management systems are commonly used in cheminformatics; the most prominent are Pipeline Pilot [27] and KNIME [28, 29]. Pipeline Pilot is a workflow management software developed by Accelrys Enterprise Platform and published as a proprietary application. It offers tools bundled into 'component

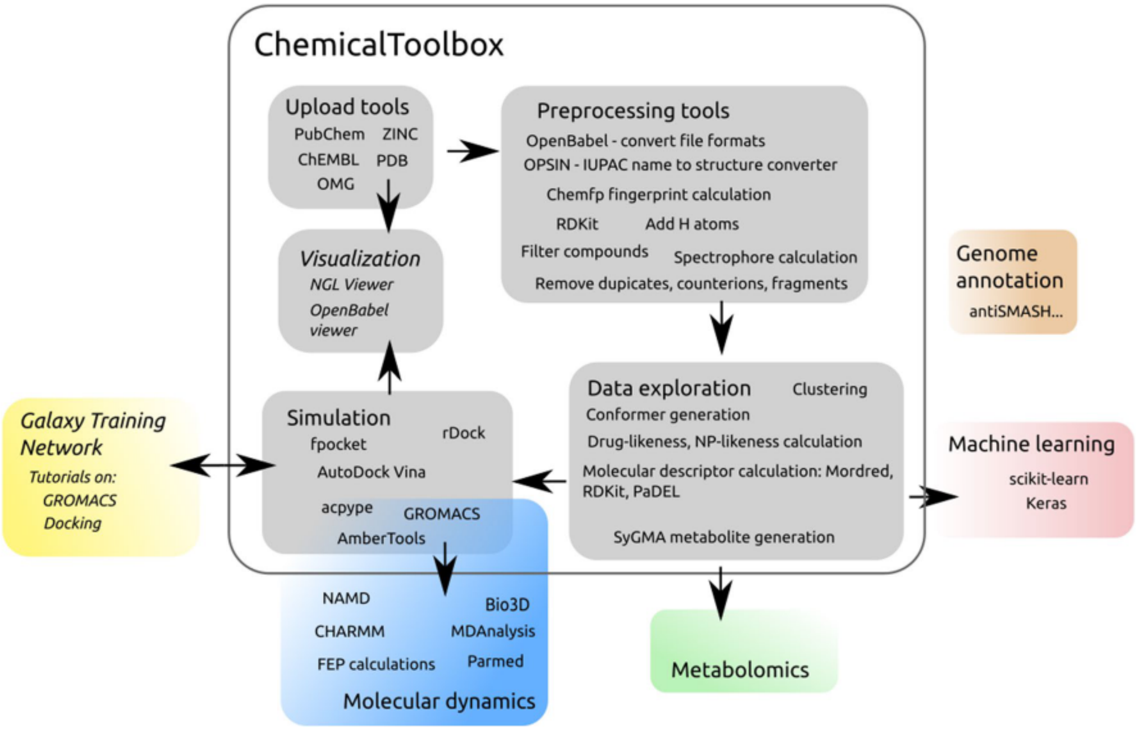


Fig. 1 Tools and visualizations available via the ChemicalToolbox. Colored boxes represent other related Galaxy communities, each with their own tools and workflows which can be flexibly combined with those of the ChemicalToolbox

collections'; two of which, the Chemistry and ADMET collections, provide similar functionality to the ChemicalToolbox. Pipeline Pilot is known for its user-friendly interface and ease of use for new users [30]. However, its proprietary nature makes reproducible research and sharing data very difficult or impossible, and the cost of purchasing a license is prohibitive for many researchers. KNIME, like the ChemicalToolbox, is open-source and free-of-charge, and also leverages well-known open-source software such as the CDK [5, 31] and RDKit in its extensions. KNIME 'nodes' are analogous to Galaxy tools, and are assembled into workflows in a similar manner. However, unlike the ChemicalToolbox, the free version of KNIME is not scalable for usage with an HPC or cloud environment; for this, a commercial license for KNIME Server must be purchased. Furthermore, the experience of using KNIME is comparable to programming with a graphical interface; KNIME describes its workflows as a 'graphic equivalent to a script'. By contrast, the ChemicalToolbox explicitly aims for accessibility to users without programming experience, as the majority of life scientists do not possess these skills.

Offering a cheminformatics toolbox as part of Galaxy has a number of advantages. Firstly, the Galaxy platform is a well-developed, mature project, and while originally developed for genomics research, it is fundamentally agnostic regarding the field of research. The ChemicalToolbox allows chemists to also access the features provided by the Galaxy platform, including a curated body of training material provided by the Galaxy Training Network [32]. Secondly, all ChemicalToolbox tools can be used via the European Galaxy server, which provides free access to generous computational resources for computational analysis, based on the de.NBI cloud [33] and the ELIXIR network [34]. However, the flexibility of the Galaxy system also allows users to download the ChemicalToolbox and run it locally or on their own server. There is already a small but active Galaxy computational chemistry community, constantly maintaining and contributing tools.

Implementation

While the ChemicalToolbox is primarily available via the European Galaxy instance, it has been designed as a dynamic cheminformatics platform, which can be implemented in diverse working environments and architectures. As it is built on top of the Galaxy framework, the ChemicalToolbox can be configured to run on diverse compute clusters, e.g. Kubernetes [35], TORQUE [36], DRMAA [37], Condor [38], or Pulsar [39]. This scalability allows users to perform compute-intensive cheminformatics calculations, including filtering, converting, and calculating hundreds of physicochemical properties and descriptors for many millions of compounds in a matter of hours.

Any software tool that is parameterizable and can be executed through a terminal command line can be wrapped as a Galaxy tool and included into the ChemicalToolbox, regardless of the programming language used for the implementation of the algorithm. Using the Galaxy ToolShed, each tool can be installed through the user's web browser by clicking on the required software—analogous to the 'app stores' provided by companies such as Apple or Microsoft. Moreover, the associated dependencies are automatically downloaded, compiled, and made accessible within the Galaxy environment. As the Galaxy ToolShed supports tool dependency versioning, the ChemicalToolbox is able to keep track of tool versions, enabling reproducibility and maintaining software provenance over time. Tool execution triggers creation of a Conda environment or download of a container with all software requirements installed, all with the specified versions. When executing outdated workflows in the ChemicalToolbox, the user is notified about newer versions of the tools and is allowed to choose specific versions for execution.

Many kinds of calculations in computational chemistry can be easily parallelized; an example is protein-ligand docking, where each of thousands of compounds in a library needs to be assessed individually. In the ChemicalToolbox, this is achieved by the use of collections. A Galaxy collection allows related files to be grouped together and processed identically. Input files (for example, a docking library in SDF format) are split according to defined parameters (the SDF delimiter), and when the AutoDock Vina or rDock tool is run on the resulting collection, docking is performed for each element of the collection separately and in parallel. Such a parallelization process is carried out automatically in the background, and can be easily parameterized and scaled-up by the server administrator responsible for maintaining the ChemicalToolbox as a suitable platform for high-performance computing.

Results

Here we present a number of case studies which demonstrate the capabilities of the ChemicalToolbox. For each case study, tools are chained together to form a 'workflow', which in the Galaxy system can be used much like an individual tool, thus enabling the flexible creation and combination of new functionalities as desired. Each of the workflows is published online under https://usegalaxy.eu/workflows/list_published and labelled with the 'cheminformatics' tag, as are sample Galaxy histories for each of the workflows under https://usegalaxy.eu/histo ries/list_published. Simplified schematic diagrams of the workflows are provided in Additional file 1, together with individual links to each workflow and history.

Hole filling and library optimization

The correct choice of chemical libraries is a crucial step in high-throughput virtual screening [40]. By using larger libraries, the chances of identifying hits increase, [41] along with the complexity and resources required for proper storage and testing. Moreover, it has been estimated that the chemical space contains more than 10^{60} molecules, a number impossible to handle currently or in the near future [42]. As a consequence, pre-filtered and focused libraries are commonly used in drug discovery, at the risk of exploring a minute portion of the chemical space (from hundreds to millions of compounds) and leaving large regions of the chemical space unexplored. As a result, hole filling and library optimization have assumed a major role in the fields of cheminformatics and drug discovery.

Here we demonstrate a ChemicalToolbox workflow which can be used to optimize a compound library using hole-filling. Downloading all drugs listed on the Therapeutic Target Database [43] (TTD) provides a small library of around 20,000 compounds. For the purpose of this workflow, our aim is to 'top-up' this library to 50,000, ensuring that added compounds are located in more sparsely occupied regions of the chemical space. Initially, we download the entirety of the PubChem database, which serves as the source for the new molecules, before calculating molecular fingerprints (using the Chemfp library [44]) for both PubChem and TTD compounds. Taylor-Butina clustering [45] is then performed on the TTD and singletons are identified, i.e. clusters which contain only a single molecule; these are used as seeds for expansion of the compound library. We then perform a similarity search to identify PubChem compounds within a distance threshold of the TTD singletons just found, which yields a total of around 2 million. In order to select compounds evenly, we perform Taylor-Butina clustering once again on our pool of 2 million molecules. A single compound is then selected from each of 30,000 different clusters, and added to the compound library, topping it up to 50,000.

Ligand library preparation

The preparation of ligand libraries is an important aspect of in silico high-throughput virtual screening, where small molecules are systematically tested in the catalytic or binding site of a protein (for example, via protein-ligand docking) aiming at the selection of candidate compounds with specific structural and physicochemical features. We provide a ChemicalToolbox workflow which offers an efficient solution for the large-scale management of data sets containing millions of molecules.

Initially, the workflow queries several freely available databases (including PubChem, ChEMBL and ZINC [46]) and automatically loads and converts all molecules to canonical SMILES for uniformity using OpenBabel. A specialist tool is used to extract all structures from the PubChem FTP site, while a general download tool can be used to access the other databases. After concatenating the resulting SMILES files and removing counterions and fragments, a final, cleaned dataset of almost 200 million unique compounds in the SMILES format was obtained (databases accessed on 04.10.2019). It is worth mentioning that the ChemicalToolbox has been specifically designed to automatically handle many format files (SDF and SMILES in the present workflow) encoding from a few hundreds or thousands up to many millions of molecules.

Protein-ligand docking

A common aim in cheminformatics is assessing the interactions of compounds with a protein. Protein-ligand docking involves estimating the interaction energy and the optimal recognition pose of a given ligand in complex with a protein [47, 48]. The ChemicalToolbox contains a number of tools which can be used for protein-ligand docking, including docking software AutoDock Vina and rDock. The fpocket tool can also be used for automatic identification of pockets which are suitable for docking [49].

Firstly, a protein structure and a compound library are created, either uploaded by the user or downloaded directly from online databases such as the PDB or ChEMBL. These can be processed using the Filter tool, which can apply either a commonly-used ruleset, such as Lipinski's rule-of-five [50], or a set of user-defined properties. In this case, we use two very different systems as illustrative examples: the Hsp90 chaperone protein (structure published under PDB accession code 2brc [51]) and the β_2-adrenergic receptor (structure published under PDB accession code 3pds [52]). Identification of a binding site allows the definition of a 3D box which is searched (using AutoDock Vina, though rDock is also available) to find a variety of possible binding positions for each of the compounds in the library. Results can be extracted from the output SD files and plotted, or used for further analysis.

Machine learning for predicting small molecule protein interactions

The Galaxy platform contains tools from multiple disciplines, which offers the opportunity to conduct interdisciplinary analyses. Recently, a suite of statistical and machine learning tools has been made available. This allows the development of quantitative structure-activity relationship (QSAR) models in the ChemicalToolbox.

As an illustrative example, we have published a Galaxy workflow for constructing a random forest classifier for predicting the activity of compounds as agonists of the estrogen receptor alpha signaling (ERα) pathway. Data are downloaded directly from the relevant PubChem assay, which forms part of the Tox21 program [53]. Initially, tools based on OpenBabel are used to remove counterions or small fragments from the compound library, as well as any duplicated molecules. For the remaining 7459 compounds, over 1800 two- and three-dimensional molecular descriptors are calculated using the Mordred tool [16] and 21 selected as features for building the classification model. A training/test split of 0.7/0.3 was used and a classification model built using the random forest method (in this case, the number of trees used by the classifier is 100) based on the descriptor values calculated for the training data. The random forest algorithm is applied using the implementation published as part of the scikit-learn Python library [54]. Aside from generation of a model that can be applied to new data, the effectiveness of the model can be tested and the results visualized in the form of a ROC curve, precision, recall and f-score plots, and confusion matrix. Here, an AUC value of 0.72 is achieved, which is reasonable considering the simple approach to feature and parameter selection applied here.

Training material

In addition to publishing the workflows described above, we have also created online tutorials providing an introduction to the features of the ChemicalToolbox, made available via the Galaxy Training Network [32], which already provides a range of introductory and advanced training material for analysis on the Galaxy platform. These tutorials may be found under https://training.galaxyproject.org/training-material/topics/computational-chemistry. For example, the tutorial on protein-ligand docking follows the workflow described above, using a small library of ligands downloaded from ChEMBL and docking them to the Hsp90 protein using AutoDock Vina. In addition, the tutorial guides the user through several other analyses of the compound library, using OpenBabel-based tools to visualize compounds and convert between different formats as required, and performing Taylor-Butina clustering based on calculated chemfp fingerprints.

The Galaxy computational chemistry community has developed a number of other more specialized tutorials, mainly focusing on molecular dynamics simulation and analysis. Other tutorials cover free energy perturbation and the application of machine learning to cheminformatics.

Conclusions

We have prepared the infrastructure and software for the ChemicalToolbox, a Galaxy-based cheminformatics webserver available via https://cheminformatics.usegalaxy.eu, and made a number of workflows available which demonstrate its capabilities, together with accompanying online introductory tutorials. Such a project can by its nature never be complete or comprehensive; new scientific advances will always result in the development of new software and new analytical approaches. However, the ChemicalToolbox is already sufficiently developed to be used to perform novel and interesting analyses, as well as for pedagogical purposes. We hope that the work published so far will provide a useful resource for chemists and cheminformaticians alike. With this publication, we hope to grow the Galaxy computational chemistry community further and to provide an impetus for further development of the ChemicalToolbox.

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1186/s13321-020-00442-7.

Additional file 1. Additional figures.

Acknowledgements

The authors would like to thank the following for discussions and instructive comments: Rolf Backofen, Chris Barnett, Leo Biscassi, Stefan Bleher, Andrew Dalke, Anika Erxleben, Stefan Günther, Gregory von Kuster, Hitesh Patel and Tharindu Senapathi. We also thank Tim Dudgeon and Rachel Skyner for discussion regarding the docking workflow and Alireza Khanteymoori for discussion regarding the machine learning workflow. Finally, we thank the entire Freiburg Galaxy Team for their support and helpful comments, particularly Joachim Wolff for reading and providing feedback on the paper.

Availability and requirements

Project name: ChemicalToolbox. Project home page: https://cheminformatics.usegalaxy.eu. Operating system(s): The ChemicalToolbox can be hosted on a Linux or OSX operating system; in addition, it is possible to run the ChemicalToolBox in a container on all operating systems that support OCI-approved containers, by means of the Galaxy Docker Project, which we recommend to users who wish to install on a Windows system (https://github.com/bgruening/docker-galaxy-stable). It is accessible for the user via a web browser on any operating system. Programming language: Python. Other requirements: dependencies for individual tools (e.g. RDKit, OpenBabel, GROMACS) installed via Conda or Singularity. License: The Galaxy software is released under the

Academic Free License 3.0. Individual tools are provided by their developers under a variety of open-source licenses.

Authors' contributions

SAB, XL and BAG planned the project and developed the software. SAB developed the training material and the workflows presented in the paper. AK integrated the visualizations and provided advice on use of Galaxy machine learning tools. BAG supervised the project. All authors contributed to writing the paper. All authors read and approved the final manuscript.

Funding

This work was supported by funding from the following organizations: S.A.B. was funded by the European Open Science Cloud (EOSC-Life) (Grant No. 824087); X.L. was funded by the Federal State of Baden Württemberg (Ministry of Arts and Science) through the "Zukunftsoffensive IV (ZO IV) Juniorprofessoren-Programm"; A.K. was funded by the German Federal Ministry of Education and Research [031 A538A de.NBI-RBC]; and B.A.G was funded by the German Research Foundation for the Collaborative Research Center 992 Medical Epigenetics [SFB 992/1 2012 and SFB 992/2 2016].

Availability of data and materials

All Galaxy tools are available via the Galaxy ToolShed under https://toolshed.g2.bx.psu.edu and may be executed on the publicly available Galaxy server https://cheminformatics.usegalaxy.eu. Example histories and workflows are available on https://cheminformatics.usegalaxy.eu via the links provided in the article text.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, Freiburg, Germany. [2] Roche Pharma Research and Early Development, Roche Innovation Center Basel, F. Hoffmann-La Roche Ltd., Grenzacherstrasse 124, Basel, Switzerland.

Received: 4 February 2020 Accepted: 16 May 2020
Published online: 01 June 2020

References

1. Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, Mutowo P, Atkinson F, Bellis LJ, Cibrián-Uhalte E et al (2016) The ChEMBL database in 2017. Nucleic Acids Res 45(D1):945–954
2. Kim S, Thiessen PA, Bolton EE, Chen J, Fu G, Gindulyte A, Han L, He J, He S, Shoemaker BA et al (2015) PubChem substance and compound databases. Nucleic Acids Res 44(D1):1202–1213
3. Landrum G (2019) RDKit: Open-Source Cheminformatics Software. https://www.rdkit.org. Accessed 23 Jan 20.
4. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR (2011) OpenBabel: an open chemical toolbox. J Cheminform 3(1):33
5. Willighagen EL, Mayfield JW, Alvarsson J, Berg A, Carlsson L, Jeliazkova N, Kuhn S, Pluskal T, Rojas-Chertó M, Spjuth O et al (2017) The Chemistry Development Kit (CDK) v2. 0: atom typing, depiction, molecular formulas, and substructure searching. J Cheminform 9(1):33
6. Trott O, Olson AJ (2009) AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem 31(2):455–461
7. Ruiz-Carmona S, Alvarez-Garcia D, Foloppe N, Garmendia-Doval AB, Juhos S, Schmidtke P, Barril X, Hubbard RE, Morley SD (2014) rDock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids. PLoS Comput Biol 10(4):1003571
8. Turney JM, Simmonett AC, Parrish RM, Hohenstein EG, Evangelista FA, Fermann JT, Mintz BJ, Burns LA, Wilke JJ, Abrams ML et al (2012) Psi4: an open-source ab initio electronic structure program. Wiley Interdiscip Rev Comput Mol Sci 2(4):556–565
9. Taschuk M, Wilson G (2017) Ten simple rules for making research software more robust. PLoS Comput Biol 13(4)
10. Grüning B, Dale R, Sjödin A, Chapman BA, Rowe J, Tomkins-Tinch CH, Valieris R, Köster J (2018) Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat Methods 15(7):475
11. Merkel D (2014) Docker: lightweight Linux containers for consistent development and deployment. Linux J 2014(239):2
12. Boettiger C (2015) An introduction to Docker for reproducible research. ACM SIGOPS Oper Syst Rev 49(1):71–79
13. Kurtzer GM, Sochat V, Bauer MW (2017) Singularity: scientific containers for mobility of compute. PloS ONE 12(5):0177459
14. Blankenberg D, Kuster GV, Bouvier E, Baker D, Afgan E, Stoler N, Taylor J, Nekrutenko A (2014) Dissemination of scientific software with Galaxy ToolShed. Genome Biol 15(2):403
15. Afgan E, Baker D, van den Beek M, Blankenberg D, Bouvier D, Čech M, Chilton J, Clements D, Coraor N, Eberhard C, Grüning B, Guerler A, Hillman-Jackson J, Von Kuster G, Rasche E, Soranzo N, Turaga N, Taylor J, Nekrutenko A, Goecks J (2016) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. Nucleic Acids Res 44(W1):3–10
16. Moriwaki H, Tian Y-S, Kawashita N, Takagi T (2018) Mordred: a molecular descriptor calculator. J Cheminform 10(1):4
17. Yap CW (2011) PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem 32(7):1466–1474
18. Senapathi T, Bray S, Barnett CB, Grüning B, Naidoo KJ (2019) Biomolecular Reaction & Interaction Dynamics Global Environment (BRIDGE). Bioinformatics 35(18):3508–3509
19. Abraham MJ, Murtola T, Schulz R, Páll S, Smith JC, Hess B, Lindahl E (2015) GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 1–2:19–25
20. Case D, et al. (2018) AmberTools Manual 2018. University of California, San Francisco. University of California. http://ambermd.org/doc12/Amber18.pdf. Accessed 23 Jan 20.
21. Swails J, Hernandez C, Mobley D, Nguyen H, Wang L, Janowski P (2016) ParmEd: Cross-program parameter and topology file editor and molecular mechanical simulator engine. https://parmed.github.io/ParmEd/html/index.html. Accessed 23 Jan 20.
22. Michaud-Agrawal N, Denning EJ, Woolf TB, Beckstein O (2011) MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. J Comput Chem 32(10):2319–2327
23. Rose AS, Bradley AR, Valasatava Y, Duarte JM, Prlić A, Rose PW (2018) NGL viewer: web-based molecular graphics for large complexes. Bioinformatics 34(21):3755–3758
24. Guitton Y, Tremblay-Franco M, Corguillé GL, Martin J-F, Pétéra M, Roger-Mele P, Delabrière A, Goulitquer S, Monsoor M, Duperier C, Canlet C, Servien R, Tardivel P, Caron C, Giacomoni F, Thévenot EA (2017) Create, run, share, publish, and reference your LC–MS, FIA–MS, GC–MS, and NMR data analysis workflows with the Workflow4Metabolomics 3.0 galaxy online infrastructure for metabolomics. Int J Biochem Cell Biol 93:89–101
25. van Rijswijk M, Beirnaert C, Caron C, Cascante M, Dominguez V, Dunn WB, Ebbels TMD, Giacomoni F, Gonzalez-Beltran A, Hankemeier T, Haug K, Izquierdo-Garcia JL, Jimenez RC, Jourdan F, Kale N, Klapa MI, Kohlbacher O, Koort K, Kultima K, Corguillé GL, Moschonas NK, Neumann S, O'Donovan C, Reczko M, Rocca-Serra P, Rosato A, Salek RM, Sansone S-A, Satagopam V, Schober D, Shimmo R, Spicer RA, Spjuth O, Thévenot EA, Viant MR, Weber RJM, Willighagen EL, Zanetti G, Steinbeck C (2017) The future of metabolomics in ELIXIR. F1000Research 6:1649
26. Stewart PA, Kuenzi BM, Mehta S, Kumar P, Johnson JE, Jagtap P, Griffin TJ, Haura EB (2019) The Galaxy platform for reproducible affinity proteomic mass spectrometry data analysis. In: Methods in molecular biology. Springer, New York, p. 249–61
27. Accelrys: BIOVIA Pipeline Pilot. 2019. https://www.3dsbiovia.com/products/collaborative-science/biovia-pipeline-pilot. Accessed 23 Jan 20.
28. Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, Ohl P, Thiel K, Wiswedel B (2009) KNIME—the Konstanz Information Miner: version 2.0 and beyond. ACM SIGKDD Explor Newsl 11(1):26–31
29. KNIME: Konstanz Information Miner. 2020. https://www.knime.com/. Accessed 31 Mar 20.
30. Warr WA (2012) Scientific workflow systems: Pipeline Pilot and KNIME. J Comput Aided Mole Des 26(7):801–804
31. Beisken S, Meinl T, Wiswedel B, de Figueiredo LF, Berthold M, Steinbeck C (2013) KNIME-CDK: workflow-driven cheminformatics. BMC Bioinform 14(1):257

32. Batut B, Hiltemann S, Bagnacani A, Baker D, Bhardwaj V, Blank C, Bretaudeau A, Brillet-Guéguen L, Čech M, Chilton J et al (2018) Community-driven data analysis training for biology. Cell Syst 6(6):752–758
33. German Network for Bioinformatics Infrastructure: de.NBI cloud. 2020. https://www.denbi.de/cloud. Accessed 31 Mar 20.
34. ELIXIR network: ELIXIR. 2020. https://elixir-europe.org/. Accessed 31 Mar 20.
35. Kubernetes: Production-Grade Container Orchestration. 2020. https://kubernetes.io/. Accessed 31 Mar 20.
36. Adaptive Computing: QUEue Manager (TORQUE). 2013. http://www.adaptivecomputing.com/products/torque. Accessed 23 Jan 20.
37. Troger P, Rajic H, Haas A, Domagalski P (2007) Standardization of an API for distributed resource management systems. In: Seventh IEEE international symposium on cluster computing and the grid (CCGrid 2007). IEEE, Rio de Janeiro
38. Tannenbaum T, Wright D, Miller K, Livny M (2001) Condor—a distributed job scheduler. In: Sterling T (ed) Beowulf cluster computing with Linux. MIT Press, Cambridge
39. Chilton J. Pulsar. 2019. https://github.com/galaxyproject/pulsar. Accessed 23 Jan 20.
40. Kumar V, Krishna S, Siddiqi MI (2015) Virtual screening strategies: recent advances in the identification and design of anti-cancer agents. Methods 71:64–70
41. Lyu J, Wang S, Balius TE, Singh I, Levit A, Moroz YS, O'Meara MJ, Che T, Algaa E, Tolmachova K, Tolmachev AA, Shoichet BK, Roth BL, Irwin JJ (2019) Ultra-large library docking for discovering new chemotypes. Nature 566(7743):224–229
42. Bohacek RS, McMartin C, Guida WC (1996) The art and practice of structure-based drug design: a molecular modeling perspective. Med Res Rev 16(1):3–50
43. Li YH, Yu CY, Li XX, Zhang P, Tang J, Yang Q, Fu T, Zhang X, Cui X, Tu G et al (2017) Therapeutic Target Database update 2018: enriched resource for facilitating bench-to-clinic research of targeted therapeutics. Nucleic Acids Res 46(D1):1121–1127
44. Dalke A (2013) The FPS fingerprint format and chemfp toolkit. J Cheminform 5(1):36
45. Butina D (1999) Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: a fast and automated way to cluster small and large data sets. J Chem Inf Comput Sci 39(4):747–750
46. Sterling T, Irwin JJ (2015) ZINC 15-ligand discovery for everyone. J Chem Inf Model 55(11):2324–2337
47. Pagadala NS, Syed K, Tuszynski J (2017) Software for molecular docking: a review. Biophys Rev 9(2):91–102
48. Sousa SF, Fernandes PA, Ramos MJ (2006) Protein-ligand docking: current status and future challenges. Proteins Struct Funct Bioinform 65(1):15–26
49. Le Guilloux V, Schmidtke P, Tuffery P (2009) Fpocket: an open source platform for ligand pocket detection. BMC Bioinform 10(1):168
50. Lipinski CA (2004) Lead- and drug-like compounds: the rule-of-five revolution. Drug Discov Today Technol 1(4):337–341
51. Cheung K-MJ, Matthews TP, James K, Rowlands MG, Boxall KJ, Sharp SY, Maloney A, Roe SM, Prodromou C, Pearl LH, Aherne GW, McDonald E, Workman P (2005) The identification, synthesis, protein crystal structure and in vitro biochemical evaluation of a new 3,4-diarylpyrazole class of Hsp90 inhibitors. Bioorg Med Chem Lett 15(14):3338–3343
52. Rosenbaum DM, Zhang C, Lyons JA, Holl R, Aragao D, Arlow DH, Rasmussen SGF, Choi H-J, DeVree BT, Sunahara RK, Chae PS, Gellman SH, Dror RO, Shaw DE, Weis WI, Caffrey M, Gmeiner P, Kobilka BK (2011) Structure and function of an irreversible agonist-β_2 adrenoceptor complex. Nature 469(7329):236–240
53. National Center for Advancing Translational Sciences: Tox21 Data Challenge 2014. 2014. https://tripod.nih.gov/tox21/challenge. Accessed 23 Jan 20
54. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

BMC

Biomolecular Reaction and Interaction Dynamics Global Environment (BRIDGE)

Biomolecular Reaction and Interaction Dynamics Global Environment (BRIDGE).
Tharindu Senapathi, **Simon Bray**, Christopher B Barnett, Björn Grüning, Kevin J Naidoo
Bioinformatics, Volume 35, Issue 18, 15 September 2019, Pages 3508–3509,
https://doi.org/10.1093/bioinformatics/btz107

Personal contribution:

I developed tools for molecular dynamics simulation in the Galaxy platform based on the GROMACS software, covering all steps from system parameterization, to solvation, energy minimization, equilibration and production simulations. In addition, I provided code review for the tools written by other co-authors, and have taken responsibility for long-term maintenance of the tools after publication. In recognition of these major contributions to the paper, I am listed as second author of the publication.

Co-authors:

Tharindu Senapathi: developed the CHARMM and NAMD molecular dynamics tools and trajectory analysis tools described in this publication.

Chris Barnett: provided technical supervision of the project and code review.

Björn Grüning: provided technical supervision of the project and code review.

Kevin Naidoo: designed and provided overall supervision of the project, and wrote the initial and final draft of the paper.

S. A. B

Simon Bray, 01.12.2021

Signatures:

The following co-authors confirm the above stated contribution:

Co-author	Date	Signature
Tharindu Senapathi	02.12.2021	Senapathi
Chris Barnett	02.12.2021	CBarnett
Björn Grüning	31.05.2022	B
Kevin Naidoo	03.12.2021	Kevin J Naidoo

Bioinformatics, 35(18), 2019, 3508–3509
doi: 10.1093/bioinformatics/btz107
Advance Access Publication Date: 13 February 2019
Applications Note

Structural bioinformatics

Biomolecular Reaction and Interaction Dynamics Global Environment (BRIDGE)

Tharindu Senapathi[1], Simon Bray[2], Christopher B. Barnett[1], Björn Grüning[2,3,*] and Kevin J. Naidoo [1,4,*]

[1]Scientific Computing Research Unit and Department of Chemistry, University of Cape Town, Rondebosch 7701, South Africa, [2]Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany, [3]Center for Biological Systems Analysis (ZBSA), University of Freiburg, 79104 Freiburg, Germany and [4]Institute for Infections Disease and Molecular Medicine, Faculty of Health Science, University of Cape Town, Rondebosch 7701, South Africa

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on October 8, 2018; revised on January 14, 2019; editorial decision on February 7, 2019; accepted on February 11, 2019

Abstract

Motivation: The pathway from genomics through proteomics and onto a molecular description of biochemical processes makes the discovery of drugs and biomaterials possible. A research framework common to genomics and proteomics is needed to conduct biomolecular simulations that will connect biological data to the dynamic molecular mechanisms of enzymes and proteins. Novice biomolecular modelers are faced with the daunting task of complex setups and a myriad of possible choices preventing their use of molecular simulations and their ability to conduct reliable and reproducible computations that can be shared with collaborators and verified for procedural accuracy.

Results: We present the foundations of Biomolecular Reaction and Interaction Dynamics Global Environment (BRIDGE) developed on the Galaxy platform that makes possible fundamental molecular dynamics of proteins through workflows and pipelines via commonly used packages, such as NAMD, GROMACS and CHARMM. BRIDGE can be used to set up and simulate biological macromolecules, perform conformational analysis from trajectory data and conduct data analytics of large scale protein motions using statistical rigor. We illustrate the basic BRIDGE simulation and analytics capabilities on a previously reported CBH1 protein simulation.

Availability and implementation: Publicly available at https://github.com/scientificcomputing/BRIDGE and https://usegalaxy.eu

Contact: kevin.naidoo@uct.ac.za or bjoern.gruening@gmail.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Molecular mechanisms underlie biological phenomena. Consequently, locating molecular modeling tools within a genomics research platform consolidates the three components of the bioinformatics ecosystem that enables a seamless progression from (i) DNA, RNA or protein sequence analysis to (ii) gene expression profiling, metabolic and functional pathway analysis to (iii) molecular structural analysis of proteins—identification of therapeutic targets, development of biomarkers and examination of protein alterations. It is this intention to link genomic analytics to molecular simulations that motivates our development, reported here, of the Biomolecular Reaction and Interaction Dynamics Global Environment (BRIDGE) within the Galaxy platform (Afgan *et al.*, 2018; Goecks *et al.*, 2010). BRIDGE is a web server, based on the Galaxy framework, to perform molecular dynamics (MD) simulations of biomolecules and conduct statistical analyses on the trajectory data produced.

© The Author(s) 2019. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Downloaded from https://academic.oup.com/bioinformatics/article/35/18/3508/5317160 by Albert-Ludwigs Universitat user on 12 July 2022

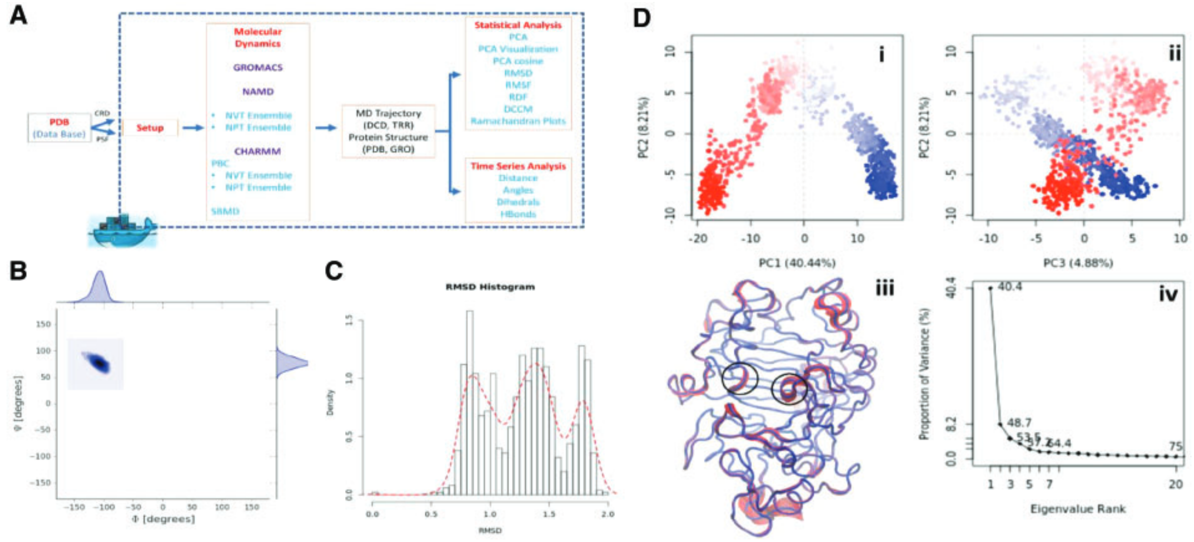


Fig. 1. (**A**) Overview of the tools to run MD and analyze in the BRIDGE MD platform (https://github.com/scientificomputing/BRIDGE). (**B**) Ramachandran plot of the cellulose ligand glycosidic bond that is targeted for hydrolysis. (**C**) Histogram of the RMSD of Cα atoms of the protein. (**D**) PCA of the Cα atoms of the protein backbone motions

Protein functions are carried out by atomistic scale binding interactions of their residues in catalytic/binding domains in the case of enzymes/lectins/antibodies and conformational changes of their backbones and loops that regulate transport, folding, etc. (Fersht, 1999). It is now universally accepted that probing the molecular mechanisms of reaction and interactions are best made using MD simulations.

The Galaxy software platform is an open-source platform that has historically focused on sequencing analysis of all kinds (Goecks *et al.*, 2010), but now is a general framework for data analysis, functioning beyond life sciences. Here we add molecular modeling functionalities to Galaxy, specifically the capability to perform (i) MD simulations, (ii) statistical mechanics analyses (PDFs, time correlation functions, etc.) on dynamics trajectories and (iii) statistical analysis and big data analytics on individual biomolecules or families of biomolecular structures and configurations.

A set of Galaxy tool wrappers (Fig. 1A) have been developed to set up and run classical MD simulations using either one (or all) of the CHARMM (Brooks *et al.*, 2009), NAMD (Phillips *et al.*, 2005) and GROMACS (Lindahl *et al.*, 2001), MD engines. Following this, wrappers were developed to include the features of MDAnalysis (Gowers *et al.*, 2016) for subsequent statistical mechanics analysis and finally wrappers for the Bio3D (Skjærven *et al.*, 2014) package were written to make the statistical analysis of structural/conformational biomolecular motions produced from trajectories possible.

2 Demonstration and conclusion

We illustrate some of the analytical tools able to investigate conformational changes by analysis of a typical short protein simulation such as for CBH1 (see Supplementary Material for details). The Ramachandran style dihedral angle plot of a key glycosidic linkage of the oligosaccharide ligand is computed using the Ramachandran Plots tool (Fig. 1B). The protein motion is analyzed using the root mean square deviation (RMSD) tool. Three distinct conformations around RMSD of 0.8, 1.2 and 1.8 Angstrom can be seen from the RMSD histogram (Fig. 1C).

Protein conformational changes can be investigated in greater detail using tools in the statistical analyses module. Here PCA was used to discover the statistically meaningful conformations in the 5 *ns* CBH1 trajectory (Fig. 1D). The principal motions within the trajectory and the vital motions needed for conformational changes were identified. Two distinct groupings along the dominant PC1 plane (Fig. 1Di and iv) indicating a non-periodic conformational change are identified. The groupings along the PC2 and PC3 planes (Fig. 1Dii) do not completely cluster separately implying that these global motions are periodic. The PC1 is linked to an active site motion (Fig. 1Diii) that limits the motion to a key glycosidic bond (Fig. 1B).

Funding

This work was supported by the South African Research Chairs Initiative (SARChI) of the Department of Science and Technology (DST) and National Research Foundation (NRF) [grant 48103].

Conflict of Interest: none declared.

References

Afgan,E. *et al.* (2018) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res.*, **46**, W537–W544.

Brooks,B.R. *et al.* (2009) CHARMM: the biomolecular simulation program. *J. Comput. Chem.*, **30**, 1545–1614.

Fersht,A. (1999) *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. W.H. Freeman, New York.

Goecks,J. *et al.* (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.*, **11**, R86.

Michaud-Agrawal,N. *et al.* (2011) MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.* **32**, 2319–2327.

Lindahl,E. *et al.* (2001) GROMACS 3.0: a package for molecular simulation and trajectory analysis. *Mol. Model. Annu.*, **7**, 306–317.

Phillips,J.C. *et al.* (2005) Scalable molecular dynamics with NAMD. *J. Comput. Chem.*, **26**, 1781–1802.

Skjærven,L. *et al.* (2014) Integrating protein structural dynamics and evolutionary analysis with Bio3D. *BMC Bioinformatics*, **15**, 399.

Downloaded from https://academic.oup.com/bioinformatics/article/35/18/3508/5317160 by Albert-Ludwigs Universitat user on 12 July 2022

Intuitive, reproducible high-throughput molecular dynamics in Galaxy: a tutorial

Intuitive, reproducible high-throughput molecular dynamics in Galaxy: a tutorial
Simon A. Bray, Tharindu Senapathi, Christopher B. Barnett, Björn A. Grüning
*Journal of Cheminformatics, V*olume 12, Article number: 54, 10 September 2020,
https://doi.org/10.1186/s13321-020-00451-6

Personal contribution:

I provided the initial idea for the project. I designed the Galaxy workflow for molecular dynamics simulation described in the paper and wrote the 'Introduction', 'Methods: simulation' and 'High-throughput workflows' sections in both the training material and the paper. I developed the 'Merge GROMACS topologies' and 'Extract energy components with GROMACS' tools which were required for the publication. In recognition of these major contributions to the paper, I am listed as first author for the publication.

Co-authors:

Tharindu Senapathi: designed the workflow for MD analysis described in the paper and co-wrote the 'Methods: analysis' section.

Chris Barnett: co-wrote the 'Methods: analysis' section and provided supervision of Tharindu Senapathi and code review.

Björn Grüning: provided supervision of Simon Bray and code review, and contributed to the paper.

S. A. B

Simon Bray, 01.12.2021

Signatures:

The following co-authors confirm the above stated contribution:

Co-author	Date	Signature
Tharindu Senapathi	02.12.2021	Senapathi
Chris Barnett	02.12.2021	CBarnett
Björn Grüning	31.05.2022	B.

Journal of Cheminformatics

EDUCATIONAL

Open Access

Intuitive, reproducible high-throughput molecular dynamics in Galaxy: a tutorial

Simon A. Bray[1], Tharindu Senapathi[2], Christopher B. Barnett[2*] and Björn A. Grüning[1*]

Abstract

This paper is a tutorial developed for the data analysis platform Galaxy. The purpose of Galaxy is to make high-throughput computational data analysis, such as molecular dynamics, a structured, reproducible and transparent process. In this tutorial we focus on 3 questions: How are protein-ligand systems parameterized for molecular dynamics simulation? What kind of analysis can be carried out on molecular trajectories? How can high-throughput MD be used to study multiple ligands? After finishing you will have learned about force-fields and MD parameterization, how to conduct MD simulation and analysis for a protein-ligand system, and understand how different molecular interactions contribute to the binding affinity of ligands to the Hsp90 protein.

Keywords: Galaxy, Molecular Dynamics, Reproducible

Introduction

Molecular dynamics (MD) is a commonly used method in computational chemistry and cheminformatics, in particular for studying the interactions between small molecules and large biological macromolecules such as proteins [1]. However, the barrier to entry for MD simulation is high; not only is the theory difficult to master, but commonly used MD software is technically demanding. Furthermore, generating reliable, reproducible simulation data requires the user to maintain detailed records of all parameters and files used, which again poses a challenge to newcomers to the field. One solution to the latter problem is usage of a workflow management system such as Galaxy [2], which provides a selection of tools for molecular dynamics simulation and analysis [3]. MD simulations are rarely performed singly; in recent years, the concept of high-throughput molecular dynamics (HTMD) has come to the fore [4, 5]. Galaxy lends itself well to this kind of study, as we will demonstrate in this paper, thanks to features allowing construction of complex workflows, which can then be executed on multiple inputs in parallel.

This tutorial provides a detailed workflow for high-throughput molecular dynamics with Galaxy, using the N-terminal domain (NTD) of Hsp90 (heat shock protein 90) as a case-study. Galaxy [2] is a data analysis platform that provides access to thousands of tools for scientific computation. It features a web-based user interface while automatically and transparently managing underlying computation details, enabling structured and reproducible high-throughput data analysis. This tutorial provides sample data, workflows, hands-on material and references for further reading. It presumes that the user has a basic understanding of the Galaxy platform. The aim is to guide the user through the various steps of a molecular dynamics study, from accessing publicly available crystal structures, to performing MD simulation (leveraging the popular GROMACS [6, 7] engine), to analysis of the results.

The entire analysis described in this article can be conducted efficiently on any Galaxy server which has the needed tools. In particular, we recommend using the Galaxy Europe server (https://cheminformatics.usega

*Correspondence: chris.barnett@uct.ac.za; gruening@informatik.uni-freiburg.de
[1] Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, Freiburg, Germany
[2] Department of Chemistry and Scientific Computing Research Unit, University of Cape Town, 7700 Cape Town, South Africa

© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

laxy.eu) or the Galaxy South Africa server (https://galaxy-compchem.ilifu.ac.za). For users who wish to run their own Galaxy server locally, we provide a Docker container (https://quay.io/repository/galaxy/computational-chemistry-training) containing a full Galaxy installation, with all tools required for the tutorial preinstalled.

The tutorial presented in this article has been developed as part of the Galaxy Training Network [8] and its most up-to-date version is accessible online on the Galaxy Training Materials website [9], under the URL https://training.galaxyproject.org/training-material/topics/computational-chemistry/tutorials/htmd-analysis/tutorial.html.

What is high-throughput molecular dynamics?

Molecular dynamics (MD) is a method to simulate molecular motion by iterative application of Newton's laws of motion. It is often applied to large biomolecules such as proteins or nucleic acids. A common application is to assess the interaction between these macromolecules and a number of small molecules (e.g. potential drug candidates). This tutorial provides a guide to setting up and running a high-throughput workflow for screening multiple small molecules, using the open-source GROMACS tools provided through the Galaxy platform. Following simulation, the trajectory data is analyzed using a range of tools to investigate structural properties and correlations over time.

Why is Hsp90 interesting to study?

The 90 kDa heat shock protein (Hsp90) is a chaperone protein responsible for catalyzing the conversion of a wide variety of proteins to a functional form; examples of the Hsp90 clientele, which totals several hundred proteins, include nuclear steroid hormone receptors and protein kinases [10]. The mechanism by which Hsp90 acts varies between clients, as does the client binding site; the process is dependent on post-translational modifications of Hsp90 and the identity of co-chaperones which bind and regulate the conformational cycle [11].

Due to its vital biochemical role as a chaperone protein involved in facilitating the folding of many client proteins, Hsp90 is an attractive pharmaceutical target. In particular, as protein folding is a potential bottleneck to cellular reproduction and growth, blocking Hsp90 function using inhibitors which bind tightly to the ATP binding site of the NTD could assist in treating cancer; for example, the antibiotic geldanamycin and its analogs are under investigation as possible anti-tumor agents [12, 13].

In the structure which will be examined during this tutorial, the ligand of concern is a resorcinol, a common class of compounds with affinity for the Hsp90 N-terminal domain. It is registered in the PubChem database under the compound ID 135508238 [14]. As can be seen by viewing the PDB structure, the resorcinol part of the structure is embedded in the binding site, bound by a hydrogen bond to residue aspartate-93. The ligand structure also contains a triazole and a fluorophenyl ring, which lie nearer to the surface of the protein.

Methods: simulation

This section guides the reader through the step-by-step process required to prepare, run and analyze Hsp90. A brief explanation of the theory and purpose of each step is provided. Refer to the hands-on sections as these describe the task with tools and parameters to be carried out using Galaxy.

Get data

Create a new Galaxy history and then download a crystal structure for the Hsp90 protein from the Protein Data Bank (PDB). The structure is provided under accession code `6HHR` [16] and shows Hsp90 in complex with the resorcinol ligand (Fig. 1).

Hands-on 1: Data upload

1. Create a new history for this tutorial
2. Search Galaxy for the **Get PDB** tool. Request the accession code `6HHR`.
3. Rename the dataset to 'Hsp90 structure'

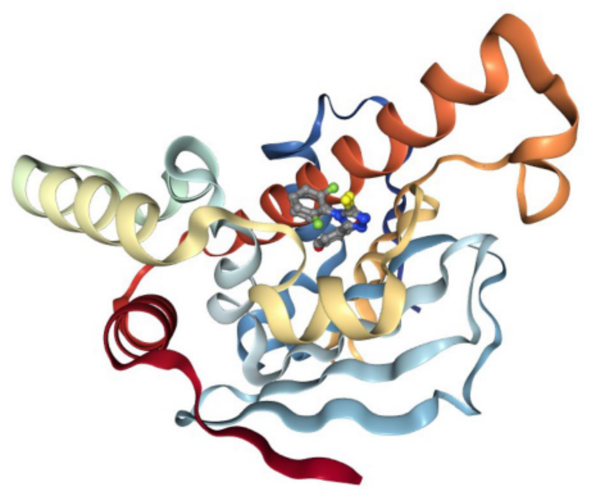


Fig. 1 Hsp90 cartoon view. Hsp90 cartoon with ligands in active site, rendered using the Galaxy NGL plugin [15]

Topology generation

The PDB structure is prepared for MD simulation in a process referred to as parameterization or topology generation.

GROMACS distinguishes between constant and dynamic attributes of the atoms in the system. The constant attributes (e.g. atom charges, bonds connecting atoms) are listed in the topology (TOP file), while dynamic attributes (attributes that can change during a simulation, e.g. atom position, velocities and forces) are stored in structure (PDB or GRO) and trajectory (XTC and TRR) files.

The PDB file includes neither parameters for simulations, nor the positions of hydrogen atoms. Therefore, before beginning simulation, this information must be calculated.

Extract protein and ligand coordinates

Parameterization is performed separately for the ligand and protein. The PDB file is separated into two sets of coordinates—one for the ligand and one for the protein—using the simple text manipulation tools integrated into Galaxy.

Hands-on 2: Separate protein and ligand coordinates

1. **Search in textfiles** with the following parameters:
   - *"Select lines from"*: 'Hsp90 structure'
   - *"that"*: `Don't Match`
   - *"Regular Expression"*: `HETATM`
2. Rename output to 'Protein (PDB)'
3. **Search in textfiles** with the following parameters:
   - *"Select lines from"*: 'Hsp90 structure'
   - *"that"*: `Match`
   - *"Regular Expression"*: `AG5E`
4. Rename output to 'Ligand (PDB)'

The PDB file is filtered twice: once for lines which do not match `HETATM`, which returns a PDB file containing only protein, not ligand and solvent; and once for lines which match the ligand's identity code `AG5E`, which returns a PDB file containing only the ligand.

Set up protein topology

The topology for the protein file is calculated with the **GROMACS initial setup** tool.

Hands-on 3: Generate protein topology

GROMACS initial setup with the following parameters:
- *"PDB input file"*: 'Protein (PDB)' file
- *"Force field"*: `AMBER99SB`
- *"Water model"*: `TIP3P`
- *"Generate detailed log"*: `Yes`

A force field is essentially a function to calculate the potential energy of a system, based on various empirical parameters (for the atoms, bonds, charges, dihedral angles and so on). There are a number of families of force fields; some of the most commonly used include CHARMM [17], AMBER [18], GROMOS [19] and OpenFF [20] (for a recent, accessible overview see [21]). One of the main AMBER force fields for protein modeling, `ff99SB`, was selected.

Apart from the force field, a water model was selected to model the solvent; a wide range of models exist for this purpose. The common TIP3P model is selected, which is an example of a 'three-site model'—so-called because the molecule is modeled using three points, corresponding to the three atoms of water (four- and five-site models include additional 'dummy atoms' representing the negative charges of the lone pairs of the oxygen atom) [22].

The tool produces four outputs: a GRO file (containing the coordinates of the protein), a TOP file (containing other information, including charges, masses, bonds and angles), an ITP file (which will be used to restrain the protein position in the equilibration steps later on), and a log file.

Please note all the GROMACS tools provided in Galaxy output a log file. These files provide useful information for debugging purposes.

Generate a topology for the ligand

The acpype [23] tool is used to generate a topology for the ligand. This provides a convenient interface to the AmberTools suite and creates the ligand topology in the format required by GROMACS.

Inspecting the contents of the ligand PDB file shows that it contains no hydrogen atoms. Hydrogens were added to the topology using the 'Compound conversion' tool (based on OpenBabel [24]).

Hands-on 4: Generate ligand topology

1. **Compound conversion**:
   - *"Molecular input file"*: 'Ligand (PDB)'
   - *"Output format"*: `Protein Data Bank format (pdb)`
   - *"Add hydrogens appropriate for pH"*: `7.0`
2. Rename the output file to ´Hydrated ligand (PDB)'
3. **Generate MD topologies for small molecules**:
   - *"Input file"*: 'Ligand (PDB)'
   - *"Charge of the molecule"*: `0`
   - *"Multiplicity"*: `1`
   - *"Force field to use for parameterization"*: `gaff`
   - *"Save GRO file?"*: `Yes`

The GAFF (general AMBER force field) is selected, which is a generalized AMBER force field [25] which can be applied to almost any small organic molecule.

Appropriate charge and multiplicity parameters are entered. The ligand studied is a simple organic molecule, with no charge; therefore, the charge is set to 0 and the multiplicity to 1. Alternative values for multiplicity need only be considered for more exotic species such as metal complexes or carbenes.

Next, the topologies are combined and the simulation box is defined.

Combine topology and GRO files

The separate topology and structure files for both protein and ligand are combined into a single set of files to continue with the simulation setup. A dedicated Galaxy tool is provided for this, using the Python library ParmEd [26].

Hands-on 5: Combine GRO and topology files

Merge GROMACS topologies with the following parameters:

- *"Protein topology (TOP) file"*: `TOP` file created by the **GROMACS initial setup tool**
- *"Ligand topology (TOP or ITP) file"*: `Topology` file created by the **acpype tool**
- *"Protein structure (GRO) file"*: `GRO` file created by the **GROMACS initial setup tool**
- *"Ligand structure (GRO) file"*: `Structure file (GRO format)` file created by the **acpype tool**

Note that, apart from this tool, the Galaxy platform also provides an integrated text editor for making more advanced changes to GROMACS topologies or configuration files.

Create the simulation box

The next step, once combined coordinate (GRO) and topology (TOP) files have been created, is to create a simulation box in which the system is situated.

Hands-on 6: Create simulation box

GROMACS structure configuration with the following parameters:

- *"Input structure"*: `System GRO file` (Input dataset)
- *"Configure box?"*: `Yes`
  - *"Box dimensions in nanometers"*: `1.0`
  - *"Box type"*: `Triclinic`
- *"Generate detailed log"*: `Yes`

This tool returns a new GRO structure file, containing the same coordinates as before, but defining a simulation box such that every atom is a minimum of 1 nm from the box boundary. A variety of box shapes are available to choose from: triclinic is selected, as it provides efficient packing in space and thus fewer computational resources need to be devoted to simulation of solvent.

Solvation

The next step is solvation of the newly created simulation box. Water was chosen as a solvent to in order to simulate biological conditions. Note that the system is charged (depending on the pH) and it is neutralized by the addition of the sodium and chloride ions (replacing existing water molecules) using the solvation tool.

Hands-on 7: Solvation

GROMACS solvation and adding ions with the following parameters:

- *"GRO structure file"*: output of **GROMACS structure configuration**
- *"System topology"*: `output`
- *"Generate detailed log"*: `Yes`

Energy minimization

After the solvation step, parameterization of the system is complete and preparatory simulations can be performed. The first of these is energy minimization, which can be carried out using the 'GROMACS energy minimization' tool. The purpose of energy minimization is to relax the structure, removing any steric clashes or unusual geometry which would artificially raise the energy of the system.

Hands-on 8: Energy minimization

GROMACS energy minimization with the following parameters:

- *"GRO structure file."*: GRO output of **GROMACS solvation and adding ions**
- *"Topology (TOP) file."*: TOP output of **GROMACS solvation and adding ions**
- *"Parameter input"*: `Use default (partially customisable) setting`
  - *"Number of steps for the MD simulation"*: `50000`
  - *"EM tolerance"*: `1000.0`
- *"Generate detailed log"*: `Yes`
- Rename GRO output to `Minimized GRO file`

The EM tolerance here refers to the maximum force which will be allowed in a minimized system. The simulation will be terminated when the maximum force is less than this value, or when 50,000 steps have elapsed. The 'Extract energy components' tool is used to plot the convergence of the potential energy during the minimization.

Hands-on 9: Checking EM convergence

1. **Extract energy components with GROMACS** with the following parameters:
   - *"EDR file"*: EDR output of **GROMACS energy minimization**
   - *"Terms to calculate"*: `Potential`
   - *"Output format"*: `Galaxy tabular`
2. On the output tabular file, click on the 'Visualize this data' icon. This provides a range of visualization options. Select 'Line chart (jqPlot)'.
3. In the visualization window which appears, click on 'Select data.' Enter the following parameters:
   - *"Provide a label"*: `Energy potential`
   - *"Values for x-axis"*: `Column: 1`
   - *"Values for y-axis"*: `Column: 2`

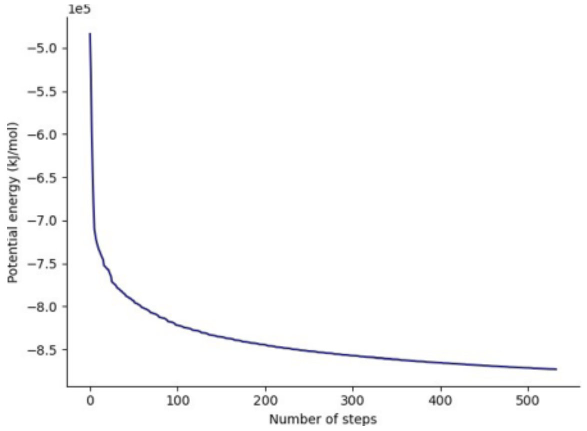


Fig. 2 Energy potential during the EM simulation

As seen in Fig. 2, the system first drops rapidly in energy, before slowly converging on the minimized state.

Equilibration

At this point equilibration of the solvent around the solute (i.e. the protein) is necessary. This is performed in two stages: equilibration under an NVT (or isothermal-isochoric) ensemble, followed by an NPT (or isothermal-isobaric) ensemble. Use of the NVT ensemble entails maintaining constant number of particles, volume and temperature, while the NPT ensemble maintains constant number of particles, pressure and temperature. Simulation under the NVT ensemble allows the solvent to be brought to the desired temperature, while simulation under the NPT ensemble brings the solvent to the correct pressure.

For equilibration, the protein is held in place while the solvent is allowed to move freely around it. This is achieved using the position restraint file (ITP) created during the system setup. This restraint does not prevent protein movement; rather movement is energetically penalized.

Hands-on 10: NVT equilibration

GROMACS simulation with the following parameters:

- *"GRO structure file"*: `Minimized GRO file` (from energy minimization step)
- *"Topology (TOP) file"*: TOP file produced by solvation step.
- In *"Inputs"*:
  - *"Position restraint (ITP) file"*: ITP file produced by initial setup step.
- In *"Outputs"*:
  - *"Trajectory output"*: `Return .xtc file (reduced precision)`
  - *"Structure output"*: `Return .gro file`
  - *"Produce a checkpoint (CPT) file"*: `Produce CPT output`
  - *"Produce an energy (EDR) file"*: `Produce EDR output`
- In *"Settings"*:
  - *"Parameter input"*: `Use default (partially customisable) setting`
    - *"Bond constraints (constraints)"*: `All bonds (all-bonds).`
    - *"Temperature /K"*: `300`
    - *"Step length in ps"*: `0.002`
    - *"Number of steps that elapse between saving data points (velocities, forces, energies)"*: `1000`
    - *"Number of steps for the simulation"*: `50000`
- *"Generate detailed log"*: `Yes`

Once the NVT equilibration is complete, it is worth using the 'Extract energy components' tool again to check whether the system temperature has converged on 300 K. This can be done as described for energy minimization, this time specifying `Temperature` under *Terms to calculate* rather than `Potential`. The plot should show the temperature reaching 300 K and remaining there, albeit with some fluctuation.

Having stabilized the temperature of the system with NVT equilibration, the pressure is stabilized by equilibrating using the NPT (constant number of particles, pressure, temperature) ensemble. The NPT simulation is continued from the NVT simulation by using the checkpoint (CPT) file saved at the end of the NVT simulation.

Hands-on 11: NPT equilibration

GROMACS simulation with the following parameters:

- *"GRO structure file"*: GRO output of **GROMACS simulation** (NVT equilibration)
- *"Topology (TOP) file"*: TOP file produced by solvation step.
- In *"Inputs"*:
  - *"Checkpoint (CPT) file"*: Output of **GROMACS simulation** (NVT equilibration))
  - *"Position restraint (ITP) file"*: ITP file produced by initial setup step.
- In *"Outputs"*:
  - *"Trajectory output"*: `Return .xtc file (reduced precision)`
  - *"Structure output"*: `Return .gro file`
  - *"Produce a checkpoint (CPT) file"*: `Produce CPT output`
  - *"Produce an energy (EDR) file"*: `Produce EDR output`
- In *"Settings"*:
  - *"Ensemble"*: `Isothermal-isobaric ensemble (NPT)`
  - *"Parameter input"*: `Use default (partially customisable) setting`
    - *"Bond constraints (constraints)"*: `All bonds (all-bonds).`
    - *"Temperature /K"*: `300`
    - *"Step length in ps"*: `0.002`
    - *"Number of steps that elapse between saving data points (velocities, forces, energies)"*: `1000`
    - *"Number of steps for the simulation"*: `50000`
- *"Generate detailed log"*: `Yes`

After the NPT equilibration is complete, 'Extract energy components' can be used again to view the pressure of the system. This is done as described for energy minimization, specifying `Pressure` under *Terms to calculate*. The plot should show convergence on 1 bar and remain there, although some fluctuation is expected.

Production simulation

The restraints are removed and a production simulation is carried out for 1 million steps. With a step size of 1 fs, this simulation represents a total time length of 1 ns. This is rather short compared to the state-of-the-art, but sufficient for the purposes of a tutorial.

Hands-on 12: Main simulation

GROMACS simulation with the following parameters:

- *"GRO structure file"*: Output of **GROMACS simulation** (NPT equilibration)
- *"Topology (TOP) file"*: Output of the solvation step
- In *"Inputs"*:
  - *"Checkpoint (CPT) file"*: Output of **GROMACS simulation** (NPT simulation))
- In *"Outputs"*:
  - *"Trajectory output"*: `Return .xtc file (reduced precision)`
  - *"Structure output"*: `Return .gro file`
  - *"Produce a checkpoint (CPT) file"*: `Produce CPT output`
- In *"Settings"*:
  - *"Ensemble"*: `Isothermal-isobaric ensemble (NPT)`
  - *"Parameter input"*: `Use default (partially customisable) setting`
    - *"Temperature /K"*: `300`
    - *"Step length in ps"*: `0.001`
    - *"Number of steps that elapse between saving data points (velocities, forces, energies)"*: `1000`
    - *"Number of steps for the simulation"*: `1000000`
- *"Generate detailed log"*: `Yes`

Methods: analysis

An analysis of the GROMACS simulation outputs (structure and trajectory file) will be carried out using Galaxy tools developed for computational chemistry [3] based on popular analysis software, such as MDAnalysis [27], MDTraj [28], and Bio3D [29]. These tools output both tabular files as well as a variety of attractive plots.

Convert coordinate and trajectory formats

Before beginning a detailed analysis, the structure and trajectory files generated previously need to be converted into different formats. The structural coordinates of the system in GRO format are converted into PDB format using the 'Convert coordinate and trajectory formats' tool (which is based on the 'editconf' GROMACS command). This PDB file will be used by most analysis tools as a starting structure. This tool can also be used to carry out initial setup (as discussed in the simulation methods section) for GROMACS simulations and to convert from PDB to GRO format. The trajectory file is converted from XTC to DCD format, as a number of tools (particularly those based on Bio3D) only accept trajectories in DCD format. This tool can also be used to interconvert between several other trajectory formats.

Hands-on 13: Convert coordinate and trajectory formats

1. **GROMACS structure configuration** with the following parameters:
   - *"Output format"*: `PDB file`
   - *"Configure box?"*: `No`
2. **MDTraj file converter** with the following parameters:
   - *"Output format"*: `DCD file`

RMSD analysis

The Root Mean Square Deviation (RMSD) and Root Mean Square Fluctuation (RMSF) are calculated to check the stability and conformation of the protein and ligand through the course of the simulation. RMSD is a standard measure of structural distance between coordinate sets that measures the average distance between a group of atoms. The RMSD of the Cα atoms of the protein backbone is calculated here and is a measure of how much the protein conformation has changed between different time points in the trajectory. Note that for more complex systems, consider a more focused selection.

For the RMSD analysis of the ligand, the 'Select domains' parameter of the tool can for convenience be set to 'Ligand'; however, this automatic selection sometimes fails. Instead the 'Residue ID' is specified in the textbox provided. In this example the ligand's Residue ID is 'G5E'. The output is the requested RMSD data as a time series, the RMSD plotted as a time series and as a histogram (for example, see Fig. 3 in "Results and discussion" section).

Hands-on 14: RMSD Analysis: protein

RMSD Analysis with the following parameters:

- *"DCD trajectory input"*: output of **MD-Traj file converter**
- *"PDB input"*: output of **GROMACS structure configuration**
- *"Select domains"*: `C-alpha`

Hands-on 15: RMSD Analysis: ligand using Residue ID

RMSD Analysis with the following parameters:

- *"DCD trajectory input"*: output of **MD-Traj file converter**
- *"PDB input"*: output of **GROMACS structure configuration**
- *"Select domains"*: `Residue ID`
  - *"Residue ID"*: `G5E`

RMSF analysis

The Root Mean Square Fluctuation (RMSF) is valuable to consider, as it represents the deviation at a reference position over time. The fluctuation in space of particular amino acids in the protein are considered. The Cα of the protein, designated by `C-alpha`, is a good selection to understand the change in protein structure. Depending on the system these fluctuations can be correlated to experimental techniques including Nuclear Magnetic Resonance (NMR) and Mössbauer spectroscopy [30, 31]. The output from the tools is the requested RMSF data and the RMSF plotted as a time series (for example, see Fig. 5 in "Results and discussion" section).

Hands-on 16: RMSF Analysis

RMSF Analysis with the following parameters:

- *"DCD trajectory input"*: output of **MD-Traj file converter**
- *"PDB input"*: output of **GROMACS structure configuration**
- *"Select domains"*: `C-alpha`

PCA

Principal component analysis (PCA) converts a set of correlated observations (movement of selected atoms in protein) to a set of principal components (PCs) which are linearly independent (or uncorrelated). Here several related tools are used. The PCA tool calculates the PCA in order to determine the relationship between statistically meaningful conformations (major global motions) sampled during the trajectory. The Cα carbons of the protein backbone are again a good selection for this purpose. Outputs include the PCA raw data and figures of the relevant principal components (PCs) as well as an eigenvalue rank plot (see Fig. 6) which is used to visualize the proportion of variance due to each principal component (remembering that the PCs are ranked eigenvectors based on the variance). Having discovered the principal components usually these are visualized. The PCA visualization tool creates trajectories of specific principal components which can be viewed in a molecular viewer such as VMD [32] or NGL viewer [15]. The PCA cosine content when close to 1 indicates that the simulation is not converged and a longer simulation is needed. For values below 0.7, no statement can be made about convergence or lack thereof.

Hands-on 17: PCA with BIO3D

PCA with the following parameters:

- *"DCD trajectory input"*: output of **MD-Traj file converter**
- *"PDB input"*: output of **GROMACS structure configuration**
- *"Select domains"*: `C-alpha`

Hands-on 18: PCA visualization

PCA visualization with the following parameters:

- *"DCD trajectory input"*: output of **MD-Traj file converter**
- *"PDB input"*: output of **GROMACS structure configuration**
- *"Select domains"*: `C-alpha`

> Hands-on 19: Cosine content calculation
>
> **Cosine Content** with the following parameters:
> - *"DCD/XTC trajectory input"*: output of **MDTraj file converter**
> - *"PDB/GRO input"*: output of **GROMACS structure configuration**

Hydrogen bond analysis

Hydrogen bonding interactions contribute to binding and are worth investigating, in particular persistent hydrogen bonds. All possible hydrogen bonding interactions between the two selected regions, here the protein and the ligand, are investigated over time using the VMD hydrogen bond analysis tool included in Galaxy. Hydrogen bonds are identified and in the output the total number of hydrogen bonds and occupancy over time is returned.

> Hands-on 20: Hydrogen bond analyis
>
> **Hydrogen Bond Analysis using VMD** with the following parameters:
> - *"DCD/XTC trajectory input"*: output of **MDTraj file converter**
> - *"PDB/GRO input"*: output of **GROMACS structure configuration**
> - *"Selection 1"*: `protein`
> - *"Selection 2"*: `resname G5E`

Results and discussion

After the completion of the simulation, the following questions arise: (1) is the simulation converged enough, and (2) what interesting molecular properties are observed. The timescale of motions of interest are in the picosecond to nanosecond range; these are motions such as domain vibration, hydrogen bond breaking, translation diffusion and side chain fluctuations. To observe meaningful conformational transitions of the protein μs sampling would be needed, but this is not the purpose here.

The PCA cosine content of the dominant motion related to PC1 is 0.93, indicating that the simulation is not fully converged. This is expected due to the short simulation length. For production level simulations, it is the norm to extend simulations to hundreds of nanoseconds in length, if not microseconds. A short simulation time of 1 ns was chosen as this tutorial is designed to be carried out on public webservers, which have finite computational resources to dedicate to training purposes.

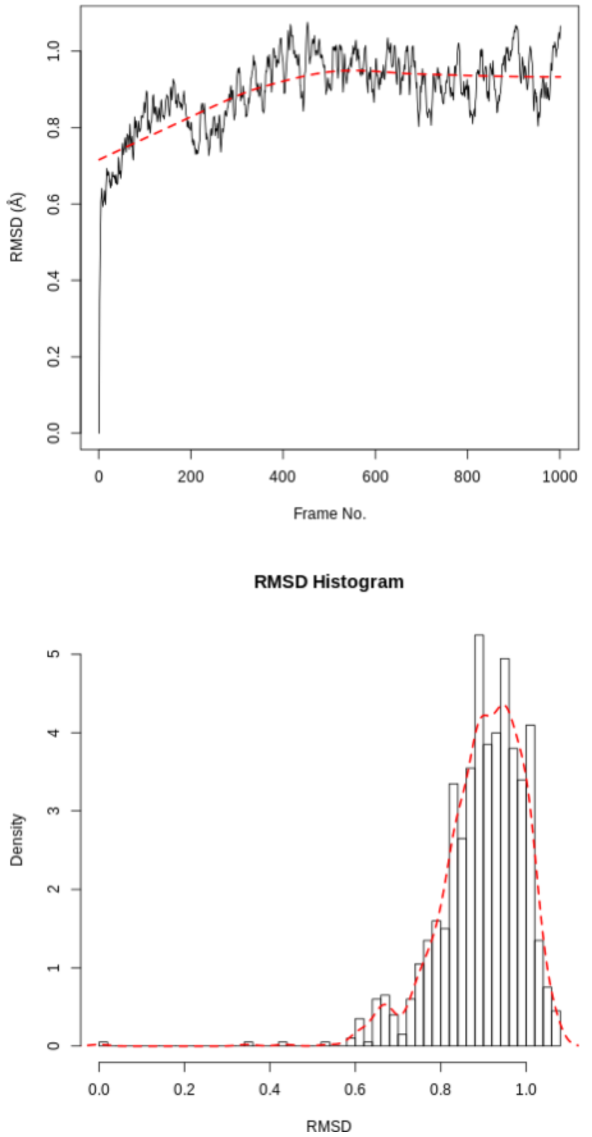


Fig. 3 RMSD for protein. RMSD time series and histogram for the protein

RMSD protein

The RMSD time series for the protein shows a thermally stable and equilibrated structure that plateaus at 1.0Å with an average RMSD between 0.8Å and 1.0Å. There are no large conformational changes during the simulation. The RMSD histogram confirms this, see Fig. 3. Note these graphs are automatically created by Galaxy as part of the tool's outputs.

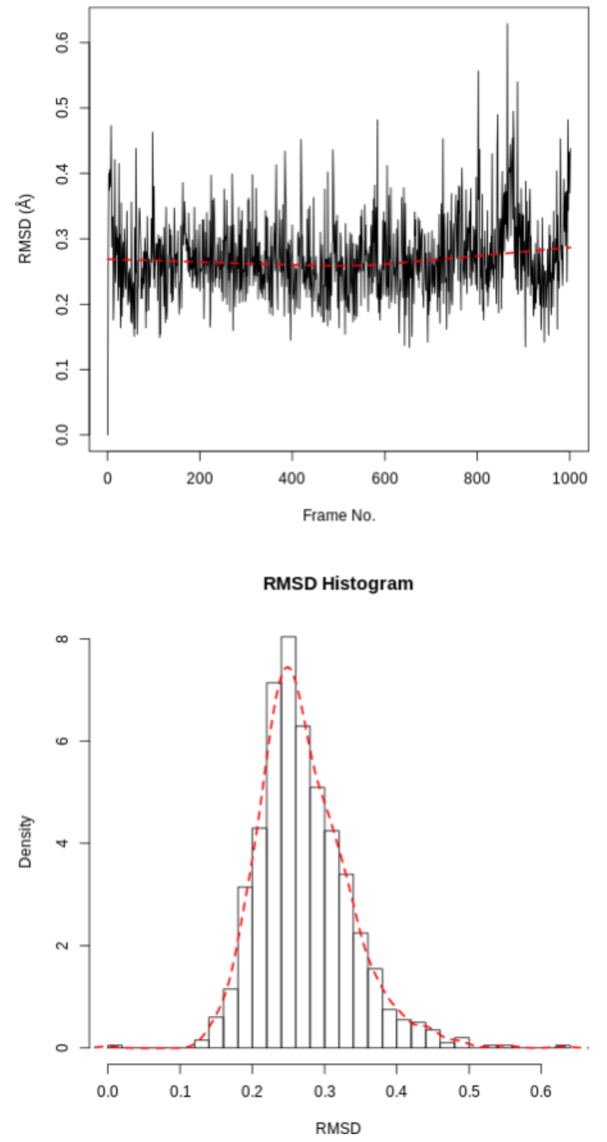


Fig. 4 RMSD for the ligand. RMSD time series and histogram for the ligand

RMSD ligand

Calculating the RMSD of the ligand is necessary to check if it is stable in the active site and to identify possible binding modes. If the ligand is not stable, there will be large fluctuations in the RMSD.

In our case the ligand is stable with a single binding mode. The RMSD fluctuates around 0.3Å, with a slight fluctuation near the end of the simulation. This is more clearly seen in the histogram, see Figure 4. The conformation seen during simulation is very similar to that in the crystal structure and the ligand is stable in the active site.

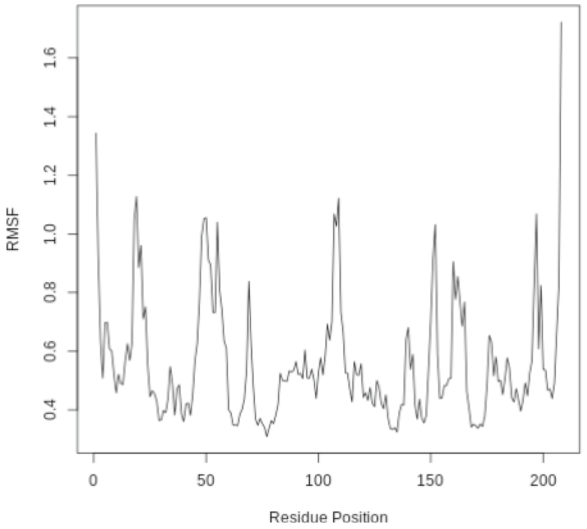


Fig. 5 RMSF for the protein. RMSF(Å) vs the residue position. Large fluctuations occur at various positions, which correspond to flexible loop regions on the surface of the protein

RMSF

When considering the RMSF (Fig. 5), fluctuations greater than 1.0Å are of interest; for example see the fluctuations near residue positions 50, 110 and 160. Inspecting the structure with molecular visualization software such as VMD, these can be seen to correspond to flexible loop regions on the protein surface. In addition, very large fluctuations are seen for the C-terminus; this is common and no investigation is needed.

Note that the first few residues of this protein are missing in the PDB, and therefore residue position 0 in the RMSF corresponds to position 17 in the Hsp90 FASTA primary sequence. This is a fairly common problem that can occur with molecular modeling of proteins, where there may be missing residues at the beginning or within the sequence.

PCA

The first three principal components are responsible for 32.8% of the total variance, as seen in the eigenvalue rank plot (Fig. 6). The first principal component (PC1) accounts for 15.4% of the variance (see PC1 vs PC2 and eigenvalue rank plots in Fig. 6). Visualization of PC1 using VMD shows a rocking motion and wagging of the C-terminus.

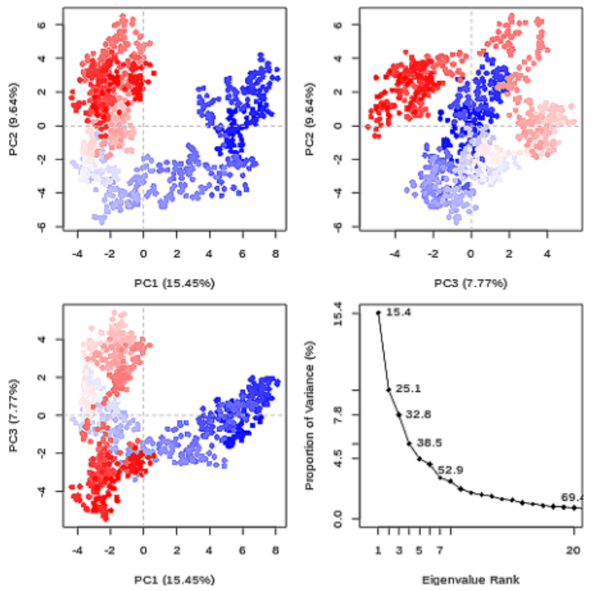


Fig. 6 Principal component analysis. PCA results which include graphs of PC2 vs PC1, PC2 vs PC3, PC3 vs PC1 colored from blue to red in order of time, and an eigenvalue rank plot (Scree plot). In the eigenvalue plot the cumulative variance is labeled for each data point

Hydrogen bonding

Multiple hydrogen bonds were identified between the active site of the protein and the ligand. The hydrogen bond between aspartate-93 and the ligand (as identified in the crystal structure) was found to be persistent, meeting the hydrogen bond criteria for 89.22% of the simulation. A hydrogen bond between the ligand and the carbonyl group of glycine-97 was found to have a 15.27% occupancy. Hydrogen bonding interactions with threonine-184, asparagine-51 and lysine-58 were also observed but these were not persistent and only present for a minority of the simulation. These values can be accessed from the 'Percentage occupancy of the H-bond' output of the hydrogen bond analysis tool.

High throughput workflows

Up until this step, Galaxy tools have been applied sequentially to datasets. This is useful to gain an understanding of the steps involved, but becomes tedious if the workflow needs to be run on multiple protein-ligand systems. Fortunately, Galaxy allows entire workflows to be executed with a single mouse-click, enabling straightforward high-throughput analyses.

The high-throughput capabilities of Galaxy are demonstrated by running the workflow detailed so far on a further three ligands [33–37].

Hands-on 21: High-throughput MD

1. Create a new history for running the high-throughput workflow and name it 'Hsp90 HTMD simulation'.
2. Upload the SD-file containing the new ligand structures from Zenodo [33] and rename it 'Ligands (SDF)'.
3. Import the simulation workflow from the European [34] or the South African Galaxy server [35].
4. Run the imported workflow with the following parameters:
   - *"SDF file with (docked) ligands"*: 'Ligands (SDF)' file.
5. Import the analysis workflow from the European [36] or the South African Galaxy server [37] (also available through Zenodo).
6. Run the imported workflow with the following parameters:
   - *"Send results to a new history"*: 'Yes'
   - *"History name"*: 'Hsp90 HTMD analysis'
   - *"GRO input"*: Collection of GRO files produced by simulation workflow
   - *"XTC input"*: Collection of XTC files produced by simulation workflow

This process runs the entire simulation and analysis procedure described so far on the new set of ligands. It uses Galaxy's collection [38] feature to organize the data; each item in the history is a collection (essentially a directory containing multiple individual datasets) containing one file corresponding to each of the input ligands.

Note that the SD-file needs to contain ligands with the correct 3D coordinates for MD simulation. The easiest way to obtain these is using a molecular docking tool such as Autodock Vina [39] or rDock [40]; tutorials and workflows are available for both of these from the Galaxy Training Network. As an example, the history in which the SD-file used in the HTMD workflow is generated (using AutoDock Vina) is provided [41].

Further information

Apart from manual setups or collections, there are several other alternatives which are helpful in scaling up workflows. Galaxy supports and provides training material for converting histories to workflows [42], using multiple histories [43], and the Galaxy Application Programming Interface (API) [44]. For beginners and users who prefer a visual interface, automation can be done using multiple histories and collections with the standard Galaxy user interface.

If you are able to write small scripts, you can automate everything you have learned here with the Galaxy API. This approach allows interaction with the server to automate repetitive tasks and create more complex workflows (which may have repetition or branching). The simplest way to access the API is through the Python library BioBlend [45]. An example Python script, which uses BioBlend to run the GROMACS simulation workflow for each of a list of ligands, is given in the hands-on box below.

Hands-on 22: BioBlend script

```
from bioblend import galaxy

# Server and account details
API_KEY = 'YOUR USEGALAXY.EU API KEY'
gi = galaxy.GalaxyInstance(key=API_KEY,
    url='https://usegalaxy.eu/')

# ID for GROMACS workflow
workflow_id = 'adc6d049e9283789'

# Dataset IDs for ligands to dock
ligands = {
# ligand_name: dataset ID,
'lig1': '11ac94870d0bb33a79c5fa18b0fd3b4c',
# ...
}

# Loop over ligands, invoking workflow
for name, _id in ligands.items():
    inv = gi.workflows.invoke_workflow(
        workflow_id,
        inputs={
            '1': {'src': 'hda', 'id': _id}
        },
        history_name=f'HTMD run on {name}'
    )
```

Conclusion

This tutorial provides a guide on how to study protein-ligand interaction using molecular dynamics in Galaxy. Performing such analyses in Galaxy makes it straightforward to set up, schedule and run workflows, removing much of the difficulty from MD simulation. Thus, the technical barrier to performing high-throughput studies is greatly reduced. Results are structured in the form of Galaxy histories or collections, and include ready-plotted diagrams, which ensure data can be easily understood and reproduced if necessary. Apart from streamlining the process for existing MD users, this tutorial should also prove useful as a pedagogical guide for educating students or newcomers to the field.

After completing the tutorial, the user will be familiar at a basic level with a range of MD analysis techniques, and understand the steps required for a typical MD simulation. Thus, they will be equipped to apply these tools to their own problems.

Acknowledgements
We thank Bérénice Batut for helpful comments and discussions, Bert Droesbeke for assistance building the Docker container and Oleg Zharkov for assistance integrating the Galaxy text editor. In addition, we thank the entire Galaxy Training Network, as well as the European Galaxy and ilifu teams, for their support.

The European Galaxy server, which was used for the calculations described, is in part funded by Collaborative Research Centre 992 Medical Epigenetics (DFG grant SFB 992/1 2012) and German Federal Ministry of Education and Research (BMBF grants 031 A538A/A538C RBC, 031L0101B/031L0101C de.NBI-epi (de.NBI)).

Authors' contributions
All authors contributed to writing Galaxy tools, creating workflows, writing training material, and writing the paper. All authors read and approved the final manuscript.

Funding
This work was supported by funding from the following organizations: S.A.B. was funded by the European Open Science Cloud (EOSC-Life) (Grant No. 824087); T.S. and C.B.B. were funded by the University of Cape Town's Research Committee (URC) and by the National Research Foundation of South Africa (Grant Numbers 115215 and 116362); and B.A.G was funded by the German Research Foundation for the Collaborative Research Center 992 Medical Epigenetics [SFB 992/1 2012 and SFB 992/2 2016].

Data and material availability
Data and materials are available on GitHub:
European Galaxy server (https://cheminformatics.usegalaxy.eu)
Galaxy Computational Chemistry South Africa server (https://galaxy-compchem.ilifu.ac.za)
Docker container providing a Galaxy installation with all required tools preinstalled. (https://quay.io/repository/galaxy/computational-chemistry-training)
Galaxy Training Network website (https://training.galaxyproject.org/topics/computational-chemistry/tutorials/htmd-analysis/tutorial.html)
Supplementary Material, including workflows and data used (https://github.com/galaxycomputationalchemistry/htmd-paper-sm)

Competing interests
The authors declare that they have no competing interests.

Received: 17 May 2020 Accepted: 27 July 2020
Published online: 10 September 2020

References

1. Berendsen HJC (2007) Simulating the physical world: hierarchical modeling from quantum mechanics to fluid dynamics. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511815348
2. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Čech M, Chilton J, Clements D, Coraor N, Grüning BA, Guerler A, Hillman-Jackson J, Hiltemann S, Jalili V, Rasche H, Soranzo N, Goecks J, Taylor J, Nekrutenko A, Blankenberg D (2018) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res 46(Web Server issue):537–544. https://doi.org/10.1093/nar/gky379
3. Senapathi T, Bray S, Barnett CB, Grüning B, Naidoo KJ (2019) Biomolecular Reaction and Interaction Dynamics Global Environment (BRIDGE). Bioinformatics 35(18):3508–3509. https://doi.org/10.1093/bioinformatics/btz107
4. Harvey MJ, Fabritiis GD (2012) High-throughput molecular dynamics: the powerful new tool for drug discovery. Drug Discov Today 17(19):1059–1062. https://doi.org/10.1016/j.drudis.2012.03.017

5. Guterres H, Im W (2020) Improving protein-ligand docking results with high-throughput molecular dynamics simulations. J Chem Inf Model 60(4):2189–2198. https://doi.org/10.1021/acs.jcim.0c00057
6. Abraham MJ, Murtola T, Schulz R, Páll S, Smith JC, Hess B, Lindahl E (2015) GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX. https://doi.org/10.1016/j.softx.2015.06.001
7. Lemkul J (2019) From proteins to perturbed Hamiltonians: a suite of tutorials for the GROMACS-2018 molecular simulation package [article v1.0]. Living J Comput Mol Sci. https://doi.org/10.33011/livecoms.1.1.5068
8. Batut et al (2018) Community-driven data analysis training for biology. Cell Syst 6():752–7581. https://doi.org/10.1016/j.cels.2018.05.012
9. Galaxy Training: Computational chemistry. https://training.galaxyproject.org/training-material/topics/computational-chemistry/tutorials/htmd-analysis/tutorial.html.
10. Pearl LH, Prodromou C (2006) Structure and mechanism of the Hsp90 molecular chaperone machinery. Annu Rev Biochem 75(1):271–294. https://doi.org/10.1146/annurev.biochem.75.103004.142738
11. Schopf FH, Biebl MM, Buchner J (2017) The HSP90 chaperone machinery. Nat Rev Mol Cell Biol 18(6):345–360. https://doi.org/10.1038/nrm.2017.20
12. Stebbins CE, Russo AA, Schneider C, Rosen N, Hartl FU, Pavletich NP (1997) Crystal structure of an Hsp90–geldanamycin complex: targeting of a protein chaperone by an antitumor agent. Cell 89(2):239–250. https://doi.org/10.1016/s0092-8674(00)80203-2
13. Hermane J, Eichner S, Mancuso L, Schröder B, Sasse F, Zeilinger C, Kirschning A (2019) New geldanamycin derivatives with anti Hsp properties by mutasynthesis. Org Biomol Chem 17(21):5269–5278. https://doi.org/10.1039/c9ob00892f
14. PubChem: 3-(2,4-Dihydroxyphenyl)-4-(2-fluorophenyl)-1H-1,2,4-triazole-5-thione. Library Catalog: pubchem.ncbi.nlm.nih.gov. https://pubchem.ncbi.nlm.nih.gov/compound/135508238 Accessed 29 Apr 2020.
15. Rose AS, Bradley AR, Valasatava Y, Duarte JM, Prlić A, Rose PW (2018) NGL viewer: web-based molecular graphics for large complexes. Bioinformatics 34(21):3755–3758. https://doi.org/10.1093/bioinformatics/bty419
16. Schuetz DA, Bernetti M, Bertazzo M, Musil D, Eggenweiler H-M, Recanatini M, Masetti M, Ecker GF, Cavalli A (2018) Predicting residence time and drug unbinding pathway through scaled molecular dynamics. J Chem Inf Model 59(1):535–549. https://doi.org/10.1021/acs.jcim.8b00614
17. Vanommeslaeghe K, Hatcher E, Acharya C, Kundu S, Shim J, Darian E, Guvench O, Lopes P, Vorobyov I, Mackerell AD (2009) CHARMM general force field: a force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. J Comput Chem. https://doi.org/10.1002/jcc.21367
18. Maier JA, Martinez C, Kasavajhala K, Wickstrom L, Hauser KE, Simmerling C (2015) ff14sb: Improving the accuracy of protein side chain and backbone parameters from ff99sb. J Chem Theory Comput 11(8):3696–3713. https://doi.org/10.1021/acs.jctc.5b00255
19. Reif MM, Hünenberger PH, Oostenbrink C (2012) New interaction parameters for charged amino acid side chains in the GROMOS force field. J Chem Theory Comput 8(10):3705–3723. https://doi.org/10.1021/ct300156h
20. Mobley DL, Bannan CC, Rizzi A, Bayly CI, Chodera JD, Lim VT, Lim NM, Beauchamp KA, Slochower DR, Shirts MR, Gilson MK, Eastman PK (2018) Escaping atom types in force fields using direct chemical perception. J Chem Theory Comput 14(11):6076–6092. https://doi.org/10.1021/acs.jctc.8b00640
21. Lemkul JA (2020) Pairwise-additive and polarizable atomistic force fields for molecular dynamics simulations of proteins. In: Computational approaches for understanding dynamical systems: protein folding and assembly. p. 1–71. New York: Elsevier. https://doi.org/10.1016/bs.pmbts.2019.12.009.
22. Onufriev AV, Izadi S (2017) Water models for biomolecular simulations. Wiley Interdiscip Rev Comput Mol Sci 8(2):1347. https://doi.org/10.1002/wcms.1347
23. da Silva AWS, Vranken WF (2012) ACPYPE—AnteChamber PYthon parser interfacE. BMC Res Notes 5(1):367. https://doi.org/10.1186/1756-0500-5-367
24. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR (2011) OpenBabel: an open chemical toolbox. J Cheminf 3(1).
25. Wang J, Wolf RM, Caldwell JW, Kollman PA, Case DA (2004) Development and testing of a general AMBER force field. J Comput Chem 25(9):1157–1174. https://doi.org/10.1002/jcc.20035
26. Swails J, Hernandez C, Mobley D, Nguyen H, Wang L, Janowski P (2016) ParmEd: Cross-program parameter and topology file editor and molecular mechanical simulator engine. Accessed 23 Jan 2020. https://parmed.github.io/ParmEd/html/index.html.
27. Michaud-Agrawal N, Denning EJ, Woolf TB, Beckstein O (2011) MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. J Comput Chem 32(10):2319–2327. https://doi.org/10.1002/jcc.21787
28. McGibbon R, Beauchamp K, Harrigan M, Klein C, Swails J, Hernández C, Schwantes C, Wang L-P, Lane T, Pande V (2015) MDTraj: a modern open library for the analysis of molecular dynamics trajectories. Biophys J 109(8):1528–1532. https://doi.org/10.1016/j.bpj.2015.08.015
29. Skjærven L, Yao X-Q, Scarabelli G, Grant BJ (2014) Integrating protein structural dynamics and evolutionary analysis with Bio3D. BMC Bioinf 15(1):399. https://doi.org/10.1186/s12859-014-0399-6
30. Berjanskii M, Wishart DS (2006) NMR: prediction of protein flexibility. Nat Protoc 1(2):683–688. https://doi.org/10.1038/nprot.2006.108
31. Kuzmanic A, Zagrovic B (2010) Determination of ensemble-average pairwise root mean-square deviation from experimental B-factors. Biophys J 98(5):861–871. https://doi.org/10.1016/j.bpj.2009.11.011
32. Humphrey W, Dalke A, Schulten K (1996) VMD—visual molecular dynamics. J Mol Graph 14:33–38
33. galaxycomputationalchemistry/htmd-paper-sm: Data and workflows—intuitive, reproducible high-throughput molecular dynamics in Galaxy: a tutorial. Zenodo. 2020. https://doi.org/10.5281/zenodo.3813283
34. Galaxy | Europe | Accessible history | Protein-ligand HTMD simulation. https://cheminformatics.usegalaxy.eu/u/sbray/w/protein-ligand-htmd-sim. Accessed 29 Apr 2020.
35. Galaxy | South Africa | Accessible History | Protein-ligand HTMD analysis. https://galaxy-compchem.ilifu.ac.za/u/sbray/w/protein-ligand-htmd-sim. Accessed 29 Apr 2020.
36. Galaxy | Europe | Accessible History | Protein-ligand HTMD analysis. https://cheminformatics.usegalaxy.eu/u/sbray/w/protein-ligand-htmd-analysis. Accessed 29 Apr 2020.
37. Galaxy | South Africa | Accessible History | Protein-ligand HTMD analysis. https://galaxy-compchem.ilifu.ac.za/u/sbray/w/protein-ligand-htmd-analysis. Accessed 29 Apr 2020.
38. Galaxy Training: Collections: Using dataset collection. https://galaxyproject.github.io/training-material/topics/galaxy-data-manipulation/tutorials/collections/tutorial.html. Accessed 29 Apr 2020.
39. Trott O, Olson AJ (2009) AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multi-threading. J Comput Chem. https://doi.org/10.1002/jcc.21334
40. Ruiz-Carmona S, Alvarez-Garcia D, Foloppe N, Garmendia-Doval AB, Juhos S, Schmidtke P, Barril X, Hubbard RE, Morley SD (2014) rDock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids. PLoS Comput Biol 10(4):1003571. https://doi.org/10.1371/journal.pcbi.1003571
41. Galaxy | Europe | Accessible History | Protein-ligand docking (6hhr). https://cheminformatics.usegalaxy.eu/u/sbray/h/protein-ligand-docking-6hhr. Accessed 29 Apr 2020.
42. Workflows: Extracting Workflows from Histories. https://galaxyproject.github.io/training-material/topics/galaxy-ui/tutorials/history-to-workflow/tutorial.html. Accessed 29 Apr 2020.
43. Galaxy Training: Histories: Understanding Galaxy history system. https://galaxyproject.github.io/training-material/topics/galaxy-ui/tutorials/history/tutorial.html. Accessed 29 Apr 2020.
44. Galaxy Training: Scripting Galaxy using the API and BioBlend. https://training.galaxyproject.org/training-material/topics/dev/tutorials/bioblend-api/slides.html. Accessed 29 Apr 2020.
45. Sloggett C, Goonasekera N, Afgan E (2013) BioBlend: automating pipeline analyses within Galaxy and CloudMan. Bioinformatics 29(13):1685–1686. https://doi.org/10.1093/bioinformatics/btt199

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

4 Benchmarking the dcTMD method against the T4 lysozyme L99A mutant

The work described in this chapter is unpublished and is therefore described more comprehensively than the other chapters of this thesis.

4.1 Introduction

4.1.1 Background

This work aimed to benchmark the dcTMD method, in particular its implementation in the Galaxy platform, against T4 lysozyme, a well-known test system for free energy calculations. Lysozyme is an enzyme capable of decomposing the peptidoglycan polymers found in bacterial cell walls by hydrolysing the glycosidic bonds, and as a result has an important role in the animal immune system. T4 lysozyme (T4L) is an analogous, but non-homologous, protein produced by the Escherichia virus T4 [93], a bacteriophage which infects Escherichia coli (E. coli) bacteria; the lysozyme is used to attack the bacterial cell wall prior to entry. T4L is a small, stable globular protein, and has been used extensively as a model system to study protein folding [94, 95]. As a result, during the course of the 1990s, hundreds of T4L mutants were engineered and the effect on folding and stability analysed [96].

Structurally, T4L consists of two domains (Figure 4.1). The C-terminal domain is formed by 5 roughly parallel α-helices, connected by short loop or helical regions, while the N-terminal domain is formed of a mix of α, β and loop secondary structures. The active site, where the peptidoglycan molecule binds and is hydrolysed, is located between the domains. The domains are linked by a flexible loop region, which allows the domains to move relative to each other, acting as a hinge. Thus, the protein can exist in two conformational forms, "open" and "closed". Transition to the closed state is triggered by substrate binding, in a "Pacman-like" mechanism.

One of the T4L mutants was produced by replacing the leucine residue at position 99 with alanine to create the T4L-L99A mutant [97, 98]. L99 is buried in the C-terminal "head" of the protein, and the replacement of the bulky isobutyl by a methyl group entailed by the mutation to alanine results in the creation of a small hydrophobic pocket, about 150 Å^3 in volume [98], which is enough to bind small hydrophobic molecules such as benzene or toluene, as well as nonpolar gas molecules such as xenon or dioxygen. The creation of the cavity destabilises the protein fold, as its hydrophobic nature prevents it being filled by a water molecule. As a result, cavity filling by the binding of benzene or a chemically similar ligand is thermodynamically favourable.

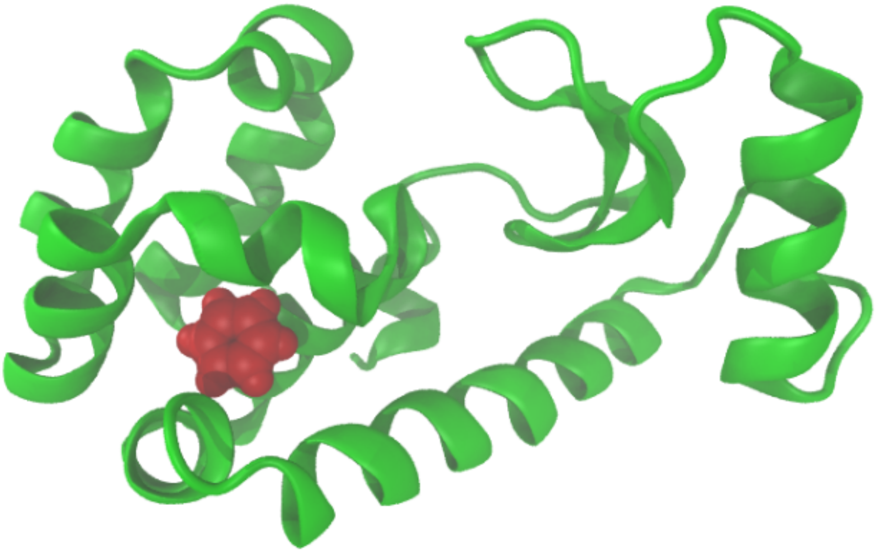


Fig. 4.1: Cartoon depiction of the T4L-L99A mutant. The C-terminal domain can be seen on the left, with benzene bound in the hydrophobic cavity. The hydrolytic active site is positioned towards the top of the image, between the two domains. Image generated using VMD [99].

4.1.2 Motivation

The T4L-L99A-benzene system is an excellent model system for studying fragment-protein binding, for a number of reasons. Firstly, the size of the pocket and the fragments which are capable of binding there are typical for the compounds generally used in fragment-based screening studies (low molecular weight, at most a few hundred daltons; benzene has a molecular weight of 78 Da). Secondly, the chemistry of the pocket is uniform; the entire pocket is hydrophobic. Thirdly, while binding of benzene to T4L-L99A is thermodynamically favourable, the strength of the binding is relatively weak; this is typical for protein-binding fragments, which generally bind weakly with one or two individual molecular interactions.

Choosing benzene as a model ligand also reduces the complexity of the analysis significantly, due to the high level of molecular symmetry; benzene belongs to the D_{6h} symmetry group, meaning that it possesses an axis of rotation with 6-fold rotational symmetry, together with a second axis of rotation and a plane of reflectional symmetry perpendicular to the main axis. As a consequence of this high level of symmetry, the problem of finding the correct binding pose within the cavity is greatly simplified. As will be discussed, separation of trajectories into pathways is an essential part of the dcTMD procedure used in this study; less symmetrical ligands (for example, toluene) create new complexity, since each of the 6 possible rotations requires separate consideration.

The T4L-L99A protein has another peculiarity which influenced its choice as a model system; the cavity in the head domain is surrounded by multiple α-helices, and thus multiple exit routes are available to the bound ligand. It is well-known that protein-ligand dissociation can in many cases follow multiple so-called pathways and that each of these may possess different characteristics. In particular, as the height of the energetic barrier to dissociation is pathway-dependent, the kinetics of dissociation are likely to vary dramatically. Computational studies of T4L-L99A have identified multiple pathways for ligand dissociation from the hydrophobic cavity. One study claims to identify 8 different pathways [100]; at least 4 are unambiguously distinguishable and attested by multiple studies. [101, 102, 100, 103]. Thus, it is a good choice of system for benchmarking dcTMD, a system for which consideration of dissociation pathways is essential, as will be elaborated in the Methods section. The most thorough review was performed by Nunes-Alves et al. [104, 105] and the discussion here will follow the nomenclature for pathway description introduced there (Figure 4.2).

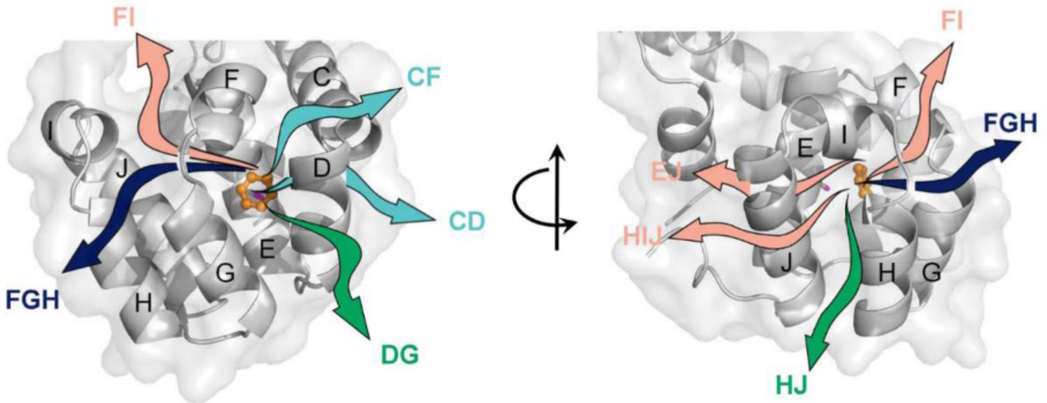


Fig. 4.2: Cartoon depiction of the T4L-L99A C-terminal domain viewed from two angles, with helices labelled from C to J. Pathways observed in computational studies are marked using arrows, and labelled according to the helices between which the pathway travels; for example, pathway HJ passes between helices H and J. Arrow colouring represents the frequency with which pathways have been reported in the literature: blue = most frequent, green = frequent, cyan = reported at least twice, pink = reported only by [100]. Information on the image source is provided in the List of Figures.

4.2 Methods

4.2.1 Dissipation-corrected targeted molecular dynamics

As discussed in Chapter 2, dissipation-corrected targeted molecular dynamics (dcTMD) can be used to generate free energy and friction profiles for the process of protein-ligand dissociation; from these, Langevin simulations can be used to deduce the kinetics of the unbinding process.

Targeted molecular dynamics (TMD) is a non-equilibrium technique which introduces a constraint force to separate two groups of atoms at a steady velocity.

A holonomic constraint function

$$\Phi(x(t)) = \sum_i x_i(t) - x_{c_i}(t) = 0 \tag{4.1}$$

is used to subject each atom i to a constraining force

$$f_{c_i} = \lambda \nabla_i \Phi = \lambda \tag{4.2}$$

Here λ represents a Lagrange multiplier, implemented using an integrator such as the leapfrog propagator.

TMD simulations produce values for the force f applied at each step to increase the constraint distance x_c, which can be integrated to give a value for the work W required to remove the ligand from the protein binding site. Unfortunately, this value does not relate exactly to the free energy - we know only that

$$W = \Delta G + W_{diss} \tag{4.3}$$

where the dissipative work W_{diss} is of unknown size. However, if we have an ensemble of TMD simulations, Jarzynski's equation can be applied to calculate an equilibrium value of ΔG directly from the nonequilibrium simulations:

$$\Delta G = -k_B T \ln \langle e^{-W/k_B T} \rangle \tag{4.4}$$

where W represents the work profile for a trajectory and $\langle ... \rangle$ is an ensemble average over all trajectories.

The dcTMD calculations do not use the Jarzynski equality directly, but instead employ a truncated cumulant expansion

$$\ln \langle e^x \rangle = \langle x \rangle + (\langle x^2 \rangle - \langle x \rangle^2)/2 + ... \tag{4.5}$$

which is truncated after two terms, giving

$$\Delta G = \langle W \rangle - \frac{\langle \delta W \rangle}{2 k_B T} \tag{4.6}$$

where $\delta W = W - \langle W \rangle$. The first term can be equated with the work exerted by the constraint force, while the second term represents the dissipated work. The nonequilibrium TMD simulation can be described by a modified Langevin equation

$$m\ddot{x}(t) = -\frac{dG}{dx} - \Gamma(x)\dot{x} + \mathcal{K}(x)\xi(t) + f_c(t) \tag{4.7}$$

containing a Newtonian force term $f = \frac{dG}{dx}$, a friction term Γ, a stochastic force ξ which averages to zero, and the constraint force f_c. An equilibrium average yields after integrating both sides

$$\Delta G(x) = \langle W(x) \rangle - v_\mathrm{c} \int_{x_0}^{x} \Gamma(x')\, \mathrm{d}x'. \tag{4.8}$$

where the second term can be equated with the dissipative work. The truncated cumulant expansion described above allows the expression of W_{diss} as

$$W_\mathrm{diss}(x) = \delta W^2(x)/kT \tag{4.9}$$

and by relating fluctuations in W to force fluctuations δf_c the friction

$$\Gamma(x) = \frac{1}{kT} \int_{t_0}^{t(x)} \delta f_c(t) \delta f_c(t') t', \tag{4.10}$$

can be obtained from the TMD simulations.

4.2.2 T-boosting

Modelling the dissociation by propagating the Langevin equation over the free energy and friction profiles obtained from dcTMD already represents a dramatic reduction in the compute resources required, compared to molecular dynamics, since the number of coordinates which need to be considered is reduced from $3N$ to 2 degrees of freedom (where N represents the number of atoms in the system). Nonetheless, the Langevin simulations require a time step of similar length to MD (on the order of femtoseconds) with the consequence that accessing second-length simulations remains impractical.

So-called “T-boosting“ can be used in conjunction with dcTMD to circumvent this issue. The following transition state expression relates the transition rates k_1 and k_2 at temperatures β_1 and β_2

$$k_2 = k_1 e^{-\Delta G^{\neq}(\beta_2 - \beta_1)}, \tag{4.11}$$

Thus, Langevin simulations can be performed using the dcTMD profiles calculated at a biologically realistic temperature (ca. 300K), but boosted to a much higher temperature, increasing the sampling of transitions between the bound and unbound state to allow calculation of the transition rate. The equation above can then be used to calculate the transition rate at the desired temperature. In fact, T-boosting can be conducted at multiple temperatures, plotted to identify a relationship

Tab. 4.1: Pulling groups

Index	Helices	Number of simulations
1	CDEG	535
2	EG	342
3	CDEGHJ	153
4	DEGHJ	50
5	EJ	49
6	EFGHIJ	49

between temperature and transition rate, and an extrapolation made to the biological temperature.

4.2.3 Definition of pulling groups

In order to implement the constraint force, two groups of atoms were defined for each group of simulations and the force applied to the atoms to separate the two groups at constant velocity. One of these groups consisted of ligand atoms, while the other consisted of a selection of atoms from the protein C-terminal domain. The effect of applying the constraint force in such a way was to pull the ligand out of the binding site; thus, the atom groups defined will be referred to here as pulling groups. The ligand pulling group was kept constant for all simulations, but the protein pulling group was varied with the aim of sampling as many different pathways as possible (Figure 4.3).

A single pair of pulling groups is insufficient to sample all of the dissociation pathways, so the protein pulling group was varied, while the ligand pulling group was left unchanged. For the protein pulling group, 6 different combinations of atoms were tested, summarised in Table 4.1 and Figure 4.4. At least 49 simulations were performed for each pulling group. Some of these did not yield new dissociation pathways, so were not sampled further.

4.2.4 Simulation

Simulations were performed using the open-source GROMACS software (version 2019.1) using the AMBER99SB forcefield [18] and the TIP3P water model [106].

The topology for benzene was created with antechamber and acpype [107] using GAFF parameters [108] and BCC charges [109]. The T4L-L99A protein crystal structure was centred in a cubic box with side length of around 7 nm. The system

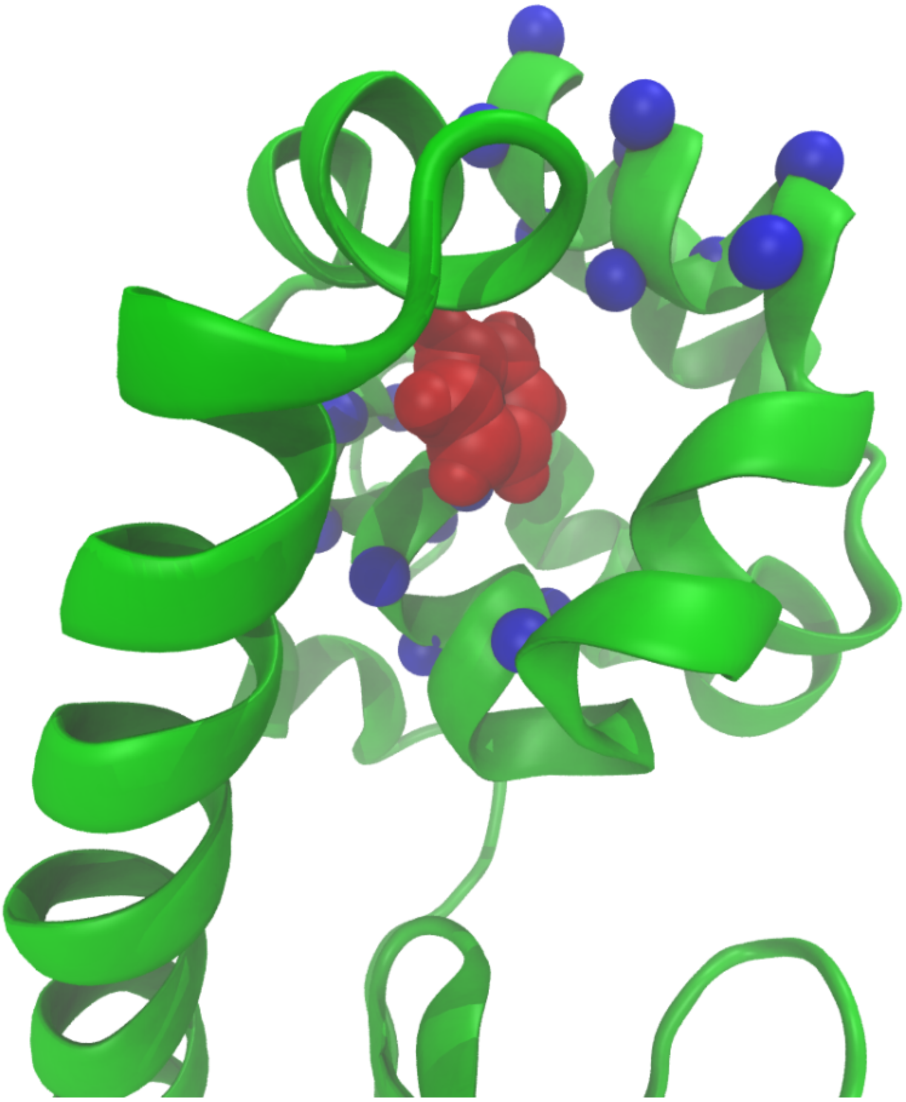
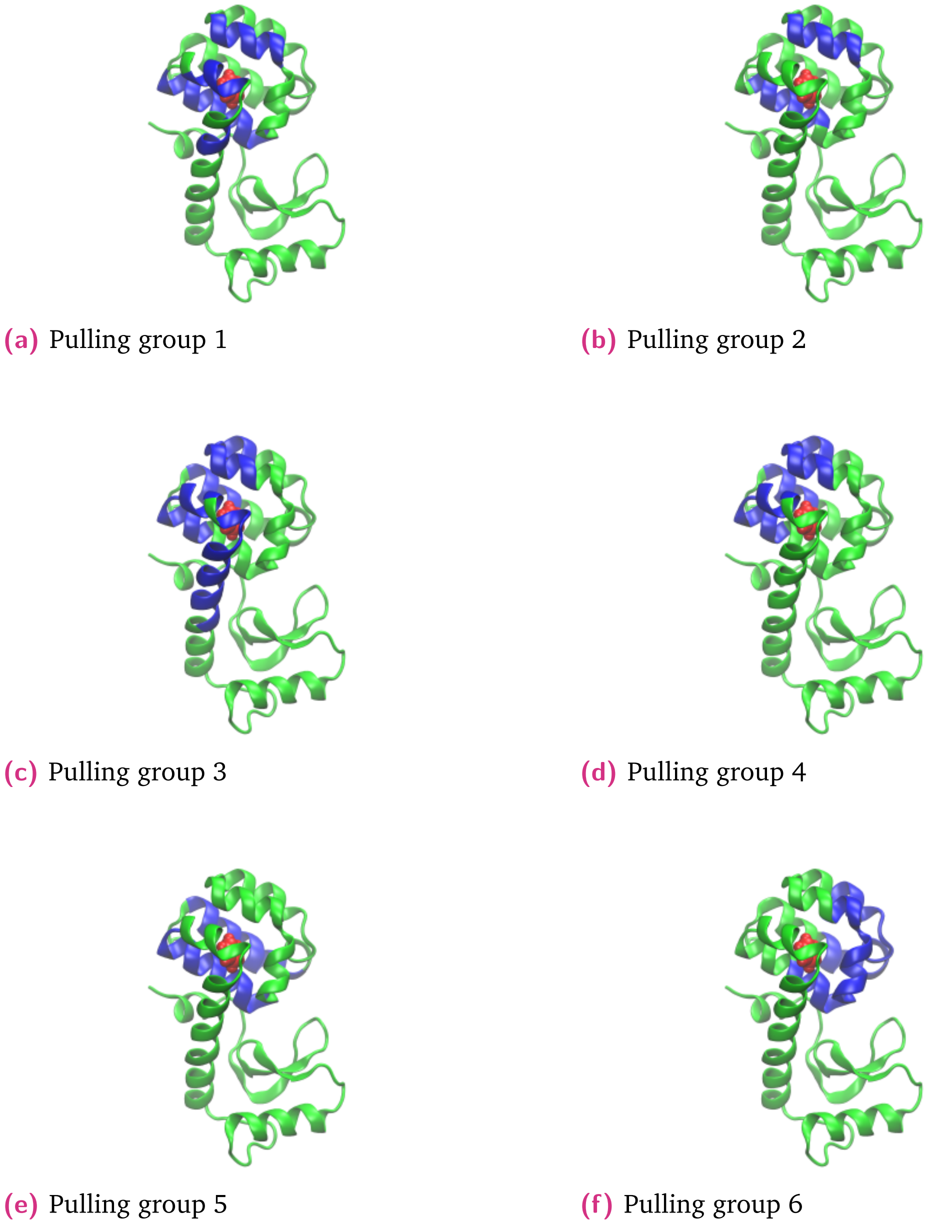


Fig. 4.3: Cartoon depiction of the T4L-L99A C-terminal domain with pulling groups marked for a particular simulation; red atoms represent the ligand (benzene), whereas atoms marked in blue belong to the protein main chain. The protein pulling group can be modified by adding or removing atoms, with the aim of influencing the dissociation path taken during the TMD simulation. In the example depicted, the pulling group consists of atoms selected from helices E and G, in order to force the ligand out from the opposite site of the protein domain, along paths DG, CD and CF (cf. Figure 4.2). Image generated using VMD [99].



(a) Pulling group 1

(b) Pulling group 2

(c) Pulling group 3

(d) Pulling group 4

(e) Pulling group 5

(f) Pulling group 6

Fig. 4.4: Pulling groups. Images generated using VMD [99].

was protonated at pH 7, solvated, and chloride ions added to ensure a charge-neutral simulation box. PME electrostatics [110] (minimal real space cut-off of 1 nm) were used during simulations, with a van der Waals cut-off of 1.2 nm. The LINCS algorithm [111] was used to constrain bonds involving hydrogen atoms. Initially, energy minimisation was performed using the steepest descent integrator and fast smooth PME electrostatics, followed by a 10 ns equilibration under the NVT ensemble.

After preparation of a minimised and equilibrated structure, ensembles of nonequilibrium TMD pulling simulations were performed for each pulling group defined, using the GROMACS PULL code in constraint mode. First a further NPT equilibration simulation was performed for 100 ps, without pulling to ensure slightly different starting coordinates for each simulation. The equilibration runs were continued for 2 ns under the NPT ensemble at 300 K and 1 bar, using the Nosé-Hoover thermostat [112] and Parrinello-Rahman barostat [113], with a fixed constraint velocity of 1 nm/ns and an integration step size of 2 fs. Values for the constraint pseudoforces were saved at each time step for use in dcTMD calculations.

4.2.5 Path separation

In order to prevent overestimation of friction (which relates to the second term of the cumulant expansion of the Jarzynski equality) methods must be employed for pathway identification and if necessary separation, to prevent combining trajectories from multiple pathways in a single analysis.

Initially, contact PCA (conPCA) [114] is used to obtain an insight into the routes taken by the ligand out of the protein cavity. To perform the clustering of the ensemble into pathways itself, a distance-based clustering method is employed. A function must be chosen which is capable of describing the similarity of two trajectories at a particular time point; a good choice is the RMSD (root mean squared distance) measure. RMSD is based on a comparison of the atomic coordinates of a molecular structure in two different positions, after superimposing all frames from the two trajectories using the C-α atoms of the protein:

$$f(x_1, x_2, t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_2(i,t), x_1(i,t))^2} \tag{4.12}$$

where N is the number of heavy atoms in the structure and δ_i the distance between the coordinates of the atom i in the two structures x_1 and x_2. Here we are dealing with two trajectories with a time component, represented by an additional variable t.

A three-dimensional tensor $M = \{m_{i,j,t}\}$ can be constructed by calculating all elements $m_{i,j,t} = f(x_i, x_j, t)$ for every trajectory. In order to perform clustering on this data, the time axis of the tensor must be removed by averaging over all frames to give a distance matrix $D = \{d_{i,j}\}$.

$$d_{i,j} = \frac{\sum_{t=0}^{tmax} m_{i,j,t}}{t_{max}} \tag{4.13}$$

Each element $d_{a,b}$ of D describe how similar the pair of trajectories a and b are according to the RMSD measure. They give an indication of how similar the ligand's motion and conformation is between the two trajectories, relative to the protein coordinates. Thus, the smaller the value of $d_{a,b}$, the more likely that a and b follow the same path.

Once the distance matrix D has been calculated, the next step is to perform hierarchical clustering. Several algorithms are available for this; one of the most widely used is UPGMA (Unweighted Pair Group Method with Arithmetic Mean), which calculates a dendrogram (or phylogenetic tree) from a distance matrix [115, 116].

UPGMA employs an iterative approach, starting with the individual nodes (trajectories in this case) and merging them together into larger clusters in a stepwise fashion. The two nodes i and j for which $d_{i,j}$ is minimised (i.e, the two closest nodes are combined into a cluster, which forms a new node k. Before iterating, distances for the new node k must be calculated and the matrix D updated, which is done according to the following formula:

$$d(k,l) = \frac{d(i,l)|C_i| + d(j,l)|C_j|}{|C_i| + |C_j|} \tag{4.14}$$

where C_k is the union of C_i and C_j, and C_l is another node. These two steps (combination of nodes and recalculation of distances) are now repeated until all nodes have been replaced by a single cluster.

UPGMA is a simple clustering approach and is ultrametric. This is a good approximation, as the change in interatomic distances recorded in D is produced artificially by the TMD constraint force and is thus known to be constant. In previous work, the neighbor-net algorithm, originally developed for finding phylogenetic networks [117, 118] was employed [119]; the circular dendrograms produced by neighbor-net have the advantage that they display the ambiguity which results from uncertainty or inaccuracies in the input dataset. The disadvantage of neighbor-net is the increased complexity of the analysis and the human intervention required to decide on suitable clusters. UPGMA is thus a good initial choice for performing pathway separation; if random sampling of trajectories indicates the quality of the clustering is poor, neighbor-net can be employed.

4.3 Results

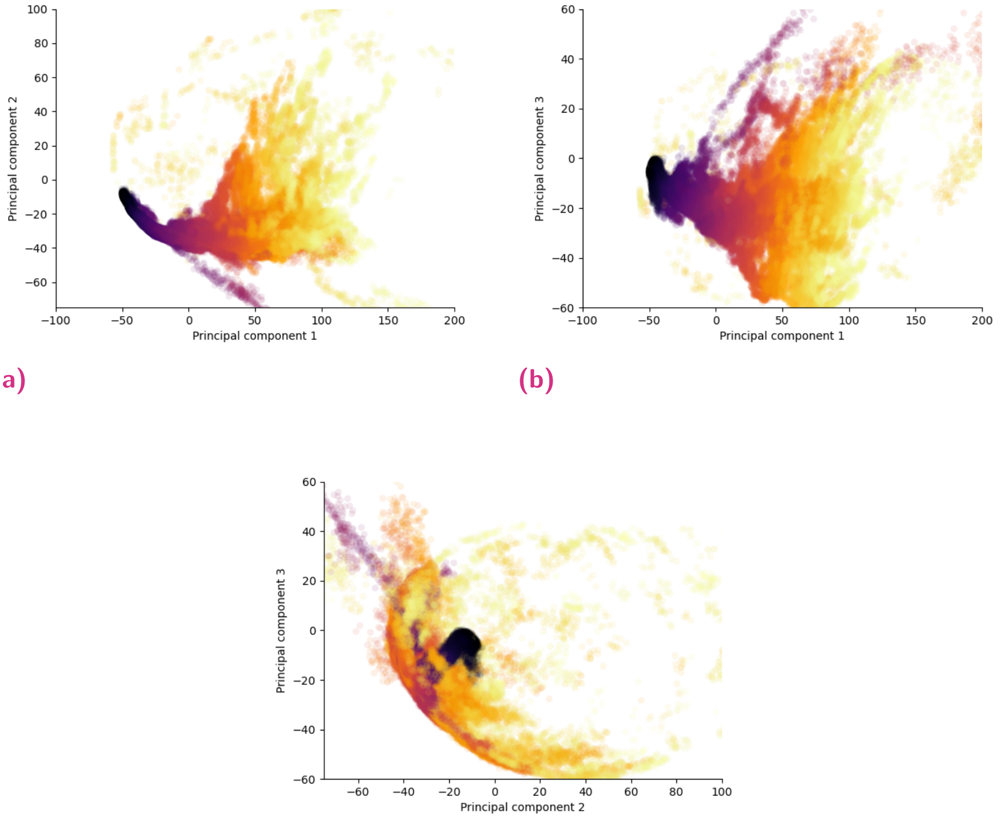
4.3.1 Dimensionality reduction

In order to gain an initial insight into the performed trajectories, contact principal component analysis (conPCA) was performed [114]. Input data for the PCA were selected protein-ligand contact distances. For all residues with IDs between 71 and 156 inclusive, the distance between the residue and ligand centre of masses were calculated for all points during the duration of the trajectories using MDAnalysis [79]. 20 trajectories were chosen randomly for each pulling group and concatenated, before performing PCA using the FastPCA software [120].

The remainder of the trajectories were then projected onto the eigenvectors created by the PCA and plots of the first three principal components were created for each pulling group. An example is shown in Figure 4.5.

The first point which is apparent from Figure 4.5 is that PC1 correlates strongly with the direction of pulling; points at the start of the trajectory have values of $PC1 \approx -50$, increasing to between 100 and 150 by the end. This is similar to observations made in previous dcTMD studies [121] and is generally expected, as the constraint force is responsible for the largest changes in ligand motion during the trajectories, but is rather surprising in this case as the PCA is derived by combining trajectories from multiple pulling groups. Thus, in contrast to previous studies, there is no single pulling direction; nonetheless, the conPCA method conflates each of the six directions and appears to represent all of them by PC1. The pathway taken by each trajectory, which is orthogonal to the pulling direction, is thus better represented by the second and third PCs (Figure 4.6).

As can be seen, each of the pulling groups results in different pathways being followed, from a central point at approximate coordinates $PC2, PC3 = (0, 0)$ to an outer ring. Inspecting MD trajectories reveals that the outer ring visible in the plots of PC 2 and 3 in Figure 4.6 represents diffusion of the ligand over the surface of

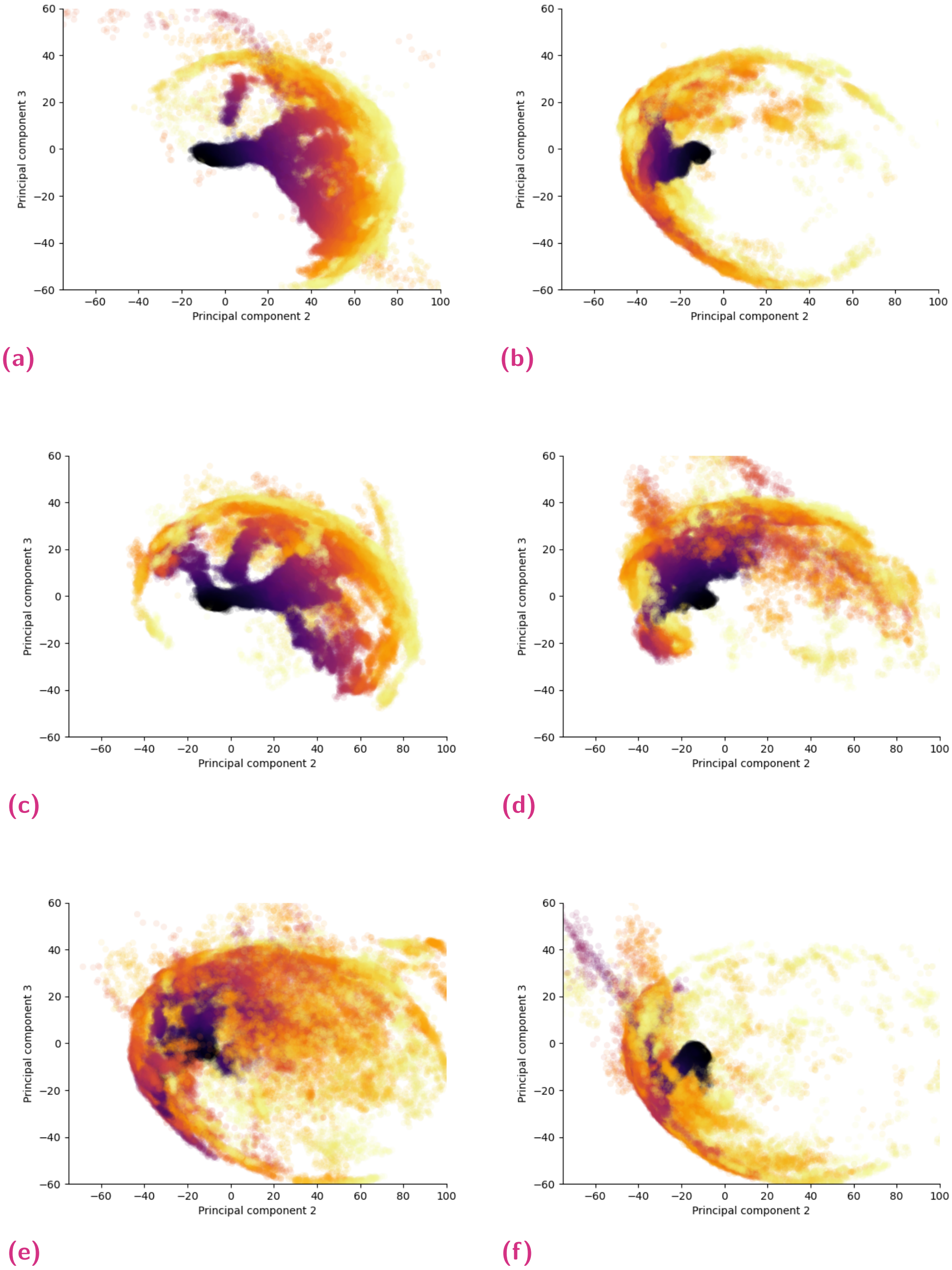


(a)

(b)

(c)

Fig. 4.5: Plots of the first 3 principal components for pulling group 6. Darker colors represent points at the beginning of the trajectories; lighter points those nearer to the end.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 4.6: Plots of PC 2 and 3 for each of the pulling groups.

the protein, once it has left the binding cavity. Thus the crescent shape visible for example for pulling group 2 (Figure 4.9b) should not be viewed as two different pathways, but rather two different diffusion routes that the ligand can take over the surface of the protein, once the dissociation process itself is complete.

Figure 4.7 shows individual trajectories from the pathways observed, projected onto principal components 2 and 3.

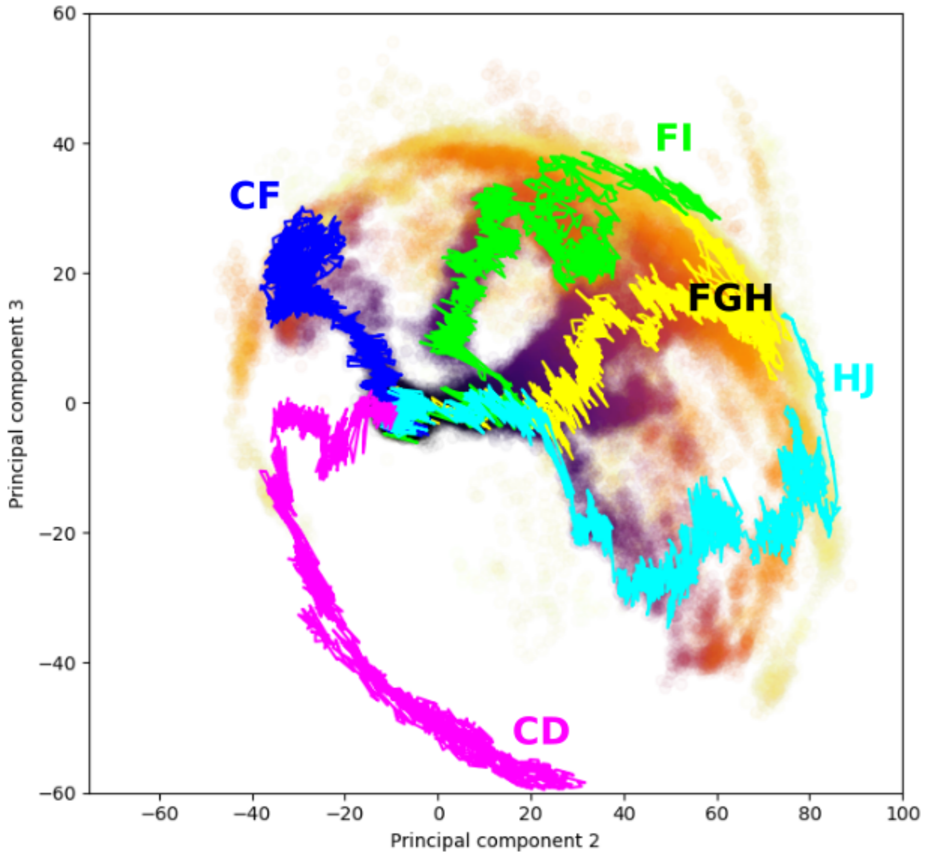


Fig. 4.7: Sample TMD trajectories projected onto principal components 2 and 3 and overlaid onto Figure 4.6c. The trajectory following path CD is derived using pulling group 2; all other trajectories are derived using pulling group 3.

4.3.2 Trajectory clustering into pathways with UPGMA

Because of the large number of simulations performed, manual classification of trajectories to a pathway based on visual inspection of the PCA plots for each trajectory was not realistic. As an alternative, UPGMA clustering of the trajectories for each pulling group was performed based on ligand RMSD, as described in the Methods section; results are summarised in Table 4.2 and an example dendrogram produced for pulling group 2 depicted in 4.8. Members of each cluster were then randomly selected and inspected to determine the correspondence between each cluster and the paths observed.

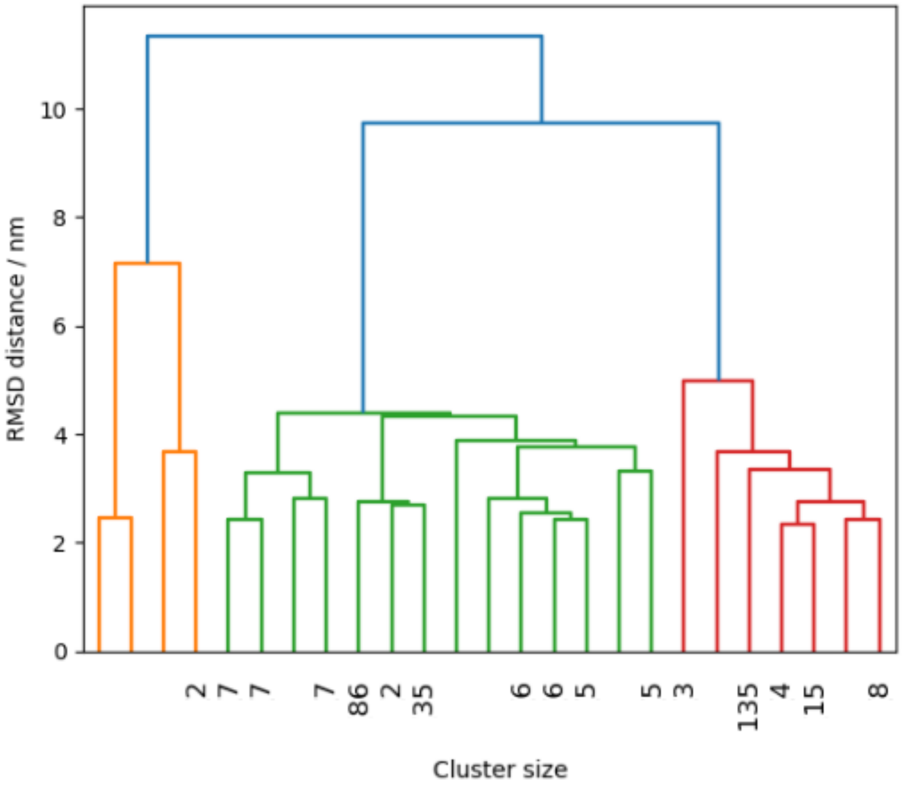
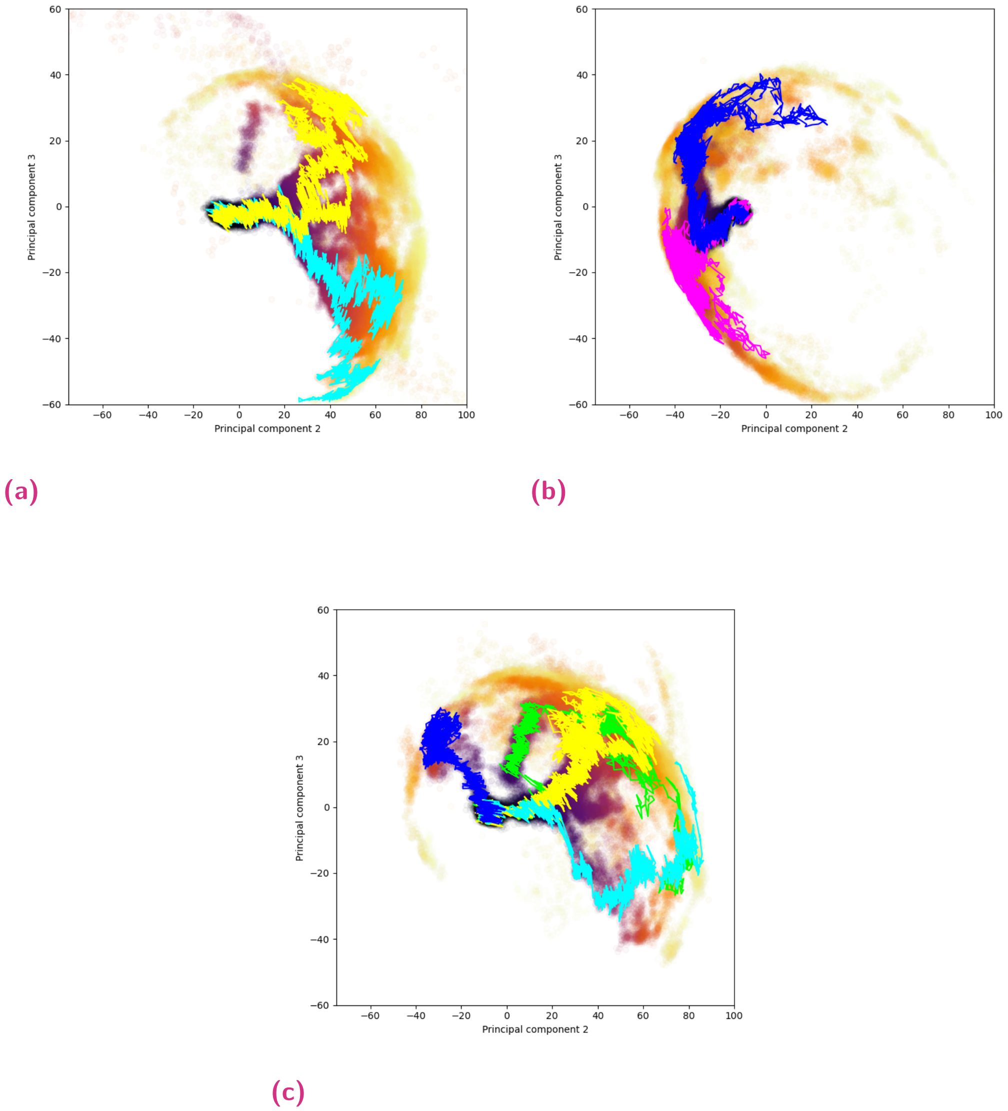


Fig. 4.8: Dendrogram created by RMSD-based hierarchical clustering with UPGMA of trajectories generated using pulling group 2. As the full dendrogram contains 342 nodes, it is truncated here to show only the last 25 merged clusters. Numbers represent the population of truncated subclusters. Two large subclusters can be easily discerned, which correspond to paths CD (green) and CF (red). The remaining 5 trajectories (orange) belong to other paths and are discarded.

For pulling groups 1, 2 and 3, Figure 4.9 provides the PC2/3 plots, with sample trajectories from each cluster superimposed.



(a)

(b)

(c)

Fig. 4.9: PC2/3 plots for pulling groups 1 (top left), 2 (top right) and 3 (bottom), with sample trajectories from each cluster (i.e. pathway) superimposed. Yellow represents FGH, cyan represents HJ, blue represents CF, magenta represents CD and green represents CI.

Tab. 4.2: Clusters found by UPGMA for each pulling group with a population of at least 5% of the ensemble.

Pulling group	Path	Number of trajectories
1	FGH	249
1	HJ	273
2	CF	167
2	CD	170
3	FGH	103
3	FI	23
3	HJ	15
3	CF	10

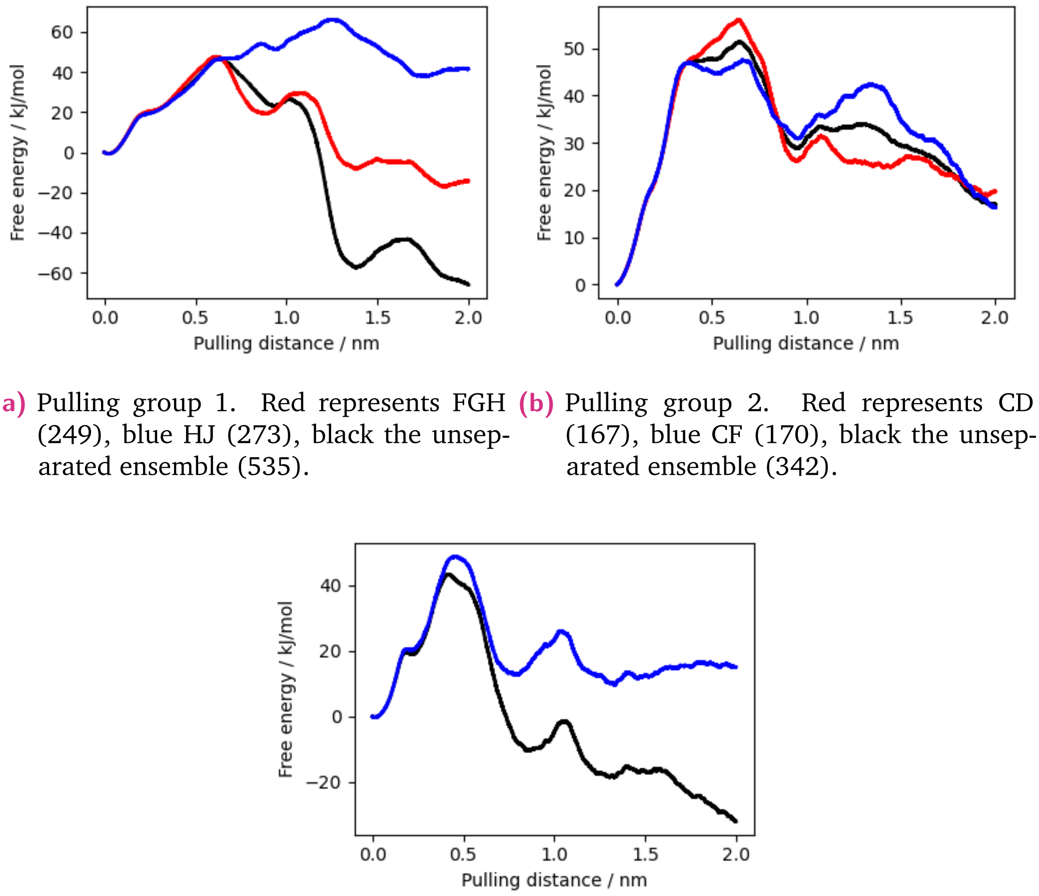
4.3.3 dcTMD calculations

dcTMD calculations were performed for each of the following classes: FGH and HJ (pulling group 1), CD and CF (pulling group 2) and FGH (pulling group 3). Other paths were not sufficiently populated to perform dcTMD analysis. Results are depicted in Figure 4.10.

The free energy profiles in Figure 4.10 show that pathway separation is essential for pulling groups 1 and 3. When the dcTMD method is applied to the entire ensemble, in both cases a serious artefact is observable in the second half of the profile due to friction overestimation. By contrast, after applying path separation, the artefact disappears completely for HJ (pulling group 1) and FGH (pulling group 3), and partially disappears for FGH (pulling group 1), being now only visible starting from around 1.2 nm from the binding site. For pulling group 2, no artefact is visible, but this is attributable to the fact that the free energy profiles for paths CD and CF are very close to one another; if the work profiles for the two paths are almost identical, a large artefact should not be expected if the path separation step is skipped.

4.3.4 Resolving paths for pulling group 1

The main unresolved point after this analysis is the FGH path for pulling group 1, which even after an initial pathway separation continues to display a friction overestimation artefact. It was already observed from projections of the trajectory onto the derived principal components that the FGH cluster obtained from UPGMA was quite diverse. Thus, subclusters revealed by the dendrogram produced by UPGMA was inspected in more detail. Three main subclusters were visible, with populations of 138, 71 and 28; these will be denoted as FGH, FGH' and the "unphysical path" respectively. Two smaller clusters with populations of 12 and 6 were not considered



(a) Pulling group 1. Red represents FGH (249), blue HJ (273), black the unseparated ensemble (535).

(b) Pulling group 2. Red represents CD (167), blue CF (170), black the unseparated ensemble (342).

(c) Pulling group 3. Blue represents FGH (103), black the unseparated ensemble (153).

Fig. 4.10: Free energy plots for paths observed for each TMD pulling group. Number in parentheses indicated the number of trajectories included in each analysis.

further. Projections of sample trajectories into the principal component space, as well as visualisations using VMD [99], are depicted in Figure 4.11.

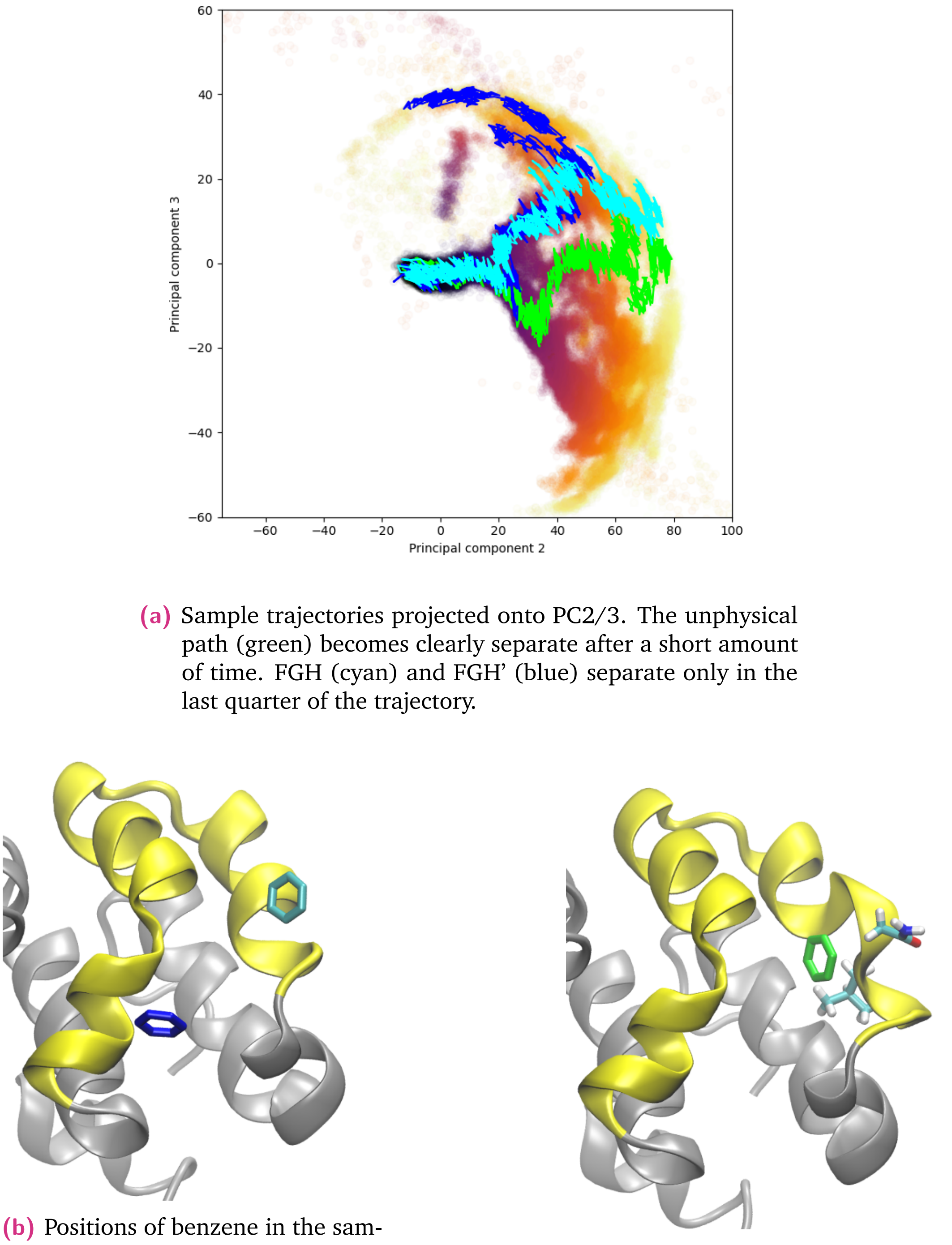
Inspection of the trajectories using VMD (Figure 4.11b) revealed that FGH' trajectories differ from the main FGH path only during the last quarter of the trajectories; after the benzene ligand has bypassed the narrow gap between helices G and H, it is free to migrate either around the outside of helix H, or over helix I, (the main FGH route) or to move in the opposite direction, towards the F helix (FGH'). The unphysical path contains trajectories in which benzene is unable to navigate successfully around helix H, but becomes caught between the sidechains of Asn132 and Leu133, resulting in the constraint force causing a large deformation of helix H (Figure 4.11c).

As a result of this analysis, the unphysical trajectories were discarded and dcTMD calculations repeated for the FGH and FGH' subclusters. Results are depicted in Figure 4.12. As the friction artefact is no longer visible, the result can be considered a success; FGH shows a very similar free energy profile to the FGH profile produced using pulling group 3. For FGH' the profile also has a very similar shape, though the peak at 1.0 nm from the binding site is around 10 kJ/mol higher.

Overall, therefore, by using carefully chosen pulling groups, 4 of the 5 pathways previously observed in multiple studies could be reproduced using TMD simulations. For each of these, after performing RMSD-based hierarchical clustering and experimenting with different cut-off values, reasonable free energy profiles could be calculated for each. Pathway DG was the only major pathway that could not be observed, despite attempts to fine-tune the pulling groups (pulling groups 4, 5 and 6, which failed to produce DG and were not used further).

4.3.5 Langevin simulations

As the next step, Langevin simulations were performed on the free energy and friction profiles generated by the dcTMD calculations, according to the T-boosting procedure previously described. For each of the 6 clusters identified, Langevin simulations were performed at 7 temperatures, at 100 K intervals between 400K and 1000 K. Experiments showed that kinetics followed is independent of ligand mass, and thus follows the Kramers' reaction-rate theory [122], so the benzene mass is artificially increased tenfold to 0.78 kg/mol. The number of transitions is recorded for each Langevin trajectory and a regression line calculated to extrapolate to the expected kinetics at 293.15 K. An example (for pulling group 6) is provided for path CF in Figure 4.13. Results for all paths are provided in Table 4.3.



(a) Sample trajectories projected onto PC2/3. The unphysical path (green) becomes clearly separate after a short amount of time. FGH (cyan) and FGH' (blue) separate only in the last quarter of the trajectory.

(b) Positions of benzene in the sample trajectories for FGH (cyan) and FGH' (blue) at 1.7 nm from the binding site.

(c) Position of benzene in the unphysical path at 1.3 nm from the binding site.

Fig. 4.11: Sample trajectories from the FGH (cyan), FGH' (blue) and unphysical (green) paths. Above: projections in the PCA space. Below: Trajectories visualised in VMD, with helices FGH highlighted in yellow and Asn132 and Leu133 shown in a stick representation. Note the distortion of helix H during the unphysical path. Images generated using VMD [99].

Tab. 4.3: Kinetic data from Langevin simulations. Experimental values (at 293.15 K) are taken from Feher et al. [123]

Path	k_on (s$_{-1}$ M$_{-1}$)	k_off (s$_{-1}$)	K_D (M)
FGH pulling group 1 (purple)	6100	124	0.0203
FGH' pulling group 1 (pink)	23.5	14.2	0.606
FGH pulling group 3 (blue)	1730	44.1	0.0255
HJ pulling group 1	112000	0.5	0.00000446
CD pulling group 2	9420	3.31	0.000352
CF pulling group 2	73500	29	0.000394
Experimental	800000 - 100000	600 - 1000	0.00068 - 0.00082

Experimental rate constants for the (un)binding of benzene and T4L-L99A are somewhat higher than the dcTMD values. Assuming the ligand binds and debinds via the faster path (HJ and FGH respectively), the experimental k_{on} is 7 to 9 times faster and the experimental k_{off} is 5 to 8 times faster. Nonetheless, a deviation of less than an order of magnitude is considered very reasonable for estimates of kinetic values. The value of K_D derived from this combination of rate constants is 0.00111, very close to the experimental range of 0.00068 - 0.00082.

4.4 Conclusion

Overall, while dcTMD does not reproduce experimental kinetic data for the T4L-L99A benzene complex exactly, the results are within an acceptable distance, considering the difficulty of making estimates of kinetic data for protein-ligand systems. This is in line with previous calculations using dcTMD on other systems, including Hsp90 and trypsin. The methods selected for pathway separation are effective and allow experimental free energy data to be reproduced accurately. (A comparison with experimental data is necessary, as free energies for protein-ligand dissociation calculated from equilibrium MD simulations are not available.) Pathway separation is clearly an essential part of the protocol; failing to apply it leads to serious misestimation of thermodynamic parameters. This was already known from previous studies, but the confirmation is helpful because the pathways in T4L-L99A are much more clearly defined compared to the previous systems studied, and T4L-L99A pathways have already been the subject of numerous studies. By carefully adjusting the pulling groups used to define the constraint force, TMD simulations were able to consistently reproduce 5 out of 8 of the previously reported pathways, including 4 out of 5 of those pathways reported in at least two publications.

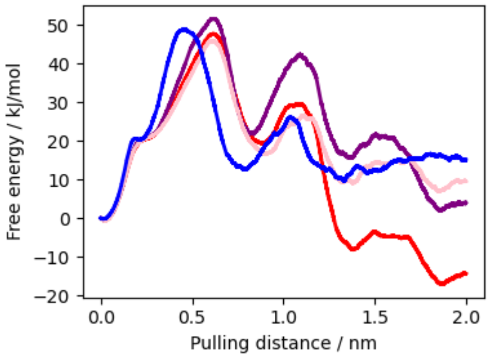


Fig. 4.12: dcTMD free energy profiles for FGH pathways, depicting the original FGH cluster (red), the subclusters FGH (pink) and FGH' (purple), and the trajectories produced using pulling group 3 (blue).

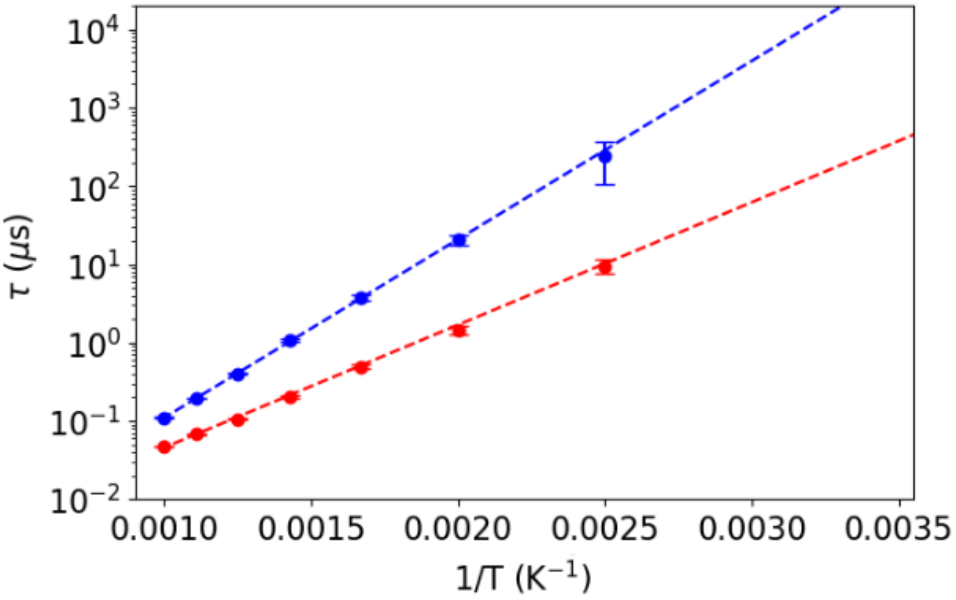


Fig. 4.13: T-boosting illustrated for path CF. The regression lines are extended to 293.15 K.

5 Virtual screening against the SARS-CoV-2 main protease

This chapter summarises the work originally described in the following publication:

- **Simon Bray**, Tim Dudgeon, Rachael Skyner, Rolf Backofen, Björn Grüning, Frank von Delft. Galaxy workflows for fragment-based virtual screening: a case study on the SARS-CoV-2 main protease. *Journal of Cheminformatics*, Volume 14, Article number: 22, 12 April 2022, https://doi.org/10.1186/s13321-022-00588-6

5.1 Introduction

5.1.1 Motivation

From the start of 2020, the SARS-CoV-2 virus spread rapidly across the world, leading to many deaths, lockdowns in many countries and corresponding societal changes [124]. Potential therapeutics against the disease rapidly became the focus of scientific research, in particular vaccines, but to a lesser extent also the development of antiviral drugs. Such a drug would prove easier to store and administer in comparison to a vaccine, and provide a useful additional weapon in the arsenal against the virus [125].

Scientific progress into characterizing the virus was rapid. The virus genome contains 29892 base pairs, which encode 29 different proteins, 4 of which have structural roles, and the viral particle has a diameter of between 60 and 140 nm [124]. At least 4 of the proteins are considered to be potentially druggable: the NTPase/helicase, main protease, papain-like protease, and RNA-dependent RNA polymerase. The main protease (M^{pro}) has been a particular target of investigation, and a crystallographic structure was derived already in January 2020, showing a clearly-defined, solvent-accessible binding site, ideal for crystallographic fragment screening [126]. This then prepared the way for the Diamond Light Source to perform such a screen, generating over 70 fragment hits [127].

In parallel, a open-science project with the aim of discovering an antiviral drug candidate, COVID Moonshot was initiated. As part of the COVID Moonshot project, in collaboration with the experimental scientists from the Diamond Light Source, who performed the fragment screening, a computational project was initiated with the aim of identifying compounds derived from the fragments discovered with strong affinity for the M^{pro} binding site.

The need for openness in data analysis and sharing during the SARS-CoV-2 pandemic has been the subject of comment [128]. A key aim of this project was to meet this need by providing fully accessible data via Galaxy, together with the computational workflows used for analysis and compute resources to repeat and verify it if necessary. Galaxy in general provides a platform which can be easily accessed by researchers and assist them to collaborate against public health emergencies such as the recent pandemic.

5.1.2 Background

The main protease of the SARS-CoV-2 virus, as for all coronaviruses, has an essential enzymatic role in viral transcription. Transcription of the viral genome results in the synthesis of two polyproteins, pp1a and pp1b. To perform their function, these polyproteins must be cleaved at several proteolytic sites; 11 of these cleavages are performed by M^{pro}, while a further 3 are performed by the papain-like protease, releasing a variety of non-structural proteins. Due to this vital role in the viral life cycle, and the fact that there are no close homologs of the protease found in humans, M^{pro} quickly became a focus of interest as a target for a potential antiviral drug.

M^{pro} consists of 305 amino acid residues and has a molecular mass of 33.8 kDa in the monomeric form, though the catalytic form is dimeric (Figure 5.1). The protein is made up of three domains: I and II have a β-barrel structure, linked to III, a cluster of α-helices, by a fifteen-residue loop region. The catalytic site, made up of a catalytic dyad of His41 and Cys145, is located between domains I and II.

The fragment screening performed by the Diamond Light Source revealed 74 fragments, of which 3 bound at the dimerisation interface and 71 in the active site (Figure 5.2). Of the latter, 48 were covalently bound and 23 non-covalently bound. The virtual screening study described here [129] focusses on the non-covalent hits. The identified fragments reveal the presence of multiple subpockets within the active

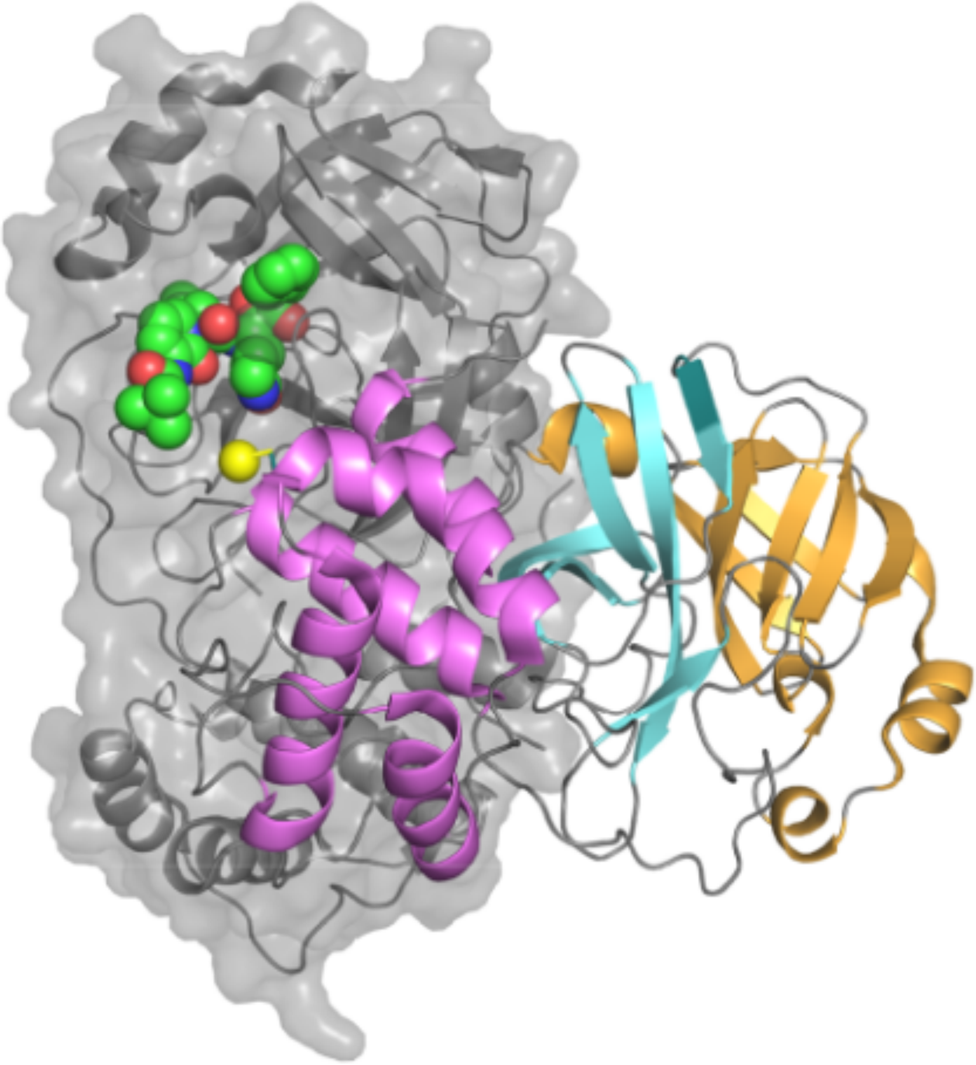


Fig. 5.1: The M^{pro} dimer. For one monomer, domains I (orange), II (cyan) and III (violet) are marked. The location of the catalytic site is marked on the other monomer by a bound inhibitor (green). Information on the image source is provided in the List of Figures.

site, labelled S1, S1', S2 and S3. Several fragments were also able to bridge over two subpockets.

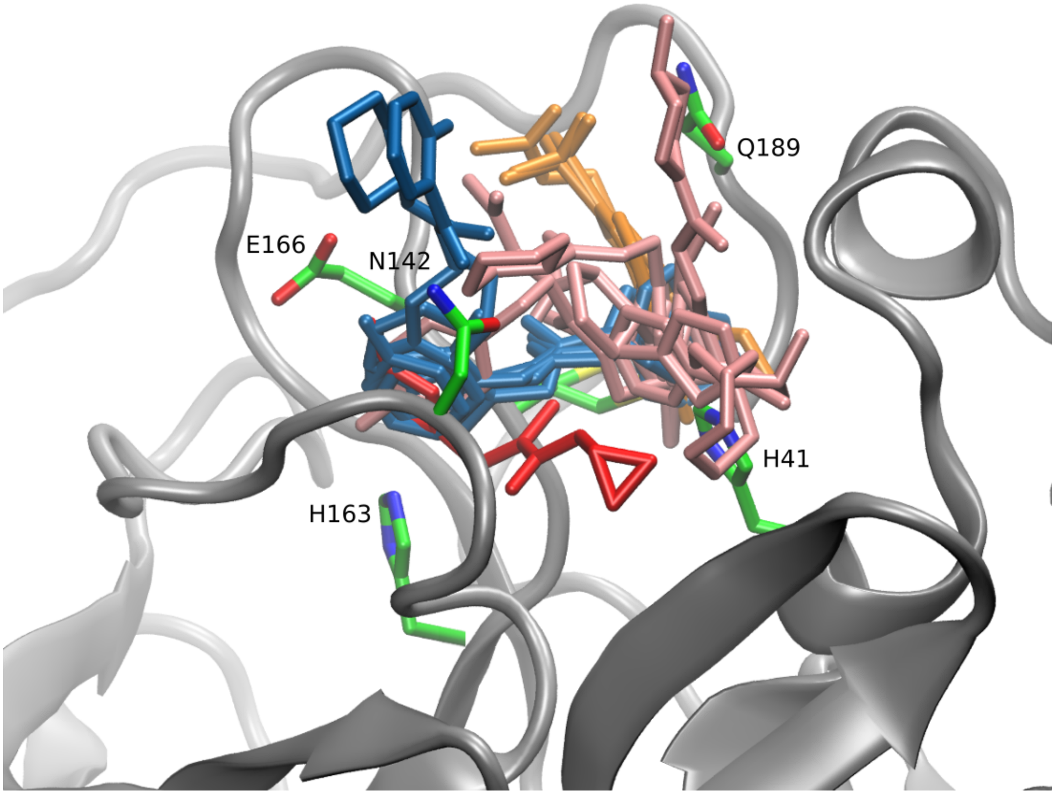


Fig. 5.2: Non-covalent fragments identified by the Diamond Light Source fragment screen. Coloring represents the four subpockets which make up the active site of the protease: S1 (blue), S1' (red, fragment x0397), S2 (pink), and S3 (orange). Information on the image source is provided in the List of Figures.

Of particular interest was a slight conformational change in the protein enforced by only a single fragment, x0397; the position of the side chains of the catalytic dyad mentioned above modifies the size and shape of S1' and open up a link between S1 and S1'. Thus, x0397 is the only fragment which bridges between S1 and S1'.

5.2 Methods

53,000 compounds were generated using the Fragalysis fragment network [130]. To investigate the affinity of these compounds, three different workflows were developed. The application of these workflows is not specific to M^{pro}. The first workflow generated three-dimensional conformers of the compounds, docks them into the M^{pro} catalytic site, scores them using the deep learning-based TransFS method [38], and validates the docked poses against the original fragment positions using the SuCOS measure [131]. The second workflow performs MMGBSA simulations of a selection of the highest-scoring compounds, while the third workflow performs dcTMD simulations on a further refined selection. The latter two workflows share a subworkflow in common for parameterisation of the starting protein-ligand complex, before continuing with solvation, energy minimisation and equilibration. Both workflows make use of Galaxy's collection feature to run ensembles of simulations in parallel, 20 simulations in the case of MMGBSA, 100 in the case of dcTMD.

5.3 Results

5.3.1 MMGBSA

Of the initial set of compounds generated by the Fragalysis fragment network, 209 of the best-scoring were selected for more compute-intensive MMGBSA simulations. Each fragment was represented by multiple derived compounds, allowing a plot of MMGBSA scores to be created per fragment in Figure 5.3. This makes it clear that there is substantial variation between the different fragments. Interestingly, compounds which are derived from the x0397 fragment bind the strongest, according to the MMGBSA calculations; as mentioned before, x0397 is the only fragment which triggers a conformational change in the M^{pro} active site, allowing it to bind in both subpockets S1 and S1'. As a result, it is the only fragment which enables significant hydrogen bonding between a derived compound and the catalytic cysteine residue.

5.3.2 dcTMD

Of the 209 compounds for which MMGBSA simulations were performed, 50 were selected for dcTMD simulations. As already discussed, dcTMD allows an insight into the kinetics of protein-ligand dissociation, and the pathway taken by the ligand out of the active site. Inspection of the trajectories generated demonstrates that there is only a single pathway available for dissociation, removing the need for pathway separation as described for the T4L-L99A system.

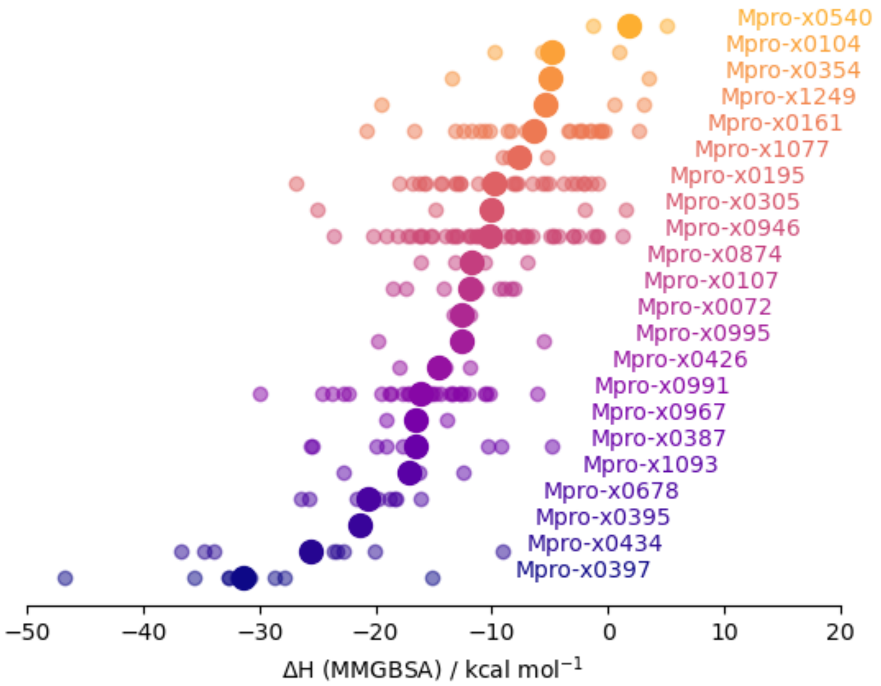
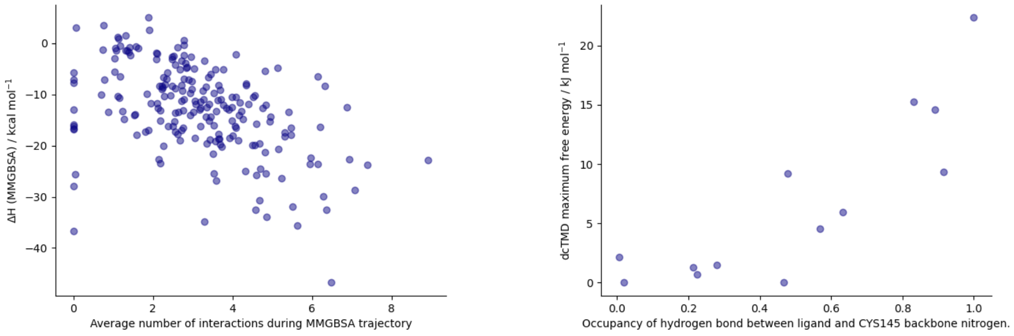


Fig. 5.3: MMGBSA enthalpies for poses derived from each of the 22 fragments studied. Information on the image source is provided in the List of Figures.

In order to validate the results of the MMGBSA and dcTMD simulations, plots were created to compare the interactions observed between the protein binding site and the ligand with the free energy values (Figure 5.4). A plot of MMGSBA free energy values against a raw number of interactions shows a clear correlation, though rather a weak one; however, this is expected, considering that the molecular interactions vary widely in terms of strength and importance. Plotting the occupancy of individual interactions with dcTMD free energy scores reveals the strongest correlation is notably for a hydrogen bond with the Cys145 catalytic residue, indicating that breaking this bond, where it exists, is the major initial barrier for departure of the compound from the active site. As already mentioned, it is compounds deriving from the x0397 fragment which are most likely to be able to access the S1' subpocket and thus to be able to form this hydrogen bond.



(a)

(b)

Fig. 5.4: Left: The average number of interactions observed and the free energy as calculated by MMGBSA are correlated (R2 = -0.46). The weakness of the relationship reflects the high variation in the strength and importance of interactions. Right: Maximum dcTMD free energy scores for compounds which display hydrogen bonding between the peptide backbone and residue Cys145 (R2 = 0.85). Information on the image source is provided in the List of Figures.

5.4 Conclusion

In this paper multiple workflows were developed: for docking compounds into a protein binding site and scoring the resulting poses, for estimation of binding free energies using the MMGBSA method and for estimation of the height of the kinetic barrier to ligand binding using the dcTMD technique. These workflows are flexible and can be applied to any system, and can be executed via either Galaxy's graphical interface or on the command line using the Planemo library. The workflow were demonstrated on the M^{pro} system, and could show that compounds derived from a particular fragment, x0397, bind especially strongly to the M^{pro} binding site, due to the fragment conferring derived compounds with the ability to modify the conformation of the binding site and thus unlock access to a hidden side pocket. Thus, these compounds are able to form some new interactions, in particular a hydrogen bond with the catalytic cysteine of the protein, which improve the compounds' binding affinity.

Galaxy workflows for fragment-based virtual screening: a case study on the SARS-CoV-2 main protease

Galaxy workflows for fragment-based virtual screening: a case study on the SARS-CoV-2 main protease. **Simon Bray**, Tim Dudgeon, Rachael Skyner, Rolf Backofen, Björn Grüning, Frank von Delft. *Journal of Cheminformatics, V*olume 14, Article number: 22, 12 April 2022, https://doi.org/10.1186/s13321-022-00588-6

Personal contribution:

The paper describes three workflows for fragment-based virtual screening, which were used to study the main protease of the SARS-CoV-2 virus. I designed, implemented and executed two of these workflows (MMGBSA and dcTMD) and assisted with the implementation of the docking and scoring workflow. I performed the analysis of the data generated and wrote the manuscript. In recognition of these major contributions to the paper, I am listed as first author for the publication.

Co-authors:

Tim Dudgeon: designed, implemented and executed the docking and scoring workflow and provided comments on the draft manuscript.

Rachael Skyner: contributed to the docking and scoring workflow.

Rolf Backofen: provided overall supervision.

Björn Grüning: provided code review, contributed to the docking and scoring workflow, and assisted with implementing and deploying tools.

Frank von Delft: initiated the project and provided the initial concept for the docking and scoring workflow.

S. A. B

Simon Bray, 13.4.2022

Signatures:

The following co-authors confirm the above stated contribution:

Co-author	Date	Signature
Tim Dudgeon	20 APR 2022	
Rachael Skyner	20/04/2022	
Rolf Backofen	1, 6, 2022	
Björn Grüning	1. 6. 2022	B. Grü
Frank von Delft	26 April 2022	Frank von Delft

Bray *et al. Journal of Cheminformatics* (2022) 14:22
https://doi.org/10.1186/s13321-022-00588-6

Journal of Cheminformatics

RESEARCH ARTICLE Open Access

Galaxy workflows for fragment-based virtual screening: a case study on the SARS-CoV-2 main protease

Simon Bray[1*], Tim Dudgeon[2], Rachael Skyner[3,5], Rolf Backofen[1,4], Björn Grüning[1] and Frank von Delft[3,5,6,7]

Abstract

We present several workflows for protein-ligand docking and free energy calculation for use in the workflow management system Galaxy. The workflows are composed of several widely used open-source tools, including rDock and GROMACS, and can be executed on public infrastructure using either Galaxy's graphical interface or the command line. We demonstrate the utility of the workflows by running a high-throughput virtual screening of around 50000 compounds against the SARS-CoV-2 main protease, a system which has been the subject of intense study in the last year.

Keywords: Fragment screening, Workflows, SARS-CoV-2, Computational chemistry

Introduction

Computational techniques are commonly used to assess the affinity of small druglike molecules to a biological target molecule, typically a protein, in a process known as virtual screening. Virtual screening is a complex, multi-step process which needs to be performed at a high-throughput level of thousands or millions of input molecules. As a result, workflow management systems such as KNIME [1], CWL [2], Nextflow [3] or Galaxy [4] prove useful to organize analyses, allowing automation and parallelization of commonly used steps and avoiding tedious manual repetition.

In previous work, we published a range of cheminformatics [5] and molecular dynamics tools [6] via the Galaxy platform. Galaxy provides a range of useful features, including a convenient web-based graphical interface, storage of essential metadata such as tool parameters, and easy construction and execution of workflows from component tools, either on the command line or via the graphical interface. Reproducibility of analyses is ensured by the installation of software dependencies using BioConda [7], conda-forge [8], or BioContainers [9]. In addition, we pointed out that using Galaxy provides access to vast public compute infrastructures, including GPU resources for molecular dynamics calculation, such as the denbi and STFC clouds which underpin the European Galaxy server, https://usegalaxy.eu, a distinctive feature which distinguishes Galaxy from other workflow management systems.

Here, we present several new workflows for protein-ligand docking, molecular dynamics and free energy calculation. These workflows are constructed out of simpler building blocks (the component Galaxy tools) and can be either used directly or modified as templates for other similar calculations. We demonstrate the utility of these workflows by running them at high scale on a system which has attracted much recent attention, namely the main protease (Mpro) of the SARS-CoV-2 virus.

The main protease of the SARS-CoV-2 virus has been intensively studied since the beginning of the global

*Correspondence: sbray@informatik.uni-freiburg.de
[1] Bioinformatics Group, Department of Computer Science, University of Freiburg, Freiburg im Breisgau, Germany
Full list of author information is available at the end of the article

© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

pandemic, with the first crystal structure released in January 2020 [10]. Subsequent experimental work, involving some of the authors, revealed the crystal structures of Mpro in complex with 96 different fragment structures, including non-covalent hits as well as hits covalently bound to the vital Cys145 residue in the protease binding site [11]. Fragment hits were also found located at the interface between the Mpro dimers. Here we focus our attention on the 22 non-covalent hits bound within the protease active site, excluding two (denoted x1086 and x0887) which bind to other pockets of the protein (the chemical structures of the fragments studied are depicted in Additional file 1: Fig. S1). We use these 22 hits as the basis for generating a list of candidate compounds using the Fragalysis [12] fragment network, a reimplementation of the Fragment Network concept originally developed by Astex Pharmaceuticals [13]. These compounds are then docked using rDock against each of the crystallographic structures from the fragment screen. The resulting docked structures are validated against the original fragment structures using the SuCOS [14] measure and scored using the TransFS [15] deep learning-based method. Based on these scores, the compounds can be ranked and the most promising of them (around 200) used for further free energy calculations. These are performed using the MMGBSA technique, using an ensemble of a total of 5 ns of simulation time per compound. Subsequently we take the 50 top-scoring compounds from the MMGBSA simulations and perform more computationally expensive dcTMD (dissipation-corrected targeted molecular dynamics) [16, 17] calculations, requiring a total of 50 ns of simulation time per compound.

The three workflows themselves (docking and scoring, MMGBSA calculations, and dcTMD calculations) can be flexibly applied to any system, not only Mpro. To facilitate usage by other users in the future, they have been deposited in the Intergalactic Workflow Commission (IWC) [18], a curated repository for Galaxy workflows. To ensure reliability and reproducibility, the workflows are packaged together with tests which are run via continuous integration (CI). If tests are successful and the submission is approved by an IWC review, the submitted workflows are deployed to Dockstore [19] and WorkflowHub [20], two recently developed platforms for sharing scientific workflows. Links for access are provided in Additional file 1: Table S2.

Methods

Three main workflows have been developed as part of this work: an initial protein-ligand docking and scoring workflow, in which hypothetical protein-ligand structures are generated and ranked; a relatively low-cost free energy calculation workflow, based on the MMGBSA technique, which is run on the most promising of the docked complexes; and a more costly free energy calculation technique, based on the recently published dcTMD method. Subsequent analysis of molecular interactions and plotting of data is performed outside Galaxy. Images of the active site are generated using VMD [21].

Protein-ligand docking and scoring

The inputs for the docking and scoring workflow consist of a protein structure for docking and a list of candidate compounds. The initial list of candidates is generated with the Fragalysis fragment network API, using the 22 selected fragment hits as inputs to be extended, generating molecules that are close neighbours of the starting molecules in the fragment network.

For those initial candidates, various charge forms between pH 4.4 and 10.4 are enumerated using DimorphiteDL [22]. A single three-dimensional conformer for each of these forms is then produced using OpenBabel [23] as the starting structure for docking. The main task of the workflow, after enumerating charge forms and conformer generation, is to dock each of the enumerated conformers into the binding sites of the fragment crystal structures to generate numerous docking poses, using the open source rDock software [24]. The workflow makes use of the Galaxy's collection feature to split the initial list of compounds and process the resulting chunks in parallel, essential given the large amount of poses generated. Pocket definition for the docking was achieved by the so-called 'Frankenstein ligand' technique of combining atomic coordinates from all fragments into a single hybrid molecule for use as a reference ligand.

Docking produced a large number of poses, which were then evaluated using two measures. Firstly, the SuCOS measure is used to assess the overlap between the putative binding position of the compound and each of the experimental fragment crystal structures. The aim is to validate the docked poses and to ensure they share a similar conformation and position to at least one of the experimental crystallographic structures. Secondly, the TransFS tool, based on a deep learning model trained on a variety of molecular interactions, is used to score each of the poses.

A schematic of the workflow is provided in Fig. 1. For our concrete use case, we provide an initial list of 53,787 compounds, which are generated by the Fragalysis fragment network. After charge enumeration and conformer generation, this value is expanded to 219,247, or around 4 conformers per compound. For each of these, 25 docking poses are generated, giving a total of over 5 million poses.

It should be noted that this workflow is run separately for each of the fragment crystal structures, i.e. 22 times, corresponding to a total of over 120 million docking

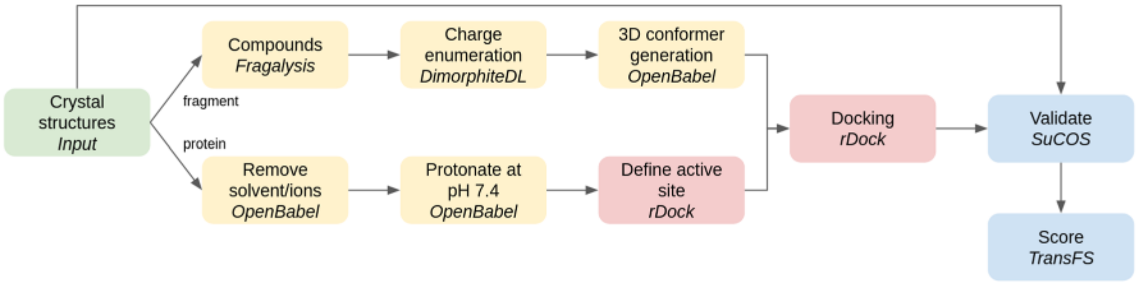

Crystal structures *Input*
fragment
protein
Compounds *Fragalysis*
Charge enumeration *DimorphiteDL*
3D conformer generation *OpenBabel*
Remove solvent/ions *OpenBabel*
Protonate at pH 7.4 *OpenBabel*
Define active site *rDock*
Docking *rDock*
Validate *SuCOS*
Score *TransFS*

Fig. 1 Schematic of the docking and scoring workflow

poses. Poses are thus validated against a single fragment during the SuCOS scoring stage. As a result, for each fragment, we obtain a separate list of poses which are ranked only on the basis of their overlap with that single fragment. All poses are also scored using the TransFS tool.

A customizable subworkflow is responsible for filtering the poses based on the assigned scores. Filtering proceeds by selecting the top 5000 compounds for each fragment (around 0.1%) by SuCOS score. As a rule of thumb, a SuCOS score of over 0.5 is acceptable; thus, all poses which differ substantially in conformation and position from the experimental structures are discarded. This subset of poses with high SuCOS scores is then filtered further in one of three ways: (1) selecting all with SuCOS > 0.6 and TransFS > 0.9, (2) selecting all with SuCOS > 0.7 and TransFS > 0.8, (3) for all fragments where these two filtering steps resulted in less than 3 outputs, the top 3 poses based on TransFS scores are selected. By applying this complex filtering, we obtain a range of poses which score highly for both TransFS and SuCOS measures, as well as ensuring a wide chemical diversity of poses with all of the component fragments represented. The filtering is implemented using the sdsort and sdfilter commands which are provided alongside rDock.

A tutorial describing the docking and scoring workflow is available via the Galaxy Training Network [25] at https://bit.ly/31vAZpI.

MMGBSA free energy workflow

The list of compounds obtained after application of the docking and scoring workflow comprises around 210 molecules. To obtain a low-cost assessment of the free energy of binding for each of the poses, we perform MMGBSA calculations, using GROMACS [26] to perform the molecular dynamics simulations and AmberTools [27] for the calculations themselves.

Firstly, a subworkflow for system parameterization is used to prepare the selected ligands for MD simulation. The docked poses are converted from SDF to MOL2 format and parameterized using the GAFF forcefield [28], using tools based on AmberTools and acpype [29]. Meanwhile, the protein structure is parameterized with the AMBER99SB forcefield, using a tool based on GROMACS's pdb2gmx. Using the tagging system provided by Galaxy, each of the poses is annotated with its respective SuCOS and TransFS value, together with the identity of its parent fragment. These metadata are inherited by datasets produced in subsequent analysis, allowing quick overview of all data for any particular compound.

Solvent (water represented with the TIP3P model) and sodium or chloride counterions are added as required to neutralize the system, before performing energy minimization. The molecular dynamics simulations themselves are performed using GROMACS with a timestep of 1 fs at a temperature of 300 K. 100 ps of equilibration simulations (50 ps under the NVT ensemble followed by 50 ps under the NVT ensemble) are performed with constraints on the protein atoms. The production simulations (length 200 ps) are then performed under the NVT ensemble. For each compound, an ensemble of 20 simulations are performed, taking advantage of Galaxy's collection functionality to create a list of datasets and apply a tool over the entire list as a single workflow step. The size of the ensemble is configurable as a workflow parameter. The production simulations are then used as a basis for the MMGBSA calculations and a mean across the ensemble is calculated. An schematic of the entire workflow is provided in Fig. 2. It should be noted that the entropic component to the free energy is not included in the calculations, so the values generated represent only the enthalpy of binding.

One of the major reasons to use the Galaxy platform for executing these workflows is that all data, as well as the parameters used for all simulations, are preserved in public Galaxy histories, ensuring full reproducibility.

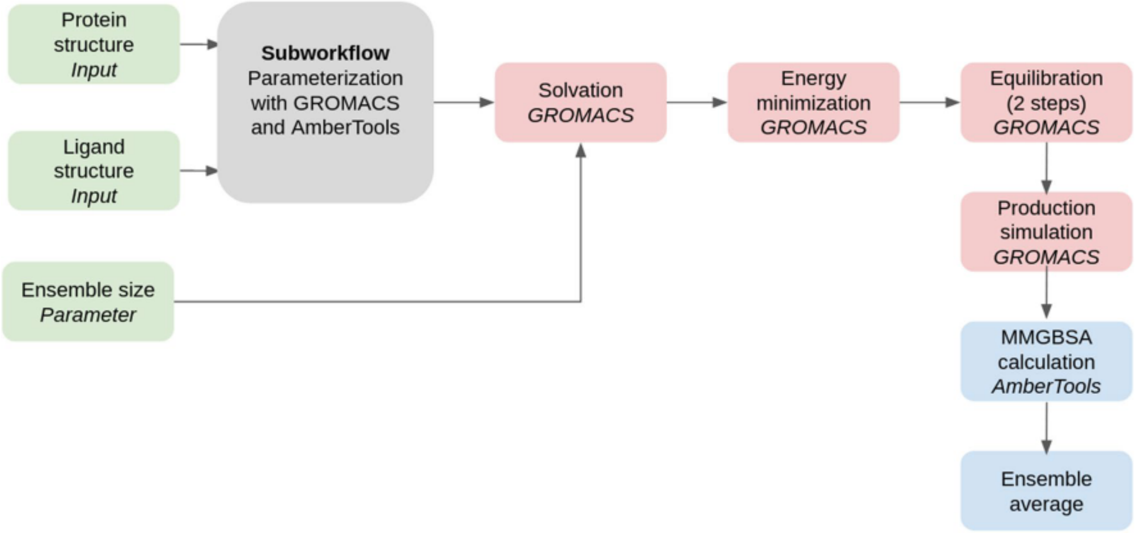

Protein structure *Input*
Ligand structure *Input*
Subworkflow Parameterization with GROMACS and AmberTools
Solvation *GROMACS*
Energy minimization *GROMACS*
Equilibration (2 steps) *GROMACS*
Production simulation *GROMACS*
MMGBSA calculation *AmberTools*
Ensemble average
Ensemble size *Parameter*

Fig. 2 Schematic of the MMGBSA workflow. A modular subworkflow for system parameterization is shared with the dcTMD workflow; see Fig. 4 for details

Links to all histories are provided in the Additional file 1.

dcTMD free energy workflow

As a further demonstration of the capabilities of our tools, and the flexibility of the Galaxy platform which allows them to be combined in numerous different ways, we have designed a third workflow which makes use of the recently published dcTMD free energy technique. The main aim of dcTMD is to provide insight into the kinetics of protein-ligand dissociation; a drug candidate which has a low rate of dissociation from the target protein and thus a high residence time [30] in the binding site will be preferred to a candidate which dissociates quickly. The theoretical background, with comparisons against various common benchmark systems, was provided in two previous publications [16, 17]; the physical basis of the method is described in detail in those two works. The main advantage of the dcTMD method is its provision of free energy and friction profiles for protein-ligand dissociation, with even sampling of the entire reaction coordinate, including areas of high free energy which are infrequently sampled at equilibrium and inherently difficult to study.

The process entails simulation of an ensemble of constraint targeted molecular dynamics (TMD) simulations, in which a constraint pulling force is applied between two atom groups (typically, the ligand and part of the protein) to separate the two groups at constant velocity. The pull groups used for Mpro simulations are depicted in Fig. 3. By applying a weighted average

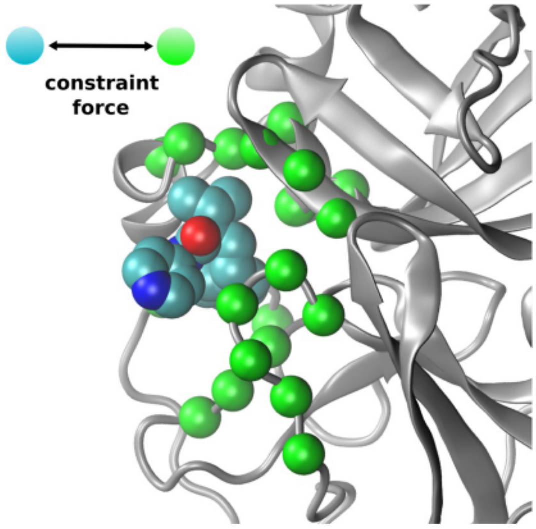


Fig. 3 Pull groups for the TMD simulations (image depicts the x0397 structure). Group 1 (cyan) consists of the ligand non-hydrogen atoms. Group 2 (green) consists of a selection of alpha-carbons in the Mpro active site. During the course of the TMD simulation, the two groups are pulled apart by means of a constant constraint force

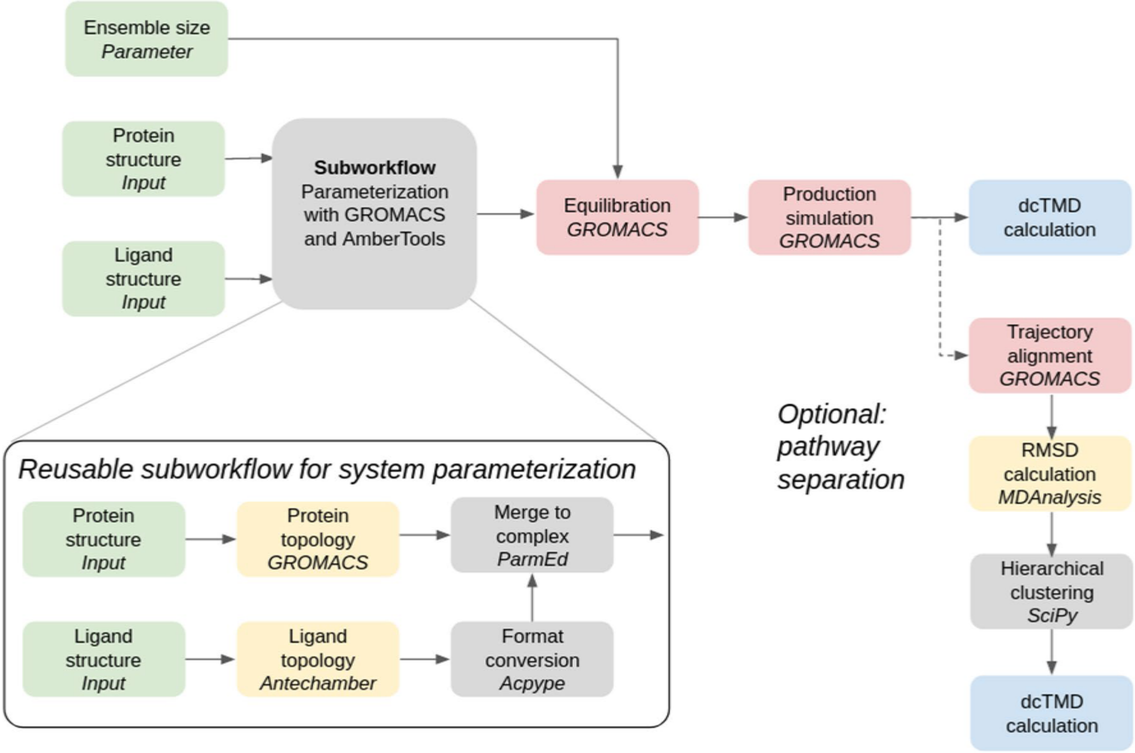

Ensemble size
Parameter
Protein structure *Input*
Ligand structure *Input*
Subworkflow
Parameterization with GROMACS and AmberTools
Equilibration *GROMACS*
Production simulation *GROMACS*
dcTMD calculation
Trajectory alignment *GROMACS*
RMSD calculation *MDAnalysis*
Hierarchical clustering *SciPy*
dcTMD calculation
Optional: pathway separation
Reusable subworkflow for system parameterization
Protein structure *Input*
Protein topology *GROMACS*
Merge to complex *ParmEd*
Ligand structure *Input*
Ligand topology *Antechamber*
Format conversion *Acpype*

Fig. 4 Schematic of the dcTMD workflow

across the ensemble, based on an approximation of the Jarzynski equality [31], free energy and friction profiles for the system at equilibrium can be calculated, despite the fact the ensemble is made up of non-equilibrium simulations.

In order to streamline the process of performing dcTMD calculations, we have developed a complete Galaxy workflow for both simulation and the subsequent calculations. This workflow functions similarly to the MMGBSA workflow, in that it represents the MD ensembles using Galaxy collections, the size of which can be parameterized using a workflow parameter. For dcTMD simulations, an ensemble size of around 100 is recommended [32]; we therefore set ensemble size to 100 for each ligand. MD simulations are performed using GROMACS using a timestep of 1 fs at a temperature of 300 K. 80 ps equilibration is performed under the NPT ensemble with restraints on the protein atoms for each simulation, followed by a 500 ps production TMD simulation under the NPT ensemble without restraints, in which the two pulling groups are separated with a velocity of 1 m/s - in other words, the ligand ends the simulation at 500 pm from its initial position bound in the active site. Pulling simulations are achieved using the PULL code incorporated into GROMACS. As the Mpro binding site is rather shallow, this simulation length is sufficient to sample the entire dissociation pathway. As for the MMGBSA workflow, all data, as well as the parameters used for all simulations, are published in Galaxy histories linked in the Additional file 1.

An essential part of the dcTMD process is pathway separation. One of the core assumptions of the dcTMD protocol is Gaussianity of the work profile of the ensemble, which is acceptable if the ligand takes a uniform path between the bound and unbound state, but breaks down if the ligand is able to take multiple paths out of the binding site. Therefore, it is essential to carry out an analysis to determine whether distinct paths are present in the ensemble. Galaxy tools are also provided to align the TMD trajectories according to the protein atoms and perform hierarchical clustering based on the RMSD between ligand positions. The user then has the option to inspect the clusters manually and to apply the dcTMD calculation again to a subcluster of the ensemble.

A schematic of both the main dcTMD workflow and the optional pathway separation is provided in Fig. 4. Our main aim in calculating the dcTMD free energy profiles is to obtain a value for the maximum free energy reached, which heavily influences the kinetics of dissociation. The position of this barrier on the reaction coordinate is also of interest; by inspecting the free energy and friction profiles generated in combination with the TMD trajectories, links can be made between features of the profiles and events along the unbinding coordinate.

Workflow execution

The workflows detailed here required a high number of executions, particularly in the case of the MMGBSA workflow, which was invoked over 200 times. Galaxy provides a graphical web-based interface for tool and workflow execution, as well as to inspect outputs, but this is of limited use for a project like this one, which requires workflows to be executed several hundred times.

Fortunately, command-line tools are available to automate this process, by providing programmatic access to Galaxy's API. Workflows are invoked using the command line tool Planemo [33], modifying the input files for each run. This can easily be scaled up using a simple shell script containing a for loop. The Python library BioBlend [34] was also used extensively to move and organize datasets, run individual tools, and restart paused workflows.

Table 1 summarizes execution statistics for each of the workflows. A summary of the number of compounds studied in each step is provided by Table 2.

Results and discussion

Docking

We have assembled three different workflows which can be applied sequentially for virtual screening of a protein. In particular, we have demonstrated the use of these workflows by running them on the SARS-CoV-2 main protease. A key point is that these workflows consist of simple building blocks which can be simply disassembled and recombined to allow different types of analyses and calculations than those covered here. Of the 50000 compounds in our original library, we have identified around 210 docking poses which are scored highly by the TransFS measure, as well as matching the conformations and positions of the component fragments well. For these compounds, we have performed MMGBSA calculations based on ensembles of MD simulations. Additionally, we demonstrate a more computationally intensive dcTMD workflow on a subset of around 50 highly scoring compounds. A summary table is provided in Table 3.

Figure 5a and b shows distributions of TransFS and SuCOS scores per fragment. TransFS scores cluster around a modal value of 0, with a small minority of compounds scoring highly; the 99th percentile lies at 0.61, but the distributions of scores are similar for all the fragments (Additional file 1: Table S1). The single exception is x1093, for which all compounds score effectively 0; the reason for this is difficult to identify, due to the black box nature of the TransFS method, so the TransFS filtering is simply skipped for this fragment. Unlike TransFS, the SuCOS scores are very unevenly distributed, depending on the compound's parent fragment. It can be observed, for example, that in general smaller fragments such as x0995 score highly, which is unsurprising, as a smaller fragment can fulfil the conditions for overlap more easily. When filtering compounds based on SuCOS score, this should be taken into account, else an unwanted bias towards these smaller fragments is introduced.

Figure 5c demonstrates that the SuCOS and TransFS scores are orthogonal, allowing effective filtering of the compounds on two different measures. While the top right corner of Fig. 5c is relatively sparsely occupied, there are enough compounds present there to select a reasonable number of candidates which score highly on both measures for further study. However, because of a difference between SuCOS score distribution between the different fragments, applying a crude cutoff would ensure certain fragments were heavily overrepresented, while others would remain completely unrepresented. We therefore have developed the more complex filtering workflow described in the Methods section, to ensure all fragments receive some representation in the filtered dataset.

Table 1 Summary of workflow resource usage

Workflow	CPU time / h	GPU time / h	Data storage / GB	Number of executions	Datasets created
Docking and scoring	3000	1	80	22	6000
MMGBSA	30	2	3	209	893
dcTMD	112	14	6	50	1726

Values for resource usage are approximate and can vary substantially between workflow invocations

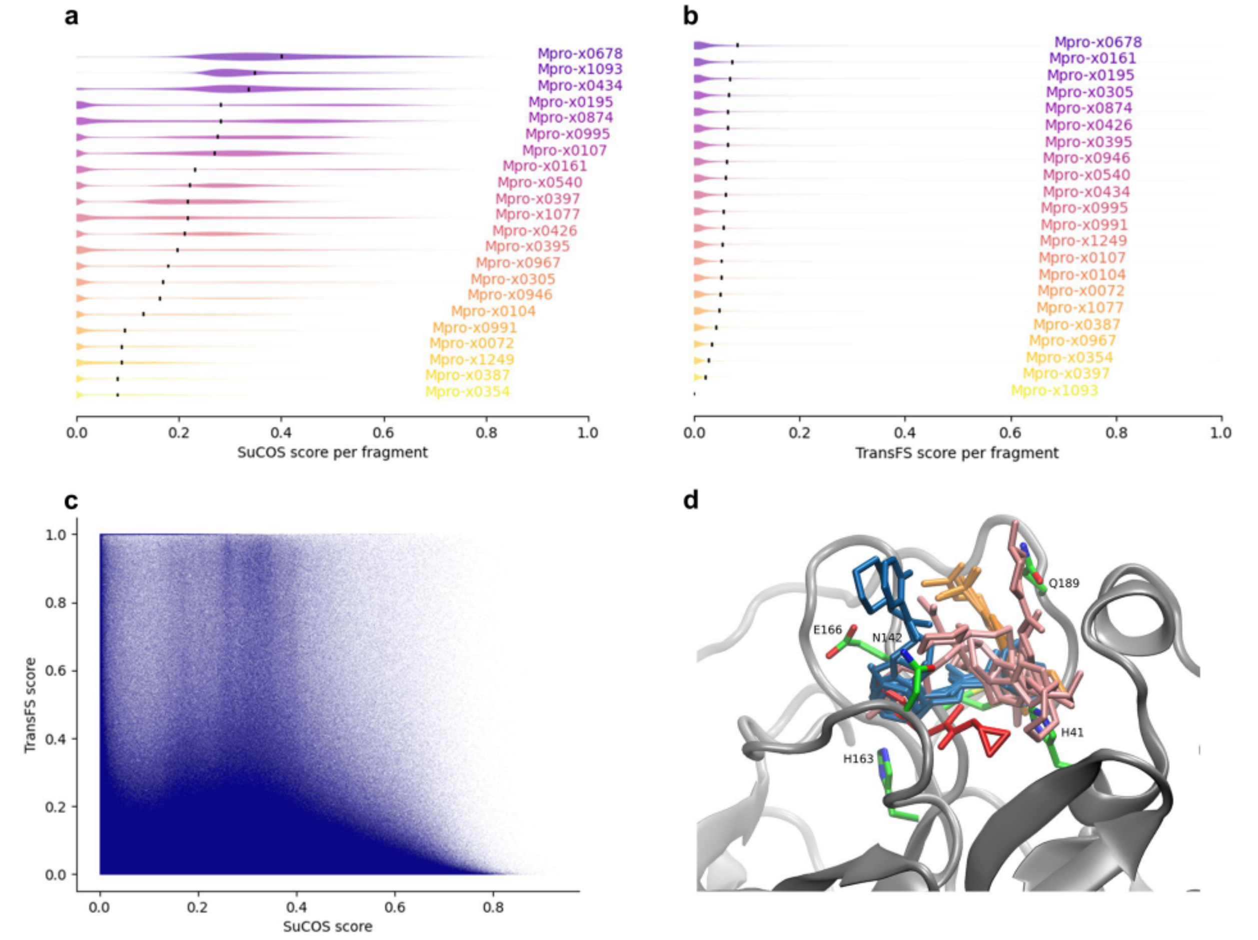


Fig. 5 **a** and **b** Distributions of SuCOS and TransFS scores per fragment; the mean values are marked in black. **c** Scatter plot of SuCOS and TransFS scores for all poses. 209 of these are filtered for further screening **d** All fragments superimposed on the protein structure and colored by the main subpocket to which the fragment binds (S1' red, S1 blue, S2 pink, S3 orange)

Table 2 Number of compounds or poses filtered and studied at each stage

Stage	Fragments	Fragalysis	Docking	MMGBSA	dcTMD
Number of compounds	22	53k	120M	209	49

MMGBSA

It is interesting to note that the strongest binders, according to the MMGBSA calculations, were those compounds derived from the x0397 fragment (Fig. 6). x0397 is notable as the only fragment which induces a conformational change in the protein; on binding, it displaces the sidechains of the Cys145 and His41 catalytic residues and allows access to an additional subpocket (S1') to which other fragments cannot bind. Considering the

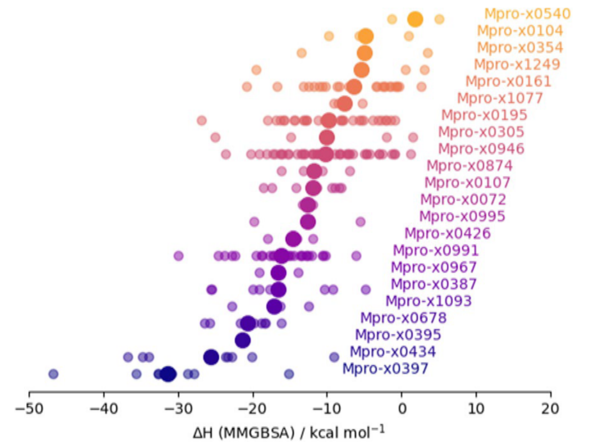


Fig. 6 Plot of MMGBSA enthalpies for poses derived from each of the 22 fragments (mean marked by the large circles)

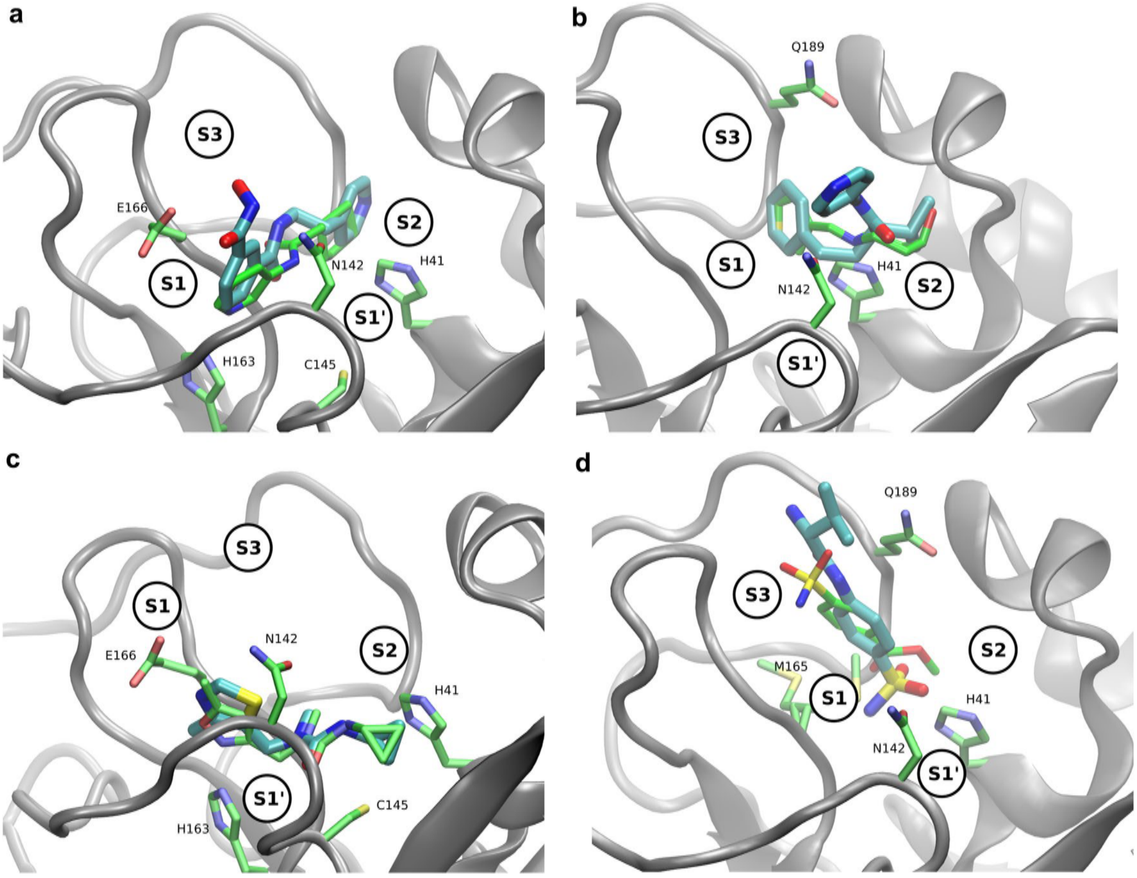


Fig. 7 Ligands (cyan) binding in pockets, overlaid on the parent fragments (green): S1′ **a**) (x0397; SuCOS 0.65)), S1 **b** (x0387; SuCOS 0.56), S2 **c** (x0678; SuCOS 0.53) and S3 **d** (x0161; SuCOS 0.60)

other subpockets, compounds derived from fragments located in both subpockets S1 (e.g. x0434, x0678) and S2 (e.g. x0395, x0387) score highly. On the other hand compounds derived from the three sulfonamide derivatives x0161, x0195 and x0946, which bind in S3, score poorly. Figure 7 depicts four fragments bound to each of the subpockets, together with a derived docking pose superimposed.

Inspection of hydrogen bonds formed during the MD simulations reveals a range of different interactions formed and a wide variation over the set of fragments, as expected. For example, fragment x0678 contains a pyridine group which forms a hydrogen bond with the side chain of His163, buried within subpocket S1. This bond is inherited by several of the compounds derived from x0678. Alternatively, for others of the compounds, the pyridine ring of x0678 is replaced with a hydroxyl or oxime group, which can then form a hydrogen bond with the side chain of Glu166, although the bond does not exist for the fragment itself. Glu166 is also able to form hydrogen bonds with some compounds from its main chain amide group, reflecting its key position at the entrance to subpocket S1.

As it provides access to S1′, x0397 is also the only fragment which enables significant hydrogen bonding with the catalytic cysteine residue.

dcTMD

Various information can be extracted from the TMD ensemble. Firstly, free energy profiles can be calculated, depicting the free energy of the system relative to the bound state at different points on the pulling coordinate. Friction profiles can also be calculated, depicting the

Table 3 Compounds with a maximum dcTMD free energy of over 10 kJ/mol, together with all other calculated scores, and interactions inherited from the component fragments

Index	dcTMD maximum free energy / kJ/mol	Parent (and other component) fragments	Distance of dcTMD maximum from binding site / nm	MMGBSA / kcal/mol	SuCOS	TransFS	Interactions, with occupancy and derived fragment
1	22.41	x0387 (x0434)	0.45	−17.74	0.56	0.94	Cys44BO HB 91.5% (x0387) Met165 HI 88.5% (x0434) Gln189 HI 94.5% (x0434) His41 pi stacking 6.5% (x0387)
2	18.4	x0387 (x0434)	0.34	−25.51	0.54	0.95	Met165 HI 94% (x0434) His41 pi stacking 44% (x0387) Gln189 HI 88% (x0434)
3	16.45	x0991 (x0946)	0.24	−29.93	0.64	0.96	
4	15.25	x0397	0.24	−31.97	0.65	0	Gly143BN HB 100% (x0397) Cys145BN HB 83.5% (x0397) Thr25 HI 10.5% (x0397)
5	14.57	x0397	0.18	−30.74	0.61	0	Gly143BN HB 85.5% (x0397) Cys145BN HB 89.5% (x0397) Thr25 HI 62.5% (x0397)
6	13.89	x0434	0.38	−25.42	0.49	0.65	Glu166BN HB 84.5% (x0434) Met165 HI 64% (x0434) Gln189 HI 19% (x0434)
7	13.61	x0678	0.73	−26.4	0.53	0.94	His163SC HB 14% (x0678) Met165 HI 50% (x0678) Glu166 HI 90% (x0678)
8	11.96	x0305	0.52	−25.07	0.54	0.94	Met165 HI 87.5% (x0305) Gln189SC HB 13% (x0305)
9	10.95	x0434	0.43	−22.71	0.52	0.68	Gln189 HI 50.5% (x0434) Met165 HI 10.5% (x0434) Glu166BN HB 3.5% (x0434)
10	10.57	x0434 (x0387)	0.29	−34.78	0.52	0.77	Glu166BN HB 77.5% (x0434) Met165 HI 61.5% (x0434) His163SC HB: 44% (x0434)

The chemical structures of the compounds are depicted in Additional file 1: Fig. S2. *BO* backbone oxygen, *BN* backbone nitrogen, *SC* side chain, *HB* hydrogen bond, *HI* hydrophobic interaction

friction present in the system over a particular point in the reaction coordinate. A classic protein-ligand dissociation free energy profile depicts a peak between the bound and unbound state, with the unbound state generally higher in free energy than the bound state (for example, Fig. 8). The height of the peak is of particular interest, as it represents the kinetic barrier to dissociation (Table 3). Secondarily, the position of the peak, or any other features in the free energy or friction profiles, can provide insight into the dissociation pathway, when considered together with manual inspection of the TMD trajectories.

For all of the ligands examined, it appears there is only a single pathway available for ligand dissociation, thus obviating the need to perform the pathway separation step. This is not surprising, given that the binding pocket of Mpro is fairly close to the protein surface.

Inspecting the TMD trajectories, various other interactions become apparent which were not observed in the equilibrium simulations already performed. For the ligands located in the S1 and/or S1' pockets, such as those derived from fragments x0397 or x0991, an interaction with Asn142 at around 0.25 nm from the binding

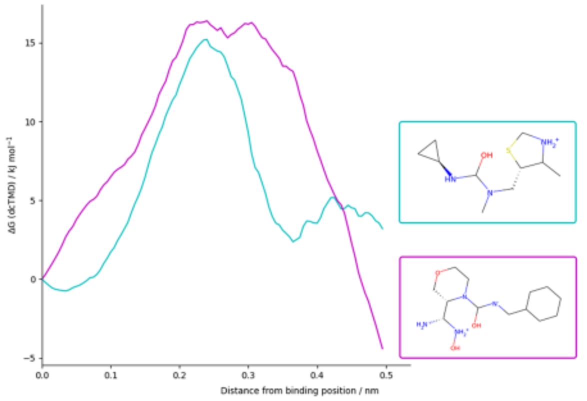


Fig. 8 Free energy curves derived from dcTMD calculations for two of the screened compounds

site can be observed. Asn142 protrudes over the active site, partially covering the entrance to S1 and S1', where many of the most successful candidate compounds are bound. Therefore, exiting from the binding site entails overcoming a steric clash with the side chain, as well as breaking any transient electrostatic interaction formed with the asparagine side chain. In support of this theory, in the TMD trajectories inspected, the dcTMD free energy peak observed at around 0.3 nm corresponds to the point at which the ligand pushes the side chain aside, having already broken the key molecular interactions, so that no major obstacles now remain to leaving the active site. For fragments exiting from the S2 subpocket, an interaction on the other side of the binding pocket is frequently observed (Fig. 9), with the short helical substructure between amino acids 44 and 50 evident, in particular Ser46, the side chain of which is optimally oriented to face the ligand as it exits the S2 subpocket.

Interactions

In order to validate the results from the dcTMD and MMGBSA workflows, the interactions between the protein binding site and the docked molecule were systematically examined. This analysis was conducted outside Galaxy using a Python script [35] based on the Open Drug Discovery Toolkit (ODDT) [36]. All hydrogen bonds and hydrophobic interactions between the crystallographic fragments and the binding site were extracted, together with the less frequently occurring salt bridges, π -stacking and π-cation interactions, and halogen bonds. Subsequently, the same script was used to analyse the MMGBSA trajectories produced for each pose, filtering to include only those interactions present in the fragments. By applying the script to one of the equilibrium MD trajectories used for MMGBSA calculation, rather than a static structure, an estimate can be obtained of the occupancy of an interaction over time, rather than simply its presence or absence.

38 interactions were found between the initial 22 fragments and the protein binding site, an average of 1.73 interactions per fragment. By contrast, averaging over the MD trajectories, each compound on average shows 3.13 interactions with the binding site, demonstrating that the method effectively combines multiple fragments to increase the number of protein–ligand interactions.

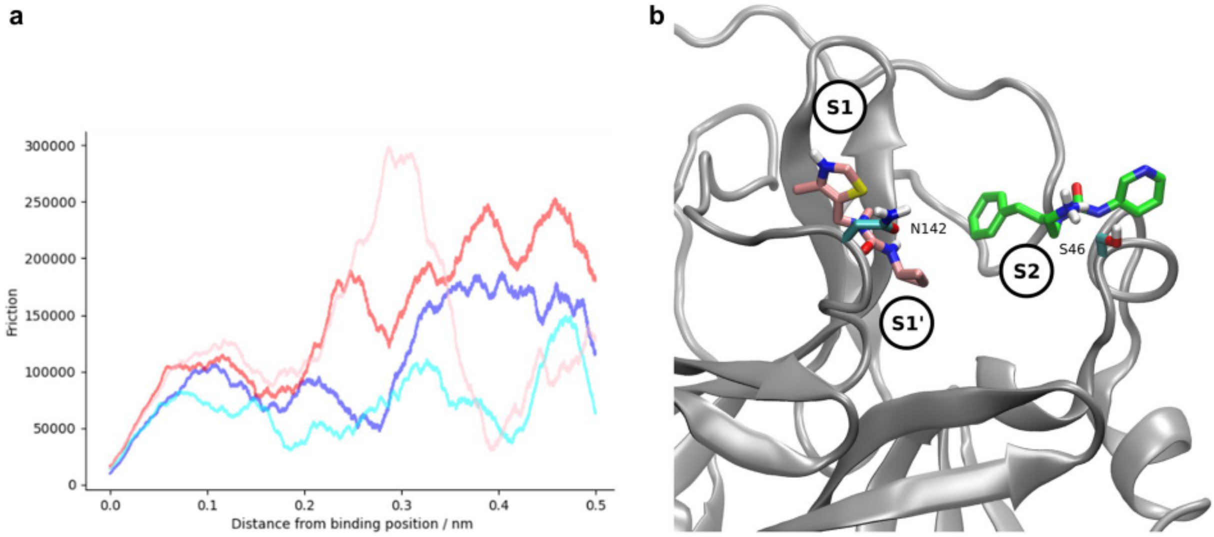


Fig. 9 **a** Friction profiles for four selected ligands; the profiles for the ligands binding in subpocket S1/S1' (red/pink) show a rise starting at 0.2 nm, whereas for those binding in subpocket 2 (blue/cyan), this is absent, with an increase being observable instead at 0.3 nm. **b** Ligands exiting the subpocket S1/S1' at 0.25 nm from the initial binding position (pink), with Asn142 highlighted, and subpocket S2 at 0.33 nm from the initial binding position (green), with Ser46 highlighted

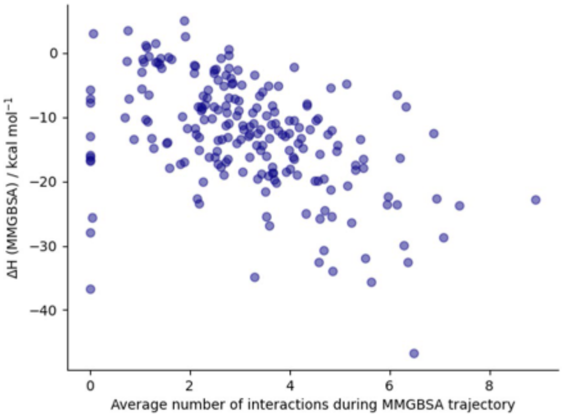


Fig. 10 The average number of interactions observed and the free energy as calculated by MMGBSA are correlated ($R^2 = -0.46$). The weakness of the relationship reflects the high variation in the strength and importance of interactions

MMGBSA free energies correlate with the number of interactions (Fig. 10), so that considering only the subset of compounds with MMGBSA of less than -20 kcal/mol gives an average of 4.57 interactions.

In addition, a search was also performed for new interactions which do not originate from the crystallographic fragments. This yielded very few results. The most common is a salt bridge between the ligand and Glu166, which is present in 11 molecules with an occupancy > 0.5. Others are even rarer: the second most common interaction not present in the original fragments is a hydrogen bond with the backbone nitrogen of Pro168, for which the maximum occupancy is 0.45; a total of only 7 have an occupancy > 0.1. Considering the chemical diversity of the fragments and their distribution through the binding site, it is not surprising that there is little scope for new interactions to appear, but it helps to confirm that the compounds found successfully replicate the chemistry of the original fragments.

According to Table 3, the majority of the highest-scoring compounds have several high-occupancy interactions inherited from the fragments of which they are composed. In particular, a hydrophobic interaction between Met165 and the ligand is present for almost all the compounds - this interaction is also present for 10 of the 22 original fragments, due to its crucial position at the intersection of the S1 and S2 subpockets. For compounds derived from the x0434 fragment, a hydrophobic interaction with Gln189 and a hydrogen bond with Glu166 also frequently recurs. For compound 3, on the other hand, no interactions can be detected; this is due to the fact that no interactions exist, at least according to the script used, between the parent fragment x0991 and the protein. For the compounds derived from the x0397 fragment, which allows a change in protein conformation and which provided the highest MMGBSA scores, other interactions predominate: hydrogen bonds with Gly143 and Cys145, and to a lesser extent a hydrophobic interaction with Thr25. Both these hydrogen bonds between the ligand and the backbone nitrogen atoms of Gly143 and Cys145 show a particularly strong relationship with the dcTMD free energy score (Fig. 11), and appear only with the x0397 fragment.

The dcTMD scores represent the peak of the free energy profile of dissociation—thus, a high correlation between these interactions and the dcTMD score implies they play an important role in raising the barrier to debinding, where they are present.

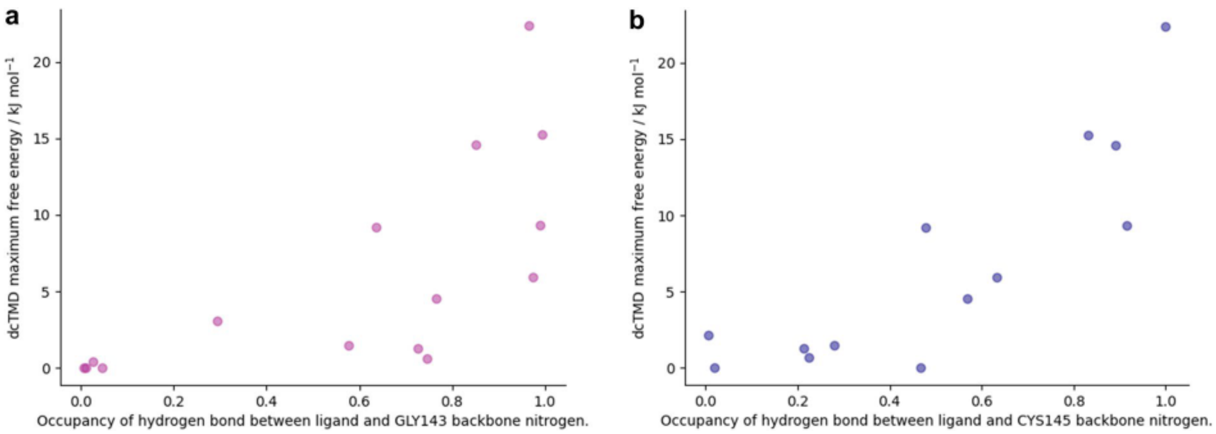


Fig. 11 Maximum dcTMD free energy scores for compounds which display hydrogen bonding with the peptide backbone at residues Gly143 ($R^2 = 0.69$) and Cys145 ($R^2 = 0.85$)

Conclusion

We have presented several new workflows for virtual screening, including protein-ligand docking and scoring, an established free energy technique (MMGBSA) and a more recently developed free energy technique (dcTMD), and demonstrated their use with a study on the main protease of the SARS-CoV-2 virus. These workflows allow us to study a very high number of initial candidate compounds, before narrowing to a smaller selection which we study using more computationally intensive MD techniques. The use of these workflows demonstrates the flexibility of the GROMACS-based MD tools in Galaxy, which can be combined together to create various different types of simulation, including non-equilibrium TMD simulations.

A key motivation for using the Galaxy platform for this kind of study is to enable reproducible, transparent research. Therefore, all datasets are available in the form of published Galaxy histories at https://usegalaxy.eu. Links to the histories are provided in the Additional file 1.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-022-00588-6.

Additional file 1: Fig. S1. Fragments used as a basis for the virtual screening. **Table S1.** 99th percentile of TransFS and SuCOS scores per fragment. **Fig. S2.** Top scoring compounds by dcTMD. **Table S2.** Links for accessing the workflows.

Acknowledgements

We thank Christopher Barnett, Bérénice Batut, Marius van den Beek, John Chilton, Alan Christie, Gianmauro Cuccuru, Kyllen Dilsook, Rubén Sánchez, and Steffen Wolf for instructive discussions, code review, assistance with system administration or comments on the paper.

Authors' contributions

S.B. designed, implemented and executed the MMGBSA and dcTMD workflows, implemented the docking and scoring workflow, performed the analysis, and wrote the manuscript. T.D. implemented and executed the docking and scoring workflow. R.S. contributed to the docking and scoring workflow. R.B. provided overall supervision. B.G. provided code review, contributed to the docking and scoring workflow, and assisted with implementing and deploying tools. F.v.D. initiated the project and provided the initial concept for the docking and scoring workflow. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—Project-ID 192904750–SFB 992. German Federal Ministry of Education and Research [031A538A de.NBI-RBC awarded to R.B.; 031L0101C de.NBI-epi awarded to B.G.]. R.B. was supported by the German Research Foundation (DFG) under Germany's Excellence Strategy [CIBSS-EXC-2189-Project ID 390939984]. S.B. was funded by the European Open Science Cloud (EOSC-Life) (Grant No. 824087) and by the University of Freiburg. T.D., R.S. and F.v.D. were funded by the Diamond Light Source.

Availability of data and materials

All data is available in the form of published Galaxy histories. Links are provided in Additional file 1.

Declarations

Competing interests

The authors have no competing interests to declare.

Author details

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Freiburg im Breisgau, Germany. [2]Informatics Matters, Yew Tree Farm, High Street, Charlton on Otmoor, Kidlington, UK. [3]Diamond Light Source Ltd, Harwell Science and Innovation Campus, Didcot, UK. [4]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Freiburg im Breisgau, Germany. [5]Research Complex at Harwell, Harwell Science and Innovation Campus, Didcot, UK. [6]Structural Genomics Consortium, University of Oxford, Oxford, UK. [7]Department of Biochemistry, University of Johannesburg, Johannesburg, South Africa.

Received: 14 December 2021 Accepted: 9 February 2022
Published online: 12 April 2022

References

1. Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, Ohl P, Thiel K, Wiswedel B (2009) KNIME—the Konstanz Information Miner: version 2.0 and beyond. AcM SIGKDD Explor Newsl 11(1):26–31
2. Crusoe MR, Abeln S, Iosup A, Amstutz P, Chilton J, Tijanic N, Ménager H, Soiland-Reyes S, Goble CA (2021) Methods included: Standardizing computational reuse and portability with the Common Workflow Language. CoRR abs/2105.07028. arXiv:2105.07028
3. Tommaso PD, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C (2017) Nextflow enables reproducible computational workflows. Nat Biotechnol 35(4):316–319. https://doi.org/10.1038/nbt.3820
4. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Čech M, Chilton J, Clements D, Coraor N, Grüning BA, Guerler A, Hillman-Jackson J, Hiltemann S, Jalili V, Rasche H, Soranzo N, Goecks J, Taylor J, Nekrutenko A, Blankenberg D (2018) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res 46(W1):537–544. https://doi.org/10.1093/nar/gky379
5. Bray SA, Lucas X, Kumar A, Grüning BA (2020) The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform. J Cheminform. https://doi.org/10.1186/s13321-020-00442-7
6. Senapathi T, Bray S, Barnett CB, Grüning B, Naidoo KJ (2019) Biomolecular reaction and interaction dynamics global environment (BRIDGE). Bioinformatics 35(18):3508–3509. https://doi.org/10.1093/bioinformatics/btz107
7. Grüning B, Dale R, Sjödin A, Chapman BA, Rowe J, Tomkins-Tinch CH, Valieris R, Köster J (2018) Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat Methods 15(7):475
8. Conda-forge community: The conda-forge Project: Community-based Software Distribution Built on the conda Package Format and Ecosystem. Zenodo (2015). https://doi.org/10.5281/zenodo.4774217
9. da Veiga Leprevost F, Grüning BA, Aflitos SA, Röst HL, Uszkoreit J, Barsnes H, Vaudel M, Moreno P, Gatto L, Weber J, Bai M, Jimenez RC, Sachsenberg T, Pfeuffer J, Alvarez RV, Griss J, Nesvizhskii AI, Perez-Riverol Y (2017) BioContainers: an open-source and community-driven framework for software standardization. Bioinformatics 33(16):2580–2582. https://doi.org/10.1093/bioinformatics/btx192
10. Jin Z, Du X, Xu Y, Deng Y, Liu M, Zhao Y, Zhang B, Li X, Zhang L, Peng C, Duan Y, Yu J, Wang L, Yang K, Liu F, Jiang R, Yang X, You T, Liu X, Yang X, Bai F, Liu H, Liu X, Guddat LW, Xu W, Xiao G, Qin C, Shi Z, Jiang H, Rao Z, Yang H (2020) Structure of Mpro from SARS-CoV-2 and discovery of its inhibitors. Nature 582(7811):289–293. https://doi.org/10.1038/s41586-020-2223-y
11. Douangamath, A, Fearon D, Gehrtz P, Krojer T, Lukacik P, Owen CD, Resnick E, Strain-Damerell C, Aimon A, Ábrányi-Balogh P, Brandão-Neto J,

Carbery A, Davison G, Dias A, Downes TD, Dunnett L, Fairhead M, Firth JD, Jones SP, Keeley A, Keserü GM, Klein HF, Martin MP, Noble MEM, O'Brien P, Powell A, Reddi RN, Skyner R, Snee M, Waring MJ, Wild C, London N, von Delft F, Walsh MA, (2020) Crystallographic and electrophilic fragment screening of the SARS-CoV-2 main protease. Nat Commun. https://doi.org/10.1038/s41467-020-18709-w

12. Fragalysis developers: Fragalysis (2022) https://diamondlightsource.atlassian.net/wiki/spaces/FRAG/overview
13. Hall RJ, Murray CW, Verdonk ML (2017) The fragment network: a chemistry recommendation engine built using a graph database. J Med Chem 60(14):6440–6450. https://doi.org/10.1021/acs.jmedchem.7b00809
14. Leung S, Bodkin M, von Delft F, Brennan P, Morris G (2019) SuCOS is better than RMSD for evaluating fragment elaboration and docking poses. https://doi.org/10.26434/chemrxiv.8100203.v1
15. Scantlebury J, Brown N, Delft FV, Deane CM (2020) Data set augmentation allows deep learning-based virtual screening to better generalize to unseen target classes and highlight important binding interactions. J Chem Inf Model 60(8):3722–3730. https://doi.org/10.1021/acs.jcim.0c00263
16. Wolf S, Stock G (2018) Targeted molecular dynamics calculations of free energy profiles using a nonequilibrium friction correction. J Chem Theory Comput 14(12):6175–6182. https://doi.org/10.1021/acs.jctc.8b00835
17. Wolf S, Lickert B, Bray S, Stock G (2020) Multisecond ligand dissociation dynamics from atomistic simulations. Nat Commun. https://doi.org/10.1038/s41467-020-16655-1
18. GitHub contributors: Intergalactic Workflow Commission (2021) https://github.com/galaxyproject/iwc. GitHub
19. Yuen D, Cabansay L, Duncan A, Luu G, Hogue G, Overbeck C, Perez N, Shands W, Steinberg D, Reid C, Olunwa N, Hansen R, Sheets E, O'Farrell A, Cullion K, O'Connor BD, Paten B, Stein L (2021) The Dockstore: enhancing a community platform for sharing reproducible and accessible computational protocols. Nucleic Acids Res 49(W1):624–632. https://doi.org/10.1093/nar/gkab346
20. Goble C, Soiland-Reyes S, Bacall F, Owen S, Williams A, Eguinoa I, Droesbeke B, Leo S, Pireddu L, Rodríguez-Navas L, Fernández JM, Capella-Gutierrez S, Ménager H, Grüning B, Serrano-Solano B, Ewels P, Coppens F (2021) Implementing FAIR Digital Objects in the EOSC-Life Workflow Collaboratory. Extended abstract, submitted to Special issue on Canonical Workflow Frameworks for Research in the journal Data Intelligence. https://doi.org/10.5281/zenodo.4605654
21. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14(1):33–38. https://doi.org/10.1016/0263-7855(96)00018-5
22. Ropp PJ, Kaminsky JC, Yablonski S, Durrant JD (2019) Dimorphite-DL: an open-source program for enumerating the ionization states of drug-like small molecules. J Cheminform. https://doi.org/10.1186/s13321-019-0336-9
23. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR (2011) Open Babel: an open chemical toolbox. J Cheminform. https://doi.org/10.1186/1758-2946-3-33
24. Ruiz-Carmona S, Alvarez-Garcia D, Foloppe N, Garmendia-Doval AB, Juhos S, Schmidtke P, Barril X, Hubbard RE, Morley SD (2014) rDock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids. PLoS Comput Biol 10(4):1003571. https://doi.org/10.1371/journal.pcbi.1003571
25. Batut B, Hiltemann S, Bagnacani A, Baker D, Bhardwaj V, Blank C, Bretaudeau A, Brillet-Guéguen L, Čech M, Chilton J et al (2018) Community-driven data analysis training for biology. Cell Syst 6(6):752–758
26. Abraham MJ, Murtola T, Schulz R, Páll S, Smith JC, Hess B, Lindahl E (2015) GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 1–2:19–25. https://doi.org/10.1016/j.softx.2015.06.001
27. Case D, et al (2021) Amber 2021. https://ambermd.org/
28. Wang J, Wolf RM, Caldwell JW, Kollman PA, Case DA (2004) Development and testing of a general amber force field. J Comput Chem 25(9):1157–1174. https://doi.org/10.1002/jcc.20035
29. da Silva AWS, Vranken WF (2012) ACPYPE—AnteChamber PYthon parser interfacE. BMC Res Notes. https://doi.org/10.1186/1756-0500-5-367
30. Copeland RA, Pompliano DL, Meek TD (2006) Drug–target residence time and its implications for lead optimization. Nat Rev Drug Discov 5(9):730–739. https://doi.org/10.1038/nrd2082
31. Jarzynski C (1997) Nonequilibrium equality for free energy differences. Phys Rev Lett 78(14)
32. Wolf S (2020) Personal communication
33. GitHub contributors: Planemo (2021) https://github.com/galaxyproject/planemo. GitHub
34. Sloggett C, Goonasekera N, Afgan E (2013) BioBlend: automating pipeline analyses within Galaxy and CloudMan. Bioinformatics 29(13):1685–1686. https://doi.org/10.1093/bioinformatics/btt199
35. Dudgeon T (2021) Python script for interaction calculation. https://github.com/InformaticsMatters/pipelines/commits/master/src/python/pipelines/xchem/calc_interactions.py
36. Wójcikowski M, Zielenkiewicz P, Siedlecki P (2015) Open drug discovery toolkit (ODDT): a new open-source player in the drug discovery field. J Cheminform. https://doi.org/10.1186/s13321-015-0078-2

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

BMC

6 Planemo and BioBlend: Automating command-line data analysis with Galaxy

This chapter summarises the work originally described in the following publications:

- **Simon Bray**, John Chilton, Matthias Bernt, Nicola Soranzo, Marius van den Beek, Bérénice Batut, Helena Rasche, Martin Čech, Peter Cock, Björn Grüning, Anton Nekrutenko. Planemo: a command-line toolkit for developing, deploying, and executing scientific data analyses. *Genome Research* (under review), https://doi.org/10.1101/2022.03.13.483965
- Wolfgang Maier, **Simon Bray**, Marius van den Beek, Dave Bouvier, Nathan Coraor, Milad Miladi, Babita Singh, Jordi Rambla De Argila, Dannon Baker, Nathan Roach, Simon Gladman, Frederik Coppens, Darren P. Martin, Andrew Lonie, Björn Grüning, Sergei L. Kosakovsky Pond, Anton Nekrutenko. Ready-to-use public infrastructure for global SARS-CoV-2 monitoring. *Nature Biotechnology*, Volume 39, pages 1178–1179, 29 September 2021, https://doi.org/10.1038/s41587-021-01069-1

In addition, other related work is described, in particular relating to the BioBlend library.

6.1 Introduction

Galaxy differs from other workflow management systems such as Nextflow [132] or Snakemake [133] in that it is meant to be used primarily via a graphical interface. A drag-and-drop window is provided for constructing workflows, which can then be executed directly from the web browser. Nonetheless, there are situations in which using the graphical interface is not convenient or realistic, for example, analyses which require entire workflows to be run hundreds or even thousands of times. Examples include the projects discussed in the previous chapters on the T4L-L99A and SARS-CoV-2 M$^{\mathrm{pro}}$ systems, which required thousands of short molecular dynamics simulations to be performed in parallel. As a result, during the course of this thesis, a substantial amount of work was invested into developing useful methods for accessing Galaxy via the command-line, for example, with the BioBlend Python library, and the Planemo application, which were then used extensively during the projects described in previous chapters. These projects are not specialised for computational chemistry, but can be useful to any scientist using Galaxy who needs to scale up their analysis and is pushing against the limits of the graphical interface. One such use-case, a pipeline for SARS-CoV-2 variant surveillance, is discussed in this chapter.

6.2 Methods

6.2.1 BioBlend

Galaxy can be accessed via an application programming interface (API), which allows all functionality to be accessed programmatically. BioBlend is a Python library which allows access to the Galaxy API. BioBlend was used extensively during this thesis to handle the large number of simulations launched, to run tools and workflows over multiple histories, and to rename, reorganise and copy datasets between histories. As some functionality required for other projects was missing, several improvements were made to both the Galaxy API itself and the BioBlend library. These include:

- Rerun failed Galaxy jobs, remapping onto the datasets already created
- Change datatypes for all datasets in a collection
- Addition of an `InvocationClient` with methods to handle Galaxy invocations
- Methods for copying Galaxy datasets between histories
- Numerous bug fixes and tests

As a result of these contributions, I was assigned the role of maintainer of the BioBlend project in November 2020.

6.2.2 Planemo

Planemo is a Python library and application with a wide range of functionality, centred around the development and deployment of Galaxy tools and workflows. Several contributions were made to the Planemo software during the course of this thesis, and as a result, I received "committer" rights in December 2020.

Workflow execution

While Galaxy workflows allow organisation of multiple tools to form analyses, which already saves a considerable amount of time and effort, sometimes it is useful to go a step further and execute a workflow multiple times. For example, in the virtual screening study of the SARS-CoV-2 main protease (chapter 5), MMGBSA simulations were run for over 200 different compounds, necessitating the execution of the same workflow 200 times. As a result, a method to initiate workflows from the command line is required, as launching the runs from the graphical user interface manually 200 times is tedious and not scalable.

As a result, various functionality was incorporated into Planemo, focussed around its `run` subcommand, with the aim of streamlining the process of automated command-line workflow execution. Planemo provides either the option of spinning up a transient Galaxy instance locally, or executing jobs on an external Galaxy server via the API. The former is useful for developing and testing new Galaxy tools and workflows, but for any large scale analysis, access to the resources of a large external Galaxy server is essential. As part of this thesis, a user-friendly command line interface for running workflows on external Galaxy servers was integrated into Planemo. Users can specify the server and account from which a workflow should be run using their API key; users can also create separate "profiles", which save this information in a configuration file and thus allow workflows to be executed without specifying the user credentials each time. Galaxy workflows and datasets can be referenced by means of the hexanumeric IDs Galaxy assigns to them; analogously to the profile concept, users can define "aliases" to refer to workflows, to provide a more memorable handle. Upon execution, users may also choose to add "tags" to the newly created Galaxy histories; these tags provide a convenient way to organise and classify different analyses and are also visible in the graphical interface.

After execution, the progress of workflow invocations can be inspected using the `list_invocations` subcommand, which depicts the number of jobs for each invocation colour-coded by the state of the job (for example: running, failed, successful).

It is possible that jobs fail transiently, for example due to unexpected errors on the server, so a `rerun` subcommand is provided which automatically reruns failed Galaxy jobs. The rerun subcommand can be supplied with either job, history, on invocation IDs; in the last two cases, all failed jobs associated with the given history or invocation are collected and rerun.

Several options are provided for configuring workflow executions. By default, the workflow is not executed if the upload of one of the input datasets fails, but users may prefer to override this behaviour, when using very large collections. The user can also configure whether datasets should be uploaded one by one, waiting for each dataset to upload successfully before beginning with the next, or whether all datasets should be uploaded simultaneously; the latter approach is faster but may risk overloading the server. Users may prefer to wait for workflow execution to complete (in this case workflow results can also be downloaded to their local computer) or just to wait for the workflow to be successfully scheduled. In addition, the possibility is provided to separate data upload and workflow execution entirely using the `upload_data` subcommand; when this is run using a workflow, data is uploaded and a new job template file generated, containing the Galaxy IDs of the newly uploaded datasets. Workflow execution can then be triggered in a second step simply by using the `run` subcommand.

When execution is completed, a report is generated in HTML as well as JSON format, describing whether the workflow invocation completed successfully, describing input datasets and parameters, and listing all individual jobs. As it is provided in HTML format, it can be viewed easily from within a web browser. The JSON output can also be parsed programmatically for use in further analysis - for example, downstream workflows.

6.3 Results

6.3.1 Intergalactic Workflow Commission

The Intergalactic Workflow Commission (IWC) is a subgroup of the Galaxy project which aims to curate, maintain and preserve high-quality Galaxy workflows. I have been involved in the IWC "working group" since its creation and have made several contributions. These involved both submitting new workflows, for example the free energy workflows developed for virtual screening of the SARS-CoV-2 main protease, and contributions to the IWC infrastructure and tooling to make developing and contributing Galaxy workflows a simpler task for Galaxy users.

6.3.2 Automating tool and workflow updates

As described in Chapter 6, numerous Galaxy tools and workflows were developed as part of this thesis. Each of these required ongoing maintenance; whenever new versions of the underlying software tools were released, the Galaxy tools in general also needed updating. This maintenance was my responsibility, although it had a low priority compared to many of my other tasks. In order to reduce the maintenance burden, I wanted to investigate possibilities for automating the process of making these tool and software updates.

This work was inspired by the pre-existing BiocondaBot [134] project, a GitHub bot which automatically updates Bioconda packages, whenever a new version of the source code is released. The new version triggers BiocondaBot to make a PR to the Bioconda recipes repository, updating the download link and checksum. Approval from a community member is still required to merge, so this is a semi-automated solution; nonetheless, it is an extremely convenient and time-saving tool. As Galaxy tools specify dependency version numbers, similar to Bioconda recipes, a similar bot was envisaged for updating Galaxy tools and workflows. It should be noted that Bioconda packages, Galaxy tools and Galaxy workflows form a hierarchy, in that workflow have tools as dependencies and tools have Bioconda packages as dependencies. Thus, such a bot enables a chain of updates, from source code → Conda package → Galaxy tool → Galaxy workflow (Figure 6.1).

The code for the automatic tool update was implemented as a Planemo subcommand. This allows it to be used either by individual users or as part of a CI job. As a Galaxy tool may well have multiple dependencies, the first step is to identify the "main requirement". If this dependency is out-of-date, i.e. a newer version is available via Conda, all dependencies are updated. Galaxy tools should always incorporate tests and test data, and the bot is capable, as a user-configurable option, of subsequently also automatically updating the test data if necessary. As for BiocondaBot, merging is not done automatically but performed by a community member.

Compared to BiocondaBot, there are some additional complexities in implementing an autoupdate procedure for Galaxy tools. Galaxy tools make extensive use of macros and the dependency versions can be specified either in the tool definition file itself or within the macro. Thus, the bot checks both files and updates the relevant one. Secondary dependencies are updated only if the primary "main requirement" requires updating. The `autoupdate` subcommand provides a range of command-line options which allow configuration - for example, which Conda channels to check (by default, Bioconda and conda-forge), which tools or requirements to skip (Python, Perl and R are skipped by default), whether to update test data after updating the requirements. The PRs themselves are created by CI jobs, defined by GitHub Actions.

After successful deployment onto the two main GitHub repositories used to maintain computational chemistry tools, the bot was also deployed onto the much larger IUC (Intergalactic Utilities Commission) repository.

As the next step, an equivalent bot for automatically updating Galaxy workflows was developed. The work built on "refactor actions" provided in the Galaxy backend for modifying workflows, which are also accessible via the API. The `autoupdate` subcommand was extended to make use of these to spin up a temporary Galaxy server with the necessary tools installed, install the workflow to updated, run the refactor actions for updating all component tools and subworkflows, and downloading the updated workflow. Once Planemo had been extended, the new bot was also activated to update workflows maintained by the IWC.

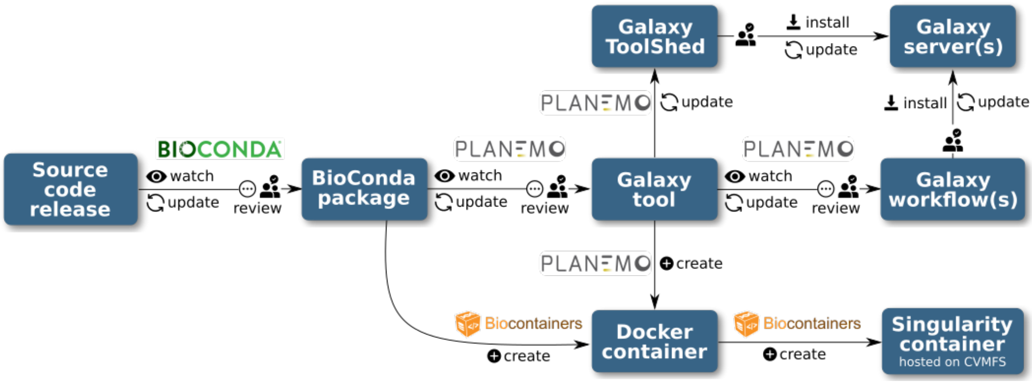


Fig. 6.1: Flowchart depicting the various automation steps, and their sequential arrangement, for Bioconda packages, BioContainers, Galaxy tools and Galaxy workflows. Information on the image source is provided in the List of Figures.

6.3.3 Implementing a continuous monitoring pipeline for SARS-CoV-2 variant surveillance

The projects described above are clearly not limited to usage in the field of computational chemistry, but for any scientific analysis which can be implemented in Galaxy. After the work done so far, I collaborated with colleagues in Freiburg and cooperation partners worldwide to implement infrastructure for continuous monitoring and download of newly published raw SARS-CoV-2 nucleotide reads, assembly and assignment to a viral lineage. My contribution here was writing the code for automatically running the workflows, building on the work described above developing BioBlend and Planemo. While the scientific field is clearly completely different to the rest of the work presented in this thesis, this project demonstrates the usefulness of BioBlend and Planemo to scientists across a range of disciplines.

In addition, a GitHub repository was created allowing scientists to request running workflows on arbitrary user-provided data: `https://github.com/usegalaxy-eu/sars-cov-2-processing-requests`. A pull request merely needs to be created containing a list of web links to the files to be analysed. Upon merging, these files are uploaded automatically to the European Galaxy server and subject to the assembly and lineage assignment workflows already mentioned.

6.4 Conclusion

As a result of this work, a paper was published describing the Planemo library and application. In addition, as part of this thesis, 33 pull requests were made and merged to the Planemo GitHub repository, 20 pull requests to the BioBlend repository, and 40 to the Galaxy repository itself.

Ready-to-use public infrastructure for global SARS-CoV-2 monitoring

Ready-to-use public infrastructure for global SARS-CoV-2 monitoring. Wolfgang Maier, **Simon Bray**, Marius van den Beek, Dave Bouvier, Nathan Coraor, Milad Miladi, Babita Singh, Jordi Rambla De Argila, Dannon Baker, Nathan Roach, Simon Gladman, Frederik Coppens, Darren P. Martin, Andrew Lonie, Björn Grüning, Sergei L. Kosakovsky Pond & Anton Nekrutenko. *Nature Biotechnology,* Volume 39, pp. 1178-1179, 27 September 2021, https://doi.org/10.1038/s41587-021-01069-1

Personal contribution:

The paper describes workflows and infrastructure for continuous SARS-CoV-2 variant surveillance on the Galaxy platform, starting from raw read data and executing multiple workflows to assign variants to each sample. I assisted with the development of a automated system ("bot") to collect newly published data and execute the analysis workflows, and made several necessary modifications to the Planemo library powering the bot. In recognition of these major contributions to the paper, I am listed as second author for the publication.

Co-authors:

Wolfgang Maier: designed tools and workflows, co-developed the analysis bot, wrote the paper.

Marius van den Beek: contributed to workflows and the analysis bot and administered usegalaxy.org compute resources used.

Dave Bouvier: contributed tools.

Nathan Coraor: administered usegalaxy.org compute resources used.

Milad Miladi: performed data analysis.

Babita Singh: data deposition and visualization.

Jordi Rambla De Argila: data deposition and visualization.

Dannon Baker: Galaxy development.

Nathan Roach: performed data analysis.

Simon Gladman: administered usegalaxy.org.au compute resources used.

Frederik Coppens: led the project and wrote the paper.

Darren P. Martin: led the project and wrote the paper.

Andrew Lonie: led the project and wrote the paper.

Björn Grüning: led the project and wrote the paper.

Sergei L. Kosakovsky Pond: led the project and wrote the paper.

Anton Nekrutenko: led the project and wrote the paper.

S. A. B

Simon Bray, 13.5.2022

Signatures:

The following co-authors confirm the above stated contribution:

Co-author	Date	Signature
Wolfgang Maier	20.6.2022	
Marius van den Beek	24.05.2022	
Björn Grüning	20.06.2022	
Anton Nekrutenko	07.06.22	

Due to the large number of authors, with most of whom I have never communicated personally, I collected signatures from only a selection of authors: Wolfgang Maier (first author), Marius van den Beek (third author), Björn Grüning and Anton Nekrutenko (corresponding authors). I hope this is acceptable to the doctoral committee, especially considering that my contribution to this paper forms only a small part of my dissertation.

Check for updates

Ready-to-use public infrastructure for global SARS-CoV-2 monitoring

To the Editor — The COVID-19 pandemic is the first health crisis characterized by large amounts of genomic data[1]. Computational infrastructure can be a bottleneck for data analysis, amplifying global inequalities in ability to track SARS-CoV-2 evolution. This is an issue even in developed countries, as computational infrastructure requires expertise in resource procurement, configuration and maintenance. Commercial computational clouds do not fully address the problem because these resources must still be configured and funded. Furthermore, commercial clouds are predominantly US-based and many countries have policies making payments to foreign providers impractical. In developing countries, research computing infrastructure is rare and researchers often cannot afford commercial cloud-based computation. Here, we present the COVID-19 effort by the Galaxy Project, which pools free worldwide public computational infrastructure, making the analysis of deep sequencing data accessible to anyone while also providing an analytical framework for global pathogen genomic surveillance based on raw sequencing-read data.

Despite the existence of well designed and validated SARS-CoV-2 data analysis approaches[2,3], the ad hoc[4] nature of their application often complicates the integration and comparison of analysis results. Public computational infrastructure (XSEDE, ELIXIR and Nectar Cloud in the United States, European Union and Australia, respectively) coupled with existing open-source software offers a solution to SARS-CoV-2 analytics challenges. However, glue is required to bind these resources into a unified platform for managing users, allocating storage and pairing analysis tools with appropriate computational resources. Such a platform is best not developed by a single principal investigator, group or institution, but rather supported by an international community of users, developers and educators.

We have developed a two-stage platform (Fig. 1) housed on three public Galaxy instances[5] in the United States (http://usegalaxy.org), the European Union (http://usegalaxy.eu) and Australia (http://usegalaxy.org.au) and capable of supporting hundreds of thousands of complex analyses per month. Anyone can run effectively unlimited computation with 250 Gb (expandable) of disk space. The COVID-19 Galaxy Project comprises two stages (Fig. 1): the software components of stage 1—mature utilities for quality control, mapping, assembly and allelic variant (AV) calling—run entirely in Galaxy and are distributed via the BioConda project[6]; the software components of stage 2 are snippets of code for data transformation, exploration and visualization running within standard web-browser-based notebook environments. Stage 1 produces variant lists whereas stage 2 uses notebooks to perform descriptive analyses of datasets. In addition, an interactive dashboard is available that tracks temporal AV dynamics. (See https://covid19.galaxyproject.org for data, workflows, notebooks, dashboard and our ongoing automated tracking of large-scale genomic surveillance projects.)

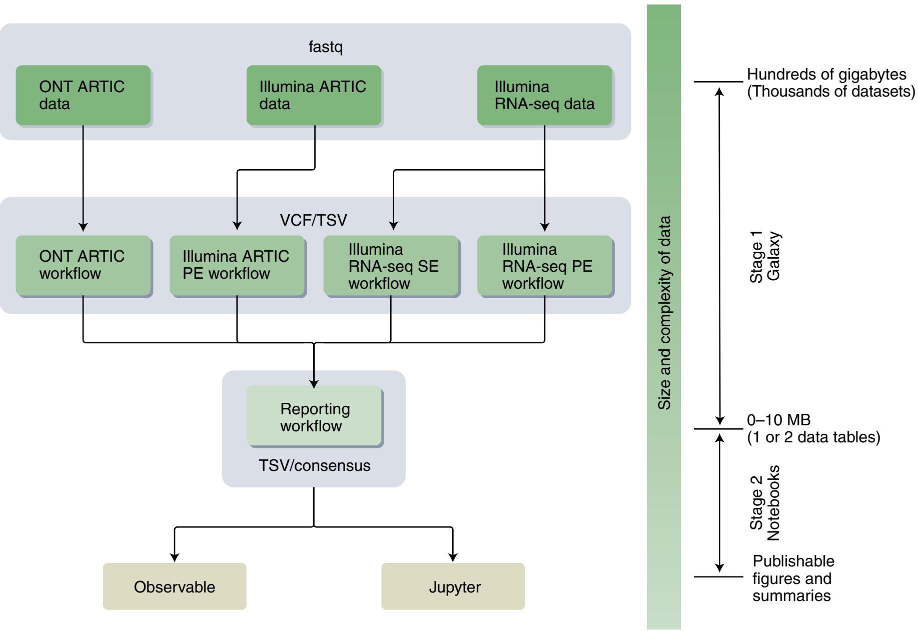

fastq
ONT ARTIC data
Illumina ARTIC data
Illumina RNA-seq data
VCF/TSV
ONT ARTIC workflow
Illumina ARTIC PE workflow
Illumina RNA-seq SE workflow
Illumina RNA-seq PE workflow
Reporting workflow
TSV/consensus
Observable
Jupyter
Size and complexity of data
Hundreds of gigabytes (Thousands of datasets)
Stage 1 Galaxy
0–10 MB (1 or 2 data tables)
Stage 2 Notebooks
Publishable figures and summaries

Fig. 1 | **Analysis flow for calling SARS-CoV-2 variants using Galaxy.** ONT, Oxford Nanopore Technologies; VCF, variant call format; TSV, tab-separated values; PE, paired end; SE, single end. For more information, see https://covid19.galaxyproject.org.

Four primary analysis workflows (Supplementary Table 1) support the identification of SARS-CoV-2 AVs from deep-sequencing reads via the production of annotated AVs through a series of steps including quality control, trimming, mapping, deduplication, AV calling and filtering. Their output is processed by the Reporting and Consensus workflows (Supplementary Table 1) to generate standardized data tables describing AVs along with consensus genome sequences. These are further processed to summarize and visualize the data using interactive notebooks.

To illustrate the platform's utility and scalability, we refer the reader to two large SARS-CoV-2 Illumina datasets (PRJNA622837, 619 samples from early SARS-CoV-2 transmission in the Boston area[7]; and PRJEB37886, ~100,000 samples analyzed as of the time of writing from the COVID-19 Genomics UK (COG-UK) genomic surveillance effort[8]) detailed in Supplementary Tables 1–3 and Supplementary Figs. 1–3. Analysis on COVID-19 Galaxy Project resources provides insights into co-occurrence patterns, presence of mutations defining variants of concern (https://cov-lineages.github.io/lineages-website/global_report.html), and intersection with sites under selection, including non-random associations among common low-frequency

AVs that may reflect shared intra-host dynamics (Supplementary Fig. 1 and Supplementary Table 2). It can also highlight the emergence of mutations interfering with binding of polyclonal antibodies[9] (for example, COG-UK data in Supplementary Fig. 2), suggesting possible intra-host dynamics. These and other interactive notebooks and dashboards on the platform could identify AVs that warrant closer monitoring as the pandemic continues.

Our system is designed to encourage scalable collaborative worldwide genomic surveillance to identify and respond to emerging variants. By relying on raw read data rather than assembled genomes and allowing every result to be traced back to its raw data, it goes a step beyond current surveillance efforts. Specifically, it enables surveillance of intra-patient minor AV frequencies—a distinction that could yield early warnings of epidemiological conditions conducive to the emergence of variants with altered pathogenicity, vaccine sensitivity or drug resistance.

Wolfgang Maier[1], Simon Bray[1], Marius van den Beek[2], Dave Bouvier[2], Nathan Coraor[2], Milad Miladi[1], Babita Singh[3], Jordi Rambla De Argila[3], Dannon Baker[4], Nathan Roach[5], Simon Gladman[6], Frederik Coppens[7,8], Darren P. Martin[9], Andrew Lonie[6], Björn Grüning[1]✉, Sergei L. Kosakovsky Pond[10]✉ and Anton Nekrutenko[2]✉

[1]University of Freiburg, Freiburg, Germany. [2]The Pennsylvania State University, University Park, PA, USA. [3]GalaxyWorks Inc, Baltimore, MD, USA. [4]Centre for Genomic Regulation, Viral Beacon Project, Barcelona, Spain. [5]Johns Hopkins University, Baltimore, MD, USA. [6]University of Melbourne, Melbourne, Victoria, Australia. [7]Ghent University, Ghent, Belgium. [8]VIB Center for Plant Systems Biology, Ghent, Belgium. [9]University of Cape Town, Cape Town, South Africa. [10]Temple University, Philadelphia, PA, USA.

✉*e-mail: gruening@informatik.uni-freiburg.de; spond@temple.edu; aun1@psu.edu*

Published online: 29 September 2021
https://doi.org/10.1038/s41587-021-01069-1

References

1. Hodcroft, E. B. et al. *Nature* **591**, 30–33 (2021).
2. Quick, J. et al. *Nat. Protoc.* **12**, 1261–1276 (2017).
3. Grubaugh, N. D. et al. *Genome Biol.* **20**, 8 (2019).
4. Baker, D. et al. *PLoS Pathog.* **16**, e1008643 (2020).
5. Jalili, V. etal. *Nucleic Acids Res.* **48** W1, W395–W402 (2020).
6. Grüning, B. et al. *Nat. Methods* **15**, 475–476 (2018).
7. Lemieux, J. et al. *Science* https://doi.org/10.1126/science.abe3261 (2021).
8. du Plessis, L. et al. *Science* **371**, 708–712 (2021).
9. Greaney, A. J. et al. *Cell Host Microbe* **29**, 463–476.e6 (2021).

Acknowledgements

The authors are grateful to the broader Galaxy community for their support and software development efforts. This work is funded by NIH grants U41 HG006620 and NSF ABI grant 1661497. Usegalaxy.eu is supported by the German Federal Ministry of Education and Research grants 031L0101C and de.NBI-epi to B.G. Galaxy and HyPhy integration is supported by NIH grant R01 AI134384 to A.N. Usegalaxy.org.au is supported by Bioplatforms Australia and the Australian Research Data Commons through funding from the Australian Government National Collaborative Research Infrastructure Strategy. The hyphy.org development team is supported by NIH grant R01GM093939. Usegalaxy.be is supported by the Research Foundation-Flanders (FWO) grant I002919N and the Flemish Supercomputer Center (VSC). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41587-021-01069-1.

Peer review information *Nature Biotechnology* thanks Jason Sahl for their contribution to the peer review of this work.

Check for updates

Rapid delivery systems for future food security

To the Editor — The current world population of 7.8 billion is predicted to reach 10 billion by 2057 (https://www.worldometers.info/world-population/#pastfuture). Future access to affordable and healthy food will be challenging, with malnutrition already affecting one in three people worldwide. The agricultural sector currently provides livelihoods for 1.1 billion people and accounts for 26.7% of global employment (https://data.worldbank.org/indicator/SL.AGR.EMPL.ZS). However, our reliance on a small number of crop species for agricultural calorie production and depletion of land, soil, water and genetic resources, combined with extreme weather events and changing disease/pest dynamics, are already jeopardizing future food security[1]. Climate change–induced reductions in the global yield of major crops (for example, rice, wheat, maize and soybean) are more pronounced in low-latitude regions and thus affect farmers in developing countries[2]. As is evident from temperate cereal crops, a robust seed system that delivers improved cultivars to replace old cultivars is a plausible approach to adapting agriculture to climate change[3]. Here we provide an overview of how seed input supply systems and new production and harvesting technologies can generate increased incomes for developing world farmers and deliver better products to consumers.

Crop improvement remains crucial to the United Nations' Sustainable Development Goal 2 (SDG 2) of 'Zero Hunger: ending malnutrition and achieving food security by 2030'. It offers sustainable solutions for food production and food security by creating high-yielding, nutritious crops that can withstand emerging biotic and abiotic stresses. Innovative crop breeding techniques that accelerate the breeding cycle (for example, speed breeding[4]), facilitate more precise genetic combinations (for example, genomic selection[5]) and enable precise genetic changes (for example, genome editing[6]) provide unprecedented opportunities for enhancing crop performance in controlled conditions and research plots[7]. Translating crop productivity gains from experimental settings to real-world farming conditions requires improving equitable access to innovative technologies for all farmers and providing legislative, economical and practical support to ensure their adoption[8].

After the development of better-performing varieties, several steps are required to realize higher crop yields and income for smallholder farmers and deliver enhanced agricultural outputs (Fig. 1). The integration of planting good-quality seeds of elite crop varieties with improved decision support tools, mechanical harvesting and post-harvest management will increase production gains. Electronic trading portals (for example, Wefarm (https://about.wefarm.com/), eNAM (https://www.enam.gov.in/web/) and Digital Mandi (https://www.iitk.ac.in/MLAsia/digimandi.htm)) and support from farmer associations should help farmers market their produce directly for fairer prices. Further processing and addition of value can also deliver improved products to consumers and increase farmer's income (Fig. 1).

Seed is the single entry point for crop resilience and productivity. The sustainability of crop production is vitally dependent on the timely supply of improved seed and other inputs. In developing countries, formal seed supply systems generally do not meet farmers' demands, such that smallholder farmers source more than 80% of their seed from

Planemo: a command-line toolkit for developing, deploying, and executing scientific data analyses

Planemo: a command-line toolkit for developing, deploying, and executing scientific data analyses. **Simon Bray**, Matthias Bernt, Nicola Soranzo, Marius van den Beek, Bérénice Batut, Helena Rasche, Martin Čech, Peter Cock, Anton Nekrutenko, Björn Grüning, John Chilton. *Genome Research* (submitted), https://doi.org/10.1101/2022.03.13.483965

Personal contribution:

The paper describes Planemo, a library and application written in Python, which assists in the development, deployment and execution of scientific software tools and workflows in the Galaxy platform. I made two major contributions. Firstly, I added numerous customisable features to enable execution of workflows on large public Galaxy servers via the command-line. Secondly, I added functionality for automating updates to Galaxy tools and workflows. Furthermore, I contributed various bugfixes and features and wrote the submitted manuscript. In recognition of these major contributions to the paper, I am listed as first author for the publication.

Co-authors:

Matthias Bernt: developed the Planemo CI GitHub Action, contributed to the Planemo software and paper

Nicola Soranzo: contributed to the Planemo software and paper

Marius van den Beek: contributed to the Planemo software and paper

Bérénice Batut: contributed features for developing Galaxy training material, contributed to the paper

Helena Rasche: contributed to the Planemo software and paper

Martin Čech: contributed to the Planemo software and paper

Peter Cock: contributed to the Planemo software and paper

Anton Nekrutenko: led the project, contributed to the paper

Björn Grüning: led the project, contributed to the Planemo software and paper

John Chilton: initiated the project, created the project architecture and the majority of features, contributed to the paper

S. A. B

Simon Bray, 13.5.2022

Signatures:

The following co-authors confirm the above stated contribution:

Co-author	Date	Signature
Matthias Bernt	19th June 2022	
Nicola Soranzo	24th May 2022	Nicola Soranzo
Marius van den Beek	**24.05.2022**	
Bérénice Batut	2022-05-30	Bérénice Batut
Helena Rasche	2022-06-02	
Martin Čech	24.5.2022	
Peter Cock	2022-05-25	Peter Cock
Anton Nekrutenko	07.06.22	
Björn Grüning	20.06.2022	
John Chilton	2022/05/22	

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Planemo: a command-line toolkit for developing, deploying, and executing scientific data analyses

Simon Bray[1], Matthias Bernt[2], Nicola Soranzo[3], Marius van den Beek[4], Bérénice Batut[1], Helena Rasche[5], Martin Čech[6], Peter Cock[7], Anton Nekrutenko[4], Björn Grüning[1] and John Chilton[4]

1. Bioinformatics Group, Department of Computer Science, Albert-Ludwigs-University Freiburg, Georges-Köhler-Allee 106, 79110 Freiburg, Germany
2. Department of Computational Biology, Helmholtz Centre for Environmental Research GmbH - UFZ, Permoserstraße 15, 04318 Leipzig, Germany
3. Earlham Institute, Norwich Research Park, Norwich, NR4 7UZ, UK
4. Department of Biochemistry & Molecular Biology, The Pennsylvania State University, University Park, PA, USA
5. Avans Hogeschool, Breda, Netherlands
6. Institute of Organic Chemistry and Biochemistry, Prague, Czech Republic
7. James Hutton Institute, Invergowrie, Dundee DD2 5DA, UK

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Abstract

There are thousands of well-maintained high-quality open-source software utilities for all aspects of scientific data analysis. For over a decade, the Galaxy Project has been providing computational infrastructure and a unified user interface for these tools to make them accessible to a wide range of researchers. In order to streamline the process of integrating tools and constructing workflows as much as possible, we have developed Planemo, a software development kit for tool and workflow developers and Galaxy power users. Here we outline Planemo's implementation and describe its broad range of functionality for designing, testing and executing Galaxy tools, workflows and training material. In addition, we discuss the philosophy underlying Galaxy tool and workflow development, and how Planemo encourages the use of development best practices, such as test-driven development, by its users, including those who are not professional software developers. Planemo is a mature project widely used within the Galaxy community which has been downloaded over 80,000 times.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Introduction

The Galaxy project provides web browser access to command-line scientific software, together with the necessary compute resources, in a convenient, shareable and reproducible way, to researchers around the world [1]. Over eight thousand tools are available for installation onto any Galaxy server; users can run these individually, connect multiple tools together to form workflows, and finally perform complex analyses, without the need to access a command line. While Galaxy itself does not require any significant computational skills to use, development and maintenance of new tools and workflows benefit from sophisticated infrastructure with both human and automated components. The process of integrating software into Galaxy requires knowledge of both the command-line interface of the underlying software and the schema used by Galaxy to define tools, in order to be able to write a ‘Galaxy tool wrapper’ mapping dataset inputs, parameter inputs and outputs between them. Once written, wrappers, as well as other Galaxy artifacts such as workflows or training material [2], are amenable to routine processes such as testing, deployment and regular updates, all of which can be automated using continuous integration (CI) systems. Here we present Planemo, a versatile library and command line application which is used extensively as a software development kit by Galaxy or Common Workflow Language (CWL) [3] tool, workflow and training material developers, and as a toolkit for Galaxy ‘power users’. Planemo provides a simple but powerful command-line interface for tool and workflow development and deployment, which encourages and enforces good practices for software development. In addition, it enables automated deployment of developed tools and automatic updates of the software dependencies used internally by each Galaxy tool. The testing functionality included in Planemo has been successfully integrated into CI workflows of the major tool and workflow repositories, which helps to ensure the creation of high quality tool wrappers and workflows.

Planemo is structured into numerous subcommands, which provide a broad range of functionality. Here we discuss a selection of the most important functionalities, grouped around the following themes: 1) development of Galaxy tools, workflows, tutorials, and CWL tools; 2) deployment of the developed tools and workflows; 3) automated tool and workflow dependency updates and 4) tool and workflow execution. Table 1 summarizes this functionality, and Fig. 1 provides a graphical overview. In addition to its use as a command-line application, Planemo can also be used as a library by other projects. An example is the Planemo Training Development Kit project (https://github.com/galaxyproject/ptdk), which provides Planemo’s functionality for creating training material for Galaxy workflows via a webserver.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Methods

Software design

Planemo is implemented as a Python package and distributed via GitHub, PyPI and Bioconda [4]. As already described in the Introduction, Planemo is a highly flexible, multifunctional software, which can be used for: 1) different types of artifacts (e.g. tools, workflows), 2) different workflow/tool languages and management systems (e.g. Galaxy, CWL), 3) different tasks (e.g. linting, testing, executing). To handle this variety, Planemo defines two central abstractions: Runnables and Engines. Runnables include tools and workflows written for either Galaxy or CWL; an Engine provides access to an external piece of software (such as Toil or Galaxy) capable of executing a particular Runnable. Each Engine has various methods (e.g. run(), test()), which define a particular interaction with a Runnable.

Engines are provided for both local and external Galaxy servers, as well as for cwltool [5] and Toil [6]. These interact with their respective workflow management systems via the cwltool and Toil Python modules (for CWL), and via the BioBlend library [7], which provides access to the Galaxy API through Python. Numerous lower-level functions and classes are provided to connect the Engines with the underlying functionality.

Some tasks cannot be easily described in the context of these abstractions; for example, linting of tool or workflow definitions requires only that the structured document containing the definition be compared with a schema. Other examples include the functionality for automatic updates of software dependencies and generation of training material. Planemo handles these cases using separate classes and functions.

Planemo is most frequently used as a command-line application, using a command-line interface written using the Click package to provide a straightforward way to access the components described above. Multiple subcommands expose some of the most important tasks a user might want to perform. For example, a user could run `planemo test tool.xml` to test a Galaxy tool wrapper. Planemo will detect the type of Runnable (Galaxy tool) represented by the filepath and start the appropriate Engine (temporary local Galaxy instance), execute the Runnable on it, collect the results, and compare them to predefined test data to determine a pass or fail status. All subcommands can be configured by appending flags and options.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Implementation of continuous integration jobs

While Planemo is designed primarily with developers and users in mind, commands often need to be executed as part of automated continuous integration (CI) jobs – for example, testing of newly created Galaxy tools after submission to a GitHub repository. Galaxy tools and workflows are hosted over multiple repositories; to ensure a unified approach to testing, a GitHub CI action is provided. The CI workflow consists of the following components:

1. Identifying modified tools and repositories using `` `planemo ci_find_repos` `` and `` `planemo ci_find_tools` ``.
2. Linting of Galaxy tools using `` `planemo lint` ``.
3. Testing the tools – as this is the most time-consuming step, the tools found are chunked and multiple jobs run in parallel.
4. Linting of Python and R scripts packaged together with the tools.
5. If the PR is approved and merged: deployment to the Toolshed with `planemo shed_update`.

Definition of terms

Planemo's features rely on and are interdependent with a variety of other subprojects within and related to the Galaxy community. We therefore first outline a few of these.

IUC: The Intergalactic Utilities Commission [8] maintains a central repository of Galaxy tool wrappers, currently hosted on GitHub. New wrappers are added by means of a GitHub pull request, reviewed by IUC members, and are tested by automated CI. After approval, the tool is automatically deployed to the Galaxy ToolShed. Tools are subject to further automatic updates, as new versions of software dependencies are released. The IUC serves as a model for smaller communities developing wrappers for more specialized tools (for example, Galaxy-P [9] for proteomics) and has developed a set of guidelines for tool development.

Bioconda/BioContainers: Each Galaxy tool has certain dependencies, which are typically installed either using the Conda package manager [10] or within a container (Docker [11] or Singularity [12]). Development and maintenance of the necessary Conda packages or containers is performed by the Bioconda and Biocontainers [14] communities, which collaborate closely with the Galaxy project.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

ToolShed: A central ‘app store’ for Galaxy tools Any user can upload to the ToolShed [13], but most high-quality tools are developed collaboratively on an open platform like GitHub (for example by the IUC) and deployed automatically.

IWC: maintains a set of curated workflows [14], consisting of multiple component Galaxy tools, which are hosted on GitHub and deployed to Dockstore [15] and the Workflow Hub [16], analogously to the development and deployment of Galaxy tools to the ToolShed by the IUC.

Galaxy Training Network: A repository for tutorials, each describing a method for data analysis in Galaxy [1]. Each tutorial is made up of multiple steps and therefore corresponds to a Galaxy workflow, which forms the skeleton around which the tutorial is built.

Continuous Integration (Workflow): A workflow run remotely on a build server which tests and deploys Galaxy artifacts developed. It should not be confused with a Galaxy workflow.

Tool: Artifact defined by a tool wrapper and stored in the ToolShed, allowing users to access the functionality of the underlying software via Galaxy.

Galaxy Tool Wrapper: Structured document defining a Galaxy tool; it maps dataset inputs and outputs and other parameters between the underlying command-line tool and the Galaxy API.

Galaxy Workflow: a directed acyclic graph in which nodes can be dataset inputs or outputs, parameter inputs, or tools. More informally, a combination of multiple individual tools into a single pipeline, which once assembled can be executed as if it were a single tool.

Collection: a group of individual datasets linked together in a directory-like structure. When a tool is run on a collection, individual jobs are generated for each of the datasets which make up the collection. In combination with workflows, collections allow Galaxy users to scale up analyses to deal with large sets of data.

Documentation

Planemo’s documentation is hosted on a ReadTheDocs site: https://planemo.readthedocs.io. In addition, several tutorials are available as part of the Galaxy Training Network:

- Creating Galaxy tools from Conda through deployment: https://training.galaxyproject.org/training-material/topics/dev/tutorials/tool-from-scratch/tutorial.html

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- Creating training material with Planemo: https://training.galaxyproject.org/training-material/topics/contributing/tutorials/create-new-tutorial/tutorial.html

- Automating Galaxy workflows using the command line: https://training.galaxyproject.org/training-material/topics/galaxy-interface/tutorials/workflow-automation/tutorial.html

- Test-driven development with Planemo: https://planemo.readthedocs.io/en/latest/writing_advanced.html#test-driven-development

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Results and Discussion

Galaxy tool development

A Galaxy tool is defined by a wrapper for an underlying software (or code), which maps its dataset inputs, parameter inputs and outputs to a command-line script executed by Galaxy. When running a tool in the Galaxy interface, a user selects their preferred choices for the exposed dataset and parameter inputs. The Galaxy server then constructs the command, schedules it as a job onto appropriate compute resources, collects the results once the job has completed, and returns them to the user.

Writing Galaxy tool wrappers requires a thorough knowledge of the underlying software and also an understanding of the Galaxy tool schema which defines how Galaxy wrappers are written. The tool schema is defined in a simple manner, in order to make the process of wrapping software as accessible as possible [17]. Planemo provides several helpful features which assist tool developers in creating high-quality wrappers that meet community-defined standards, such as those [18] developed by the Intergalactic Utility Commission (IUC). These features are implemented as subcommands, e.g. `planemo test`. Planemo also helps to enforce software development best practices such as writing tests for all tools and linting the wrapper definitions to avoid bugs and ensure a coherent and readable style. Further support for tool development standards is provided by the Galaxy Language Server [19], an implementation of the Language Server Protocol [20] and a Visual Studio Code extension for Galaxy tools, which can be used side-by-side with Planemo.

A common starting point for tool development is the `tool_init` subcommand. To use this, the developer provides a variety of options, including an example command line, tool name, inputs, outputs and software requirements, from which Planemo generates a skeleton tool wrapper. Most of the `tool_init` parameters are optional, but the more that are provided, the more detailed the initial skeleton will be.

The developer can then inspect and edit the generated file, adding more parameters and increasing the complexity of the wrapper logic by incorporating conditionals and repeat elements if necessary. As they continue to edit, they can use the `lint` subcommand to validate the wrapper under development. Planemo's linting forces wrappers to match Galaxy's tool schema, ensuring stylistic consistency and preventing some errors such as mismatched file formats. Crucially, Planemo recommends that wrappers define at least one test case to ensure

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

the development of high-quality, portable, reliable and functional tools, and this recommendation is strictly enforced by the IUC's and other tool repositories. Once tests are defined, together with an initial tool definition, the developer can start to run the tests using the `test` subcommand. This launches a transient Galaxy server on the developer's computer, installs the Galaxy tool under development, together with all software dependencies, and executes the tests specified within the tool wrapper. The results of the tests are then returned to the developer, by default using a report defined using JSON and HTML, although other format types are also supported (xUnit, jUnit, Markdown and Allure).

Planemo encourages the use of test-driven development [21], a software development principle which states test cases should be written before a new feature is developed. Test-driven development is an industry-wide best practice. Defining extensive test cases at the start of the process covering the required features provides a focus for development, and results in more robust and better documented code containing fewer bugs. The tool developer is forced to adopt the perspective of the Galaxy user from the start to consider possible use-cases of the software for which tests need to be written. Initial test failures lead to iterative refinement of the wrapper, until a fully-functional Galaxy tool, which passes all tests, is produced.

Once tests are passing, the developer should optimize the tool interface which is presented to the user of the tool. To facilitate this, Planemo provides the `serve` subcommand, which launches a Galaxy server with the new tool installed, allowing the developer to inspect the rendering of the wrapper in the graphical interface and to perform manual testing. The developer should also improve the documentation of the tool, by annotating each of the tool parameters, as well as writing a help section to explain the tool's aim and usage, which appears beneath the tool parameters in the graphical interface.

Common Workflow Language tool development

In addition to Galaxy tools, Planemo also acts as a software development kit for CWL tools. The same subcommands described can be used for this purpose, including `tool_init` and `test`. By appending the `--cwl` argument to the `tool_init` subcommand, Planemo generates a template for a CWL tool definition, rather than a Galaxy wrapper. The test and lint commands then detect that the input file is a CWL wrapper and process it accordingly. Tools are tested by executing with the CWL engine cwltool and comparing the result with test data or specified assertions, in the same way as for Galaxy tools. The completed wrapper can be run using any CWL engine, such as cwltool, Toil, Arvados [22] or Galaxy.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Galaxy workflow development

Workflows are created in Galaxy by connecting together multiple tools (i.e. an output of one tool becomes an input for the following one) in order to automate complex analyses. Unlike tools, workflows can be defined and edited in Galaxy's graphical workflow editor; often the starting point is an interactive analysis (a Galaxy history) from which a workflow can be extracted automatically. It is also possible to manually author workflows in the gxformat2 workflow language [23], and the user can switch between manually writing workflows and editing in the graphical interface using the `` `workflow_edit` `` subcommand, which spins up a Galaxy instance with the workflow under development pre-installed for editing. Planemo additionally facilitates the creation of test cases by providing the option of generating them automatically from a pre-existing workflow invocation.

Once a draft version of the workflow exists, it should be iteratively improved in the same way as for tools, using the same lint, test and serve subcommands already introduced. The `` `workflow_lint` `` subcommand checks workflows for errors and conformance with best practices—a command-line interface mirroring functionality which is also provided by the Galaxy graphical workflow editor. For example, workflows which are missing test cases, labeled outputs, or essential metadata fail linting. Running the `` `test` `` subcommand launches a local Galaxy instance, installs the tools used in the workflow, uploads the workflow and executes it on the provided input test data. In the same way as for tool testing, the workflow outputs are downloaded and compared to the test data, resulting in either a pass or fail status. In some cases, it can be convenient to run testing on an existing public server, such as https://usegalaxy.org, https://usegalaxy.eu, or https://usegalaxy.org.au; this is also supported by Planemo. Running the `` `serve` `` subcommand provides a local Galaxy server with the workflow and the needed tools pre-installed, which can be used for workflow development and fine-tuning.

The philosophy of Galaxy tool and workflow development

After the previous discussion of the process of tool and workflow development, the question arises how software complexity should be divided between the tool and the workflow level. Should most of the effort go into developing workflows, keeping tools as simple as possible and flexibly rewrapping the underlying software depending on the demands of a particular workflow, or should developers invest time creating complex and multifunctional tools which can be reused without modification in multiple workflows?

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Galaxy leans heavily towards the second of these two options, as does CWL, though the following discussion will focus on Galaxy. Galaxy encourages the creation of modular tools which are usable in isolation, so they can be used interchangeably in multiple different workflows. Tools generally encapsulate most of the complexity of the underlying software, allowing workflows to be simply constructed in a graphical interface by connecting the component tools. Workflows can thus be thought of as complex structures built from the same fundamental building blocks, which can be constructed without knowledge of the internal functionality of the individual tools. This has several advantages with regard to the user experience: building workflows becomes a far less daunting task, and tools can also be used individually in the graphical interface, which makes Galaxy accessible to new users and enables its use as a teaching environment for scientific analysis.

Another advantage of this approach is the “separation of concerns”, a design principle in computer science. Different groups of scientists can develop and apply specialized and complementary areas of knowledge: the tool developer can concentrate on describing and developing the Galaxy tool, without considering any downstream workflows that will be created later. On the other hand, the workflow developer can construct complex, high-level pipelines, without the detailed understanding of the component tools and the command-line possessed by the tool developer. This has the dual advantage that workflows can be treated on a more abstract level and that the workflow creation process is made accessible for a far greater number of users.

Separation of concerns between tools and workflows also benefits security. Executing untrusted software on a compute cluster is highly undesirable; thus workflows need to be assessed for security risks before execution. For many workflow management systems, this assessment must be repeated for each workflow. By contrast, as the Galaxy tool review process involves checking tools for security issues before merging, a system administrator can deploy tools developed by the IUC or similar high-trust communities with confidence. The question of workflow security is thus made redundant: if the component tools are trusted, a workflow based on those tools can likewise be trusted.

These advantages must be balanced against the time investment required from community members to build up a diverse set of tools, to allow the construction of scientifically interesting workflows. Nonetheless, the Galaxy community, facilitated by Planemo, has succeeded in developing such a toolset and making it available to the scientific community.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Continuous integration for community repositories

Galaxy has a large and vibrant community of tool and workflow developers, creating Galaxy tools in a wide range of scientific fields, ranging from genomics to proteomics, computational chemistry and climate science. As a result, a large number of high quality tools already exist and are actively maintained over several GitHub repositories, centered around the main IUC repository; the IWC (see Methods for definition) performs the equivalent function of a repository for Galaxy workflows. Building these communities has required many years of work by multiple contributors; in order to streamline the process and ease the burden on the tool developers, developing infrastructure to facilitate human review and automate as much as possible is essential. Planemo forms the core of this infrastructure.

Once a developer has completed the tool wrapper or workflow, they can submit it to a community repository, usually hosted on GitHub, for review. Alternatively, they may also deploy it themselves (for example, to the ToolShed or WorkflowHub), but submission to a community repository is encouraged to ensure the code is thoroughly reviewed and to publicize the new tool or workflow. Community repositories are configured to run the linting and testing checks already described after submission, via a continuous integration (CI) workflow. Planemo provides a couple of simple subcommands, `ci_find_repos` and `ci_find_tools`, to identify tools which have been added or modified. Both of these allow chunking of tools in order to parallelize the testing process over multiple CI jobs. As part of the CI testing, linting and testing of the tools is repeated, as well as linting of any Python and R scripts added together with the new tool wrappers. These steps ensure the submitted tools are of high quality, enforce consistent standards on the code and reduce the maintenance burden for the entire community.

If all tests pass and the proposed new tool or workflow is accepted by the community, another CI job is initiated to deploy it to the ToolShed. This makes use of Planemo's `shed_update` command, which uses the ToolShed credentials associated with the repository to upload the newly created tool. Once it is available on the ToolShed, it can easily be installed onto any Galaxy server.

The entire process, consisting of automated testing, human review and automated deployment, ensures the creation of high-quality, trustworthy tools which can be safely installed and used. It requires several more specialized steps, which go beyond the simple Planemo subcommands that the developer runs on their local machine. To package these CI workflows into a single unit, a GitHub Action is provided [24] which can be reused in other tool repositories. New tool

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

repositories with the same structure as the IUC repository can be conveniently created from a template repository created by the Galaxy community [25].

Automation of tool and workflow updates

Another feature offered by Planemo is automatic updates of Galaxy tool and workflow software dependencies, using the `autoupdate` subcommand. In combination with separate autoupdate features already developed by the Bioconda and conda-forge [26] communities, this forms a sequence of semi-automated software update procedures, which are triggered by an official release of new source code. After this new release appears, this chain ensures that new Conda packages, new Docker and Singularity containers, updated Galaxy tools and finally updated Galaxy workflows are generated (Fig. 2). At each step, a CI job detects the artifact published in the previous step and initiates the process of updating a dependent artifact, generally by means of a GitHub pull request (PR).

The CI pipelines developed by Bioconda and conda-forge monitor the Conda recipes they maintain, regularly checking the links provided in the recipes for new releases. When the developers of an upstream software package release a new version, the CI creates a PR to update the package recipe. Once the PR is reviewed and merged, newly built packages are uploaded to the Anaconda repository.

In parallel, a bot [29] running the `autoupdate` subcommand monitors the Galaxy tool wrappers maintained by the IUC, as well as a few other smaller communities, checking the dependencies defined in the tool wrapper. Once an updated Bioconda or conda-forge package is published in the step above, the Planemo autoupdate bot detects this and updates the dependencies section of the Galaxy tool accordingly. A PR is then submitted to the GitHub repository, to be reviewed and manually updated if necessary, before it is merged and deployed as described in the “CI for community repositories” section.

Galaxy tools can specify multiple dependencies. If these dependencies are installed via Conda, the packages can be simply installed into a single environment, but if dependency installation is achieved using containers, a new container must be built for each required combination of dependencies. This is achieved by the ‘mulled build’ infrastructure; a CI job triggers the building of a Docker container for each new combination of packages, on publication of new Galaxy tool versions. Another CI job is responsible for generating Singularity containers from the new

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Docker containers, which are made available by the BioContainers and Galaxy communities via a CernVM file system (CVMFS) [27]. These steps do not require manual review.

The Planemo autoupdate bot also monitors the Galaxy workflows maintained by the IWC and checks whether new versions exist for each of the component tools. Once a new tool version is created (either by the upstream tool autoupdate step, or a tool developer), the workflow definition file hosted by the IWC is modified accordingly and a PR submitted for review (Fig. 3).

Execution

Apart from providing assistance with tool and workflow development and deployment, Planemo is also a useful resource for Galaxy power users who need to launch high-throughput data analyses. Galaxy is traditionally accessed via a graphical interface in the web browser, and features such as Galaxy collections already provide a high level of parallelization to users of the graphical interface. Nonetheless, there are important scenarios in which a user might need to run individual workflows hundreds or thousands of times, in which the data cannot be grouped into collections ahead of time—for example, for variant calling of SARS-CoV-2 genomic data, in which a huge amount of new data is published continuously [28]. As a convenient alternative to the graphical interface, Planemo allows workflow execution to be scheduled programmatically using the `` `run` `` subcommand, either on a local machine or a larger Galaxy server. `` `planemo run` `` can be embedded in scripts of varying complexity, which can be scheduled and controlled via CI systems or message queues to run workflows on demand - such as on new data appearing or tool updates.

Internally, Planemo executes workflows by submitting them to the chosen server via Galaxy's API. Requests to the API are made using BioBlend, a library which wraps many API endpoints as Python methods. It is also possible to execute workflows directly using BioBlend, or simply by making API calls using a tool such as cURL. While this approach does offer a high level of flexibility, it requires the user to possess a high level of knowledge of the API (for example, the correct format to submit workflow parameters) and often requires the creation of custom scripts. By contrast, Planemo's `run` subcommand offers a high-level interface to execute workflows, monitor them during execution, and report on their status after completion, packaged as a single command.

For tool and workflow development, the artifacts under development are generally tested against an ephemeral local Galaxy instance, which is deleted after use. While this is also

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

supported by the `run` subcommand, with the workflow outputs saved to a specified location, this approach is not scalable for workflows which demand long compute times, with large data inputs, or with workflows which need to be executed multiple times. In many cases, the user may prefer to make use of established, stable infrastructures, such as a public Galaxy instance or a private instance administered by their research group. Planemo allows external Galaxy instances to be specified for all `run` and `test` commands by providing the server URL and user API authentication key on the command line. As it is inconvenient and insecure to enter the API key with each command, Planemo also allows users to define profiles, in which the URL and API key is configured for each server. The user can then define multiple profiles and run workflows on different servers simply by appending, e.g. `--profile usegalaxy-org` or `--profile private-server` to the command.

Planemo provides numerous command line options to configure the workflow execution process. The name of the history in which the new invocation is created, as well as a list of Galaxy tags to add, can be specified via the command line. In addition, Planemo and Galaxy allow both datasets and workflows to be specified via hexadecimal IDs which point towards a Galaxy object on an external server, rather than by referring to a local path. This has the advantage of avoiding multiple uploads of the same dataset or workflow, if the workflow has to be executed multiple times. Planemo can also be configured to either wait until the workflow has completed, and download the output datasets created, or to terminate once the workflow has been successfully scheduled. In the latter case, the `list_invocations` command can be used to monitor running workflows and to return the number of jobs which have succeeded, failed, or incomplete. If jobs have failed—for example, due to transient server issues— the user can also choose to restart them using the `rerun` subcommand.

Training material

Planemo provides utilities for developing tutorials for different types of data analysis with Galaxy. The Galaxy Training Network, accessible via https://training.galaxyproject.org, provides a range of training material including slide decks, tutorials and videos. In particular, the tutorials are written in Markdown and rendered using Jekyll, and often feature ‘hands-on boxes’ which describe the exact combination of parameters and input which users need to submit when running a Galaxy tool. Most tutorials instruct the trainees to run several Galaxy tools in sequence, and thus correspond to a Galaxy workflow.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Planemo provides two subcommands, `` `training_init` `` and `` `training_generate_from_wf` ``, which generate a directory structure for a new tutorial, containing skeleton Markdown files defining the tutorials. These files already contain sections and hands-on boxes for each tool, with the tool inputs and parameters predefined, ensuring a high level of consistency in the appearance and quality of the tutorials produced. The training developer can then take these templates and expand them with additional information, questions, diagrams and citations to produce the completed training. They also need to provide input datasets, which are usually stored on Zenodo. To populate a Galaxy server with these datasets, the training developer should also provide a data library file, which can be generated using the `` `training_fill_data_library` `` subcommand, including the Zenodo links and file formats of the datasets.

A major aim of the Galaxy Training Network project is improving accessibility for new contributors, including for scientists who are not comfortable with command-line software. As a result, the Planemo functionality relating to training material development is provided in webserver form as the Planemo Training Development Kit (PTDK). The application is written using Flask and deployed with Heroku; it can be accessed via https://ptdk.herokuapp.com. The interface allows the selection of the same options as the Planemo commands, with the additional option of specifying a workflow for generating the training using its ID from one of the major public Galaxy servers.

Conclusion

We have presented Planemo, a library and application which has already achieved widespread usage among Galaxy tool, workflow and training material developers, Galaxy power users, and as part of numerous automated deployment solutions. Planemo provides the developers of command-line software with an easy way to create a graphical interface, taking advantage of the many features developed by the Galaxy community and the compute resources provided by public Galaxy instances. We have described the complex infrastructure the Galaxy community has developed for creating and interacting with artifacts such as tools, workflows and training material. Planemo plays the crucial role of bridging the gaps between the human and automated components of this infrastructure, freeing members of the community to devote their time to developing, reviewing and performing novel scientific analyses.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Acknowledgements

The authors are grateful to the broader Galaxy community for their support and software development efforts. This work is funded by NIH Grants U41 HG006620 and NSF ABI Grant 1661497. Usegalaxy.eu is supported by the German Federal Ministry of Education and Research grants 031L0101C and de.NBI-epi to BG. Usegalaxy.org.au is supported by Bioplatforms Australia and the Australian Research Data Commons through funding from the Australian Government National Collaborative Research Infrastructure Strategy. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Tables

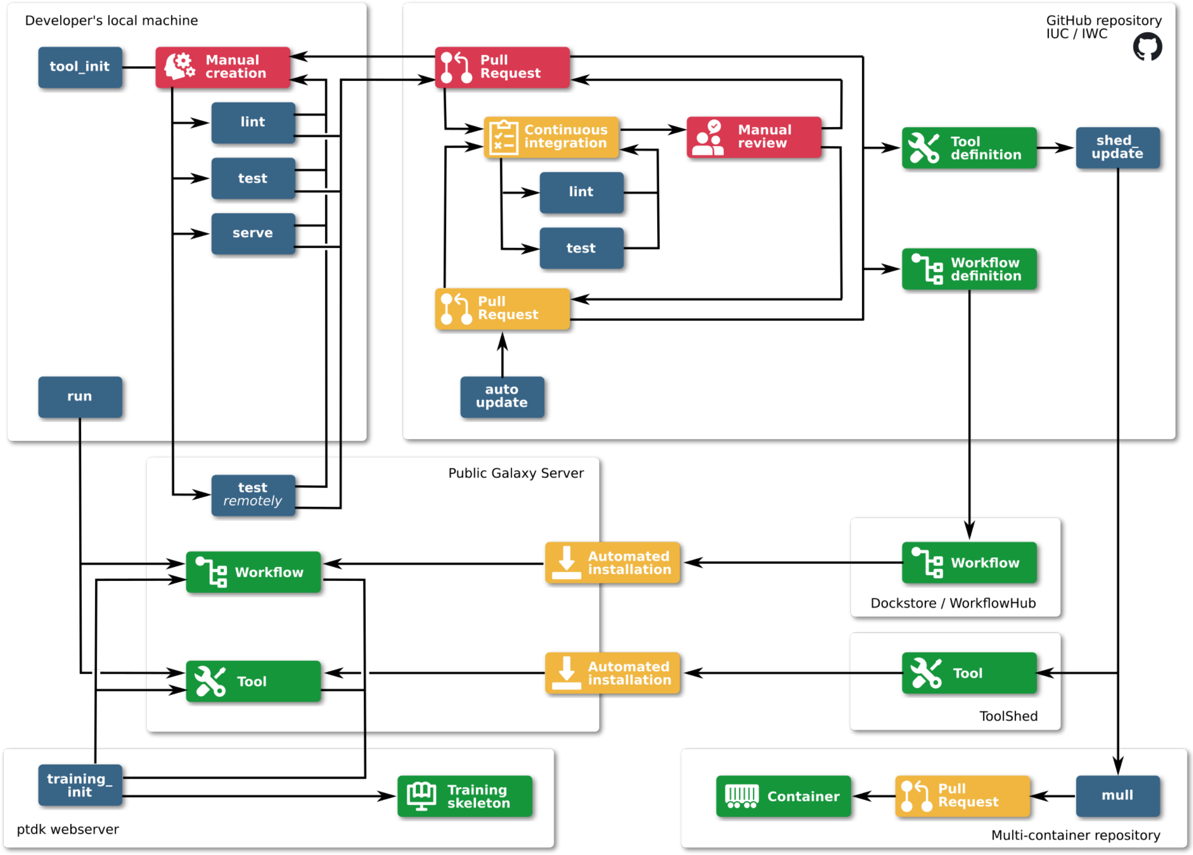
Object → **Function ↓**	**Galaxy tool**	**Galaxy workflow**	**CWL**	**Galaxy training material**
Initial template creation	`tool_init`	`workflow_test_init`	`tool_init`	`training_init`, `training_generate_from_wf`
Development	`test`, `lint`, `serve`	`test`, `lint`, `serve`	`test`, `lint`	
Deployment	`test`, `ci_*`, `shed_*`	`test`, `ci_*`, `shed_*`	-	*GTN*
Execution	`run`	`run`	`run`	*GTN*
Automated updates	`autoupdate`	`autoupdate`		-

Table 1. Overview of Planemo functionality and subcommands. Columns represent artifacts that can be created or manipulated with Planemo, rows represent different actions that can be performed on them. Italics represent actions which are performed without using Planemo: trainings are deployed using Jekyll and executed by users following the training material in the graphical interface.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

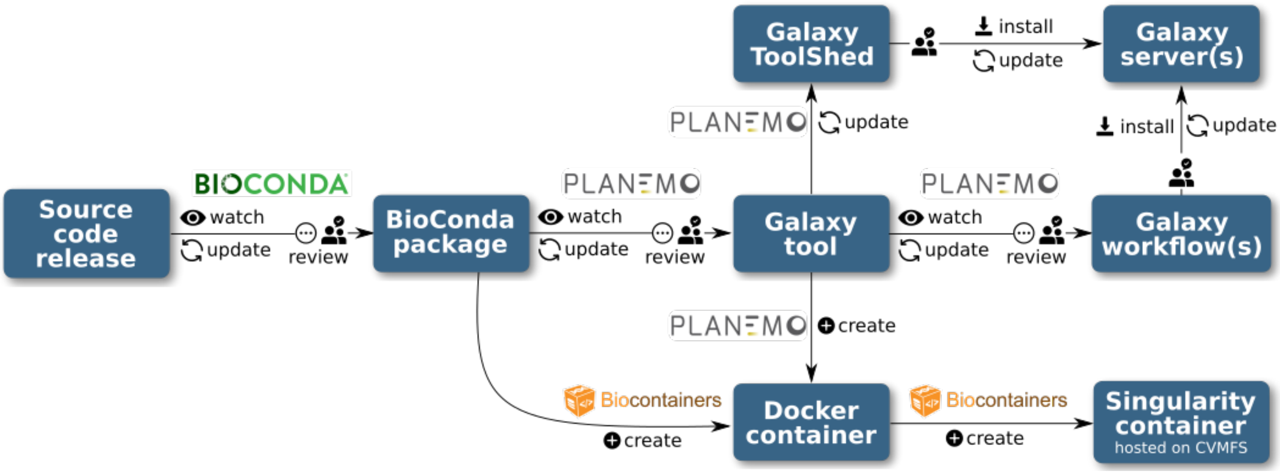
Figures

Figure 1. Overview of the use of Planemo for development, deployment, and execution of Galaxy tools, workflows and training materials. Red = manual work, blue = Planemo commands, yellow = automated steps, green = created artifacts.



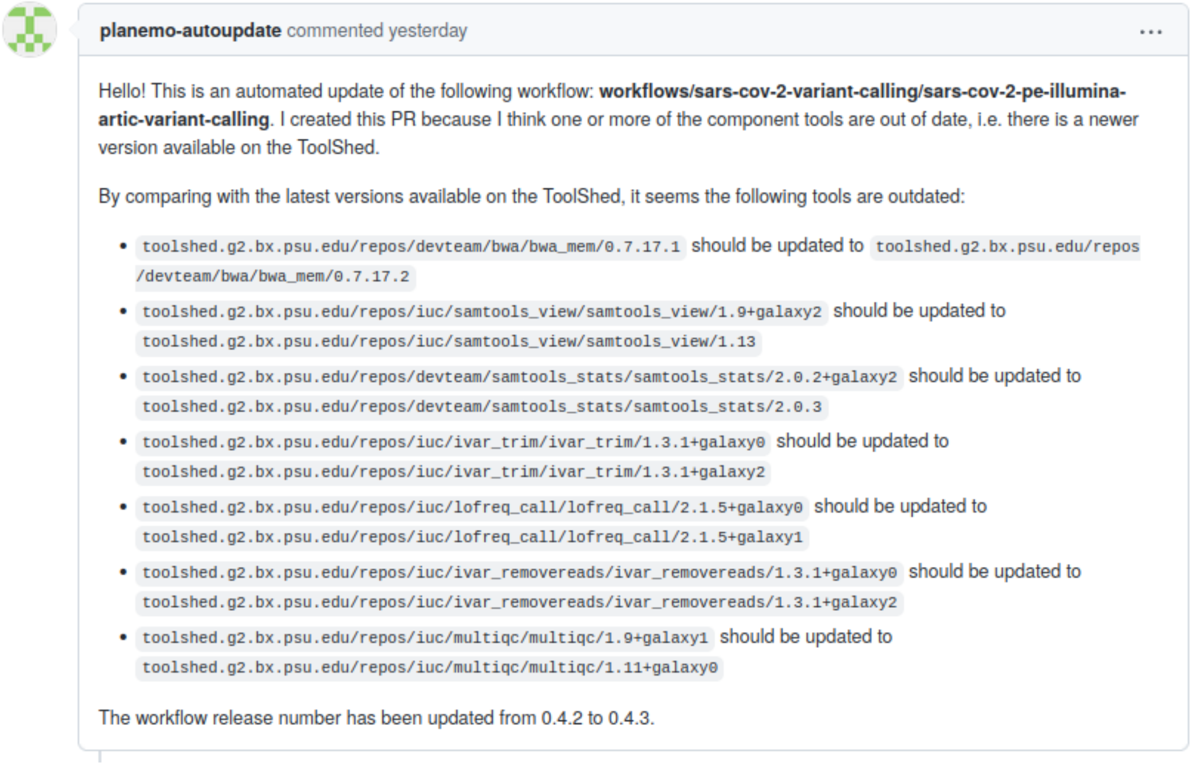
bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Figure 2. Automation pipeline for Bioconda packages, BioContainers, Galaxy tools and workflows. Steps marked in red require human review; steps marked in blue are fully automated.



bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Figure 3. An example GitHub pull request created by the Planemo autoupdate bot, updating a workflow hosted on the IWC.



planemo-autoupdate commented yesterday

Hello! This is an automated update of the following workflow: **workflows/sars-cov-2-variant-calling/sars-cov-2-pe-illumina-artic-variant-calling**. I created this PR because I think one or more of the component tools are out of date, i.e. there is a newer version available on the ToolShed.

By comparing with the latest versions available on the ToolShed, it seems the following tools are outdated:

- `toolshed.g2.bx.psu.edu/repos/devteam/bwa/bwa_mem/0.7.17.1` should be updated to `toolshed.g2.bx.psu.edu/repos/devteam/bwa/bwa_mem/0.7.17.2`
- `toolshed.g2.bx.psu.edu/repos/iuc/samtools_view/samtools_view/1.9+galaxy2` should be updated to `toolshed.g2.bx.psu.edu/repos/iuc/samtools_view/samtools_view/1.13`
- `toolshed.g2.bx.psu.edu/repos/devteam/samtools_stats/samtools_stats/2.0.2+galaxy2` should be updated to `toolshed.g2.bx.psu.edu/repos/devteam/samtools_stats/samtools_stats/2.0.3`
- `toolshed.g2.bx.psu.edu/repos/iuc/ivar_trim/ivar_trim/1.3.1+galaxy0` should be updated to `toolshed.g2.bx.psu.edu/repos/iuc/ivar_trim/ivar_trim/1.3.1+galaxy2`
- `toolshed.g2.bx.psu.edu/repos/iuc/lofreq_call/lofreq_call/2.1.5+galaxy0` should be updated to `toolshed.g2.bx.psu.edu/repos/iuc/lofreq_call/lofreq_call/2.1.5+galaxy1`
- `toolshed.g2.bx.psu.edu/repos/iuc/ivar_removereads/ivar_removereads/1.3.1+galaxy0` should be updated to `toolshed.g2.bx.psu.edu/repos/iuc/ivar_removereads/ivar_removereads/1.3.1+galaxy2`
- `toolshed.g2.bx.psu.edu/repos/iuc/multiqc/multiqc/1.9+galaxy1` should be updated to `toolshed.g2.bx.psu.edu/repos/iuc/multiqc/multiqc/1.11+galaxy0`

The workflow release number has been updated from 0.4.2 to 0.4.3.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

References

1. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Čech M, Chilton J, Clements D, Coraor N, Grüning BA, Guerler A, Hillman-Jackson J, Hiltemann S, Jalili V, Rasche H, Soranzo N, Goecks J, Taylor J, Nekrutenko A, Blankenberg D. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. Oxford Academic; 2018 May 22;46(W1):W537–W544.

2. Batut B, Hiltemann S, Bagnacani A, Baker D, Bhardwaj V, Blank C, Bretaudeau A, Brillet-Guéguen L, Čech M, Chilton J, Clements D, Doppelt-Azeroual O, Erxleben A, Freeberg MA, Gladman S, Hoogstrate Y, Hotz H-R, Houwaart T, Jagtap P, Larivière D, Le Corguillé G, Manke T, Mareuil F, Ramírez F, Ryan D, Sigloch FC, Soranzo N, Wolff J, Videm P, Wolfien M, Wubuli A, Yusuf D, Galaxy Training Network, Taylor J, Backofen R, Nekrutenko A, Grüning B. Community-Driven Data Analysis Training for Biology. Cell Syst. 2018 Jun 27;6(6):752–758.e1. PMCID: PMC6296361

3. Crusoe MR, Abeln S, Iosup A, Amstutz P, Chilton J, Tijanić N, Ménager H, Soiland-Reyes S, Gavrilovic B, Goble C. Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language. 2021 May 14 [cited 2022 Mar 11]; Available from: http://dx.doi.org/10.1145/3486897

4. Grüning B, Dale R, Sjödin A, Chapman BA, Rowe J, Tomkins-Tinch CH, Valieris R, Köster J, Bioconda Team. Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat Methods. 2018 Jul;15(7):475–476. PMID: 29967506

5. Common Workflow Language. GitHub - common-workflow-language/cwltool: Common Workflow Language reference implementation [Internet]. GitHub. [cited 2022 Mar 11]. Available from: https://github.com/common-workflow-language/cwltool

6. Vivian J, Rao AA, Nothaft FA, Ketchum C, Armstrong J, Novak A, Pfeil J, Narkizian J, Deran AD, Musselman-Brown A, Schmidt H, Amstutz P, Craft B, Goldman M, Rosenbloom K, Cline M, O'Connor B, Hanna M, Birger C, Kent WJ, Patterson DA, Joseph AD, Zhu J, Zaranek S, Getz G, Haussler D, Paten B. Toil enables reproducible, open source, big biomedical data analyses. Nat Biotechnol. Nature Publishing Group; 2017 Apr 11;35(4):314–316.

7. Sloggett C, Goonasekera N, Afgan E. BioBlend: automating pipeline analyses within Galaxy and CloudMan. Bioinformatics. Oxford Academic; 2013 Apr 28;29(13):1685–1686.

8. Intergalactic Utilities Commission [Internet]. [cited 2022 Mar 11]. Available from: https://galaxyproject.org/iuc/

9. Blank C, Easterly C, Gruening B, Johnson J, Kolmeder CA, Kumar P, May D, Mehta S, Mesuere B, Brown Z, Elias JE, Hervey WJ, McGowan T, Muth T, Nunn BL, Rudney J, Tanca A, Griffin TJ, Jagtap PD. Disseminating Metaproteomic Informatics Capabilities and Knowledge Using the Galaxy-P Framework. Proteomes. Multidisciplinary Digital Publishing Institute; 2018 Jan 31;6(1):7.

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

10. Conda — conda 4.12.0.post4+8c8af5e3 documentation [Internet]. [cited 2022 Mar 11]. Available from: https://docs.conda.io/projects/conda/en/latest/index.html

11. Empowering App Development for Developers [Internet]. Docker. [cited 2022 Mar 11]. Available from: https://www.docker.com/

12. Kurtzer GM, Sochat V, Bauer MW. Singularity: Scientific containers for mobility of compute. PLoS One. Public Library of Science; 2017 May 11;12(5):e0177459.

13. Blankenberg D, Von Kuster G, Bouvier E, Baker D, Afgan E, Stoler N, Team G, Taylor J, Nekrutenko A. Dissemination of scientific software with Galaxy ToolShed. Genome Biol. BioMed Central Ltd; 2014 Jan 1;15(2):403. PMCID: PMC4038738

14. Galaxy Project. GitHub - galaxyproject/iwc: Intergalactic Workflow Commission [Internet]. GitHub. [cited 2022 Mar 11]. Available from: https://github.com/galaxyproject/iwc

15. Yuen D, Cabansay L, Duncan A, Luu G, Hogue G, Overbeck C, Perez N, Shands W, Steinberg D, Reid C, Olunwa N, Hansen R, Sheets E, O'Farrell A, Cullion K, O'Connor BD, Paten B, Stein L. The Dockstore: enhancing a community platform for sharing reproducible and accessible computational protocols. Nucleic Acids Res. Oxford Academic; 2021 May 12;49(W1):W624–W632.

16. Goble C, Soiland-Reyes S, Bacall F, Owen S, Williams A, Eguinoa I, Droesbeke B, Leo S, Pireddu L, Rodríguez-Navas L, Fernández JM, Capella-Gutierrez S, Ménager H, Grüning B, Serrano-Solano B, Ewels P, Coppens F. Implementing FAIR Digital Objects in the EOSC-Life Workflow Collaboratory. 2021 Mar 12 [cited 2022 Mar 11]; Available from: https://zenodo.org/record/4605654

17. Galaxy Tool XML File — Galaxy Project 22.05.dev0 documentation [Internet]. [cited 2022 Mar 11]. Available from: https://docs.galaxyproject.org/en/latest/dev/schema.html

18. Galaxy Intergalactic Utilities Commission Standards and Best Practices — Galaxy IUC Standards and Best Practices 0.1 documentation [Internet]. [cited 2022 Mar 11]. Available from: https://galaxy-iuc-standards.readthedocs.io/

19. Galaxy Project. GitHub - galaxyproject/galaxy-language-server: Galaxy Language Server to help in Galaxy (https://galaxyproject.org/) tool wrappers development [Internet]. GitHub. [cited 2022 Mar 11]. Available from: https://github.com/galaxyproject/galaxy-language-server

20. Language Server Protocol [Internet]. Available from: https://microsoft.github.io/language-server-protocol/

21. Siddiqui S. Learning Test-Driven Development: A Polyglot Guide to Writing Uncluttered Code. O'Reilly Media; 2021.

22. Arvados [Internet]. Arvados. [cited 2022 Mar 11]. Available from: https://arvados.org/

23. gxformat2 [Internet]. Available from: https://github.com/galaxyproject/gxformat2

24. Galaxy Project. GitHub - galaxyproject/planemo-ci-action: Test, deploy, or lint changed

bioRxiv preprint doi: https://doi.org/10.1101/2022.03.13.483965; this version posted March 14, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Galaxy tools or workflows using Planemo [Internet]. GitHub. [cited 2022 Mar 11]. Available from: https://github.com/galaxyproject/planemo-ci-action

25. Galaxy tool repository template [Internet]. Available from: https://github.com/galaxyproject/galaxy-tool-repository-template

26. conda-forge community. The conda-forge Project: Community-based Software Distribution Built on the conda Package Format and Ecosystem. 2015 Jul 12 [cited 2022 Mar 11]; Available from: https://zenodo.org/record/4774217

27. Switzerland JBC, Switzerland PBP-S, Thomas Fuhrmann Technische Universität München, München, Germany. CernVM-FS [Internet]. ACM Conferences. [cited 2022 Mar 11]. Available from: https://dl.acm.org/doi/abs/10.1145/2110217.2110225

28. Maier W, Bray S, van den Beek M, Bouvier D, Coraor N, Miladi M, Singh B, De Argila JR, Baker D, Roach N, Gladman S, Coppens F, Martin DP, Lonie A, Grüning B, Kosakovsky Pond SL, Nekrutenko A. Ready-to-use public infrastructure for global SARS-CoV-2 monitoring. Nat Biotechnol. Nature Publishing Group; 2021 Sep 29;39(10):1178–1179.

Bibliography

[1] Jeremy Berg, John Tymoczko, and Lubert Stryer. *Biochemistry*. San Francisco, USA: W.H. Freeman, 2002 (cit. on pp. 1, 7, 9).

[2] Regine S Bohacek, Colin McMartin, and Wayne C Guida. "The art and practice of structure-based drug design: a molecular modeling perspective". In: *Medicinal Research Reviews* 16.1 (1996), pp. 3–50 (cit. on p. 1).

[3] Christopher W. Murray and David C. Rees. "The rise of fragment-based drug discovery". In: *Nature Chemistry* 1.3 (June 2009), pp. 187–192 (cit. on pp. 1, 12).

[4] Richard J. Hall, Christopher W. Murray, and Marcel L. Verdonk. "The Fragment Network: A Chemistry Recommendation Engine Built Using a Graph Database". In: *Journal of Medicinal Chemistry* 60.14 (July 2017), pp. 6440–6450 (cit. on p. 1).

[5] Harrison Green, David R. Koes, and Jacob D. Durrant. "DeepFrag: a deep convolutional neural network for fragment-based lead optimisation". In: *Chemical Science* 12.23 (2021), pp. 8036–8047 (cit. on p. 2).

[6] Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings 1PII of original article: S0169-409X(96)00423-1. The article was originally published in Advanced Drug Delivery Reviews 23 (1997) 3–25. 1". In: *Advanced Drug Delivery Reviews* 46.1-3 (Mar. 2001), pp. 3–26 (cit. on p. 2).

[7] Herman J. C. Berendsen. *Simulating the Physical World: Hierarchical Modeling from Quantum Mechanics to Fluid Dynamics*. Cambridge, United Kingdom: Cambridge University Press, 2007 (cit. on pp. 2, 19).

[8] Samuel Genheden and Ulf Ryde. "The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities". In: *Expert opinion on drug discovery* 10.5 (2015), pp. 449–461 (cit. on pp. 2, 20).

[9] Laura Wratten, Andreas Wilm, and Jonathan Göke. "Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers". In: *Nature Methods* 18.10 (Sept. 2021), pp. 1161–1168 (cit. on p. 3).

[10] J Harry Moore, Matthias R Bauer, Jeff Guo, et al. "Icolos: a workflow manager for structure-based post-processing of ide novo/i generated small molecules". In: *Bioinformatics* (Sept. 2022). Ed. by Alfonso Valencia (cit. on p. 3).

[11] Monya Baker. "1, 500 scientists lift the lid on reproducibility". In: *Nature* 533.7604 (May 2016), pp. 452–454 (cit. on p. 3).

[12]L. Polgár. “The catalytic triad of serine peptidases”. In: *Cellular and Molecular Life Sciences* 62.19-20 (July 2005), pp. 2161–2172 (cit. on p. 10).

[13]Christopher J. Oldfield and A. Keith Dunker. “Intrinsically Disordered Proteins and Intrinsically Disordered Protein Regions”. In: *Annual Review of Biochemistry* 83.1 (June 2014), pp. 553–584 (cit. on p. 10).

[14]Matthew K .Higgins and Susan M Lea. “On the state of crystallography at the dawn of the electron microscopy revolution”. In: *Current Opinion in Structural Biology* 46 (Oct. 2017), pp. 95–101 (cit. on p. 10).

[15]David Eliezer. “Biophysical characterisation of intrinsically disordered proteins”. In: *Current Opinion in Structural Biology* 19.1 (2009). Folding and binding / Protein-nuclei acid interactions, pp. 23–30 (cit. on p. 10).

[16]Editorial. “So Much More to Know”. In: *Science* 309.5731 (July 2005), pp. 78–102 (cit. on p. 11).

[17]John Jumper, Richard Evans, Alexander Pritzel, et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (July 2021), pp. 583–589 (cit. on p. 11).

[18]Viktor Hornak, Robert Abel, Asim Okur, et al. “Comparison of multiple Amber force fields and development of improved protein backbone parameters”. In: *Proteins: Structure, Function, and Bioinformatics* 65.3 (2006), pp. 712–725 (cit. on pp. 11, 65).

[19]Lutea A.A. de Jong, Donald R.A. Uges, Jan Piet Franke, and Rainer Bischoff. “Receptor–ligand binding assays: Technologies and Applications”. In: *Journal of Chromatography B* 829.1 (2005), pp. 1–25 (cit. on p. 11).

[20]Priyabrata Pattnaik. “Surface plasmon resonance”. In: *Applied Biochemistry and Biotechnology* 126 (2005), pp. 79–92 (cit. on p. 11).

[21]Katarzyna Smietana, Marcin Siatkowski, and Martin Møller. “Trends in clinical success rates”. In: *Nature Reviews Drug Discovery* 15.6 (2016), pp. 379–80 (cit. on p. 12).

[22]Magdalena Bacilieri and Stefano Moro. “Ligand-Based Drug Design Methodologies in Drug Discovery Process: An Overview”. In: *Current Drug Discovery Technologies* 3.3 (Sept. 2006), pp. 155–165 (cit. on p. 12).

[23]Paweł Śledź and Amedeo Caflisch. “Protein structure-based drug design: from docking to molecular dynamics”. In: *Current Opinion in Structural Biology* 48 (2018). Folding and binding in silico, in vitro and in cellula • Proteins: An Evolutionary Perspective, pp. 93–102 (cit. on p. 12).

[24]Robert A. Copeland. “The drug–target residence time model: a 10-year retrospective.” In: *Nature Reviews Drug Discovery* 15.2 (2015), pp. 87–95 (cit. on p. 15).

[25]Andrew Dalke. “The FPS fingerprint format and chemfp toolkit”. In: *Journal of Cheminformatics* 5.1 (2013), pp. 1–1 (cit. on p. 16).

[26]Christopher M Dobson et al. “Chemical space and biology”. In: *Nature* 432.7019 (2004), pp. 824–828 (cit. on p. 17).

[27] Petra Schneider, W Patrick Walters, Alleyn T Plowright, et al. “Rethinking drug design in the artificial intelligence era”. In: *Nature Reviews Drug Discovery* 19.5 (2020), pp. 353–364 (cit. on p. 17).

[28] Artem Cherkasov, Eugene N Muratov, Denis Fourches, et al. “QSAR modeling: where have you been? Where are you going to?” In: *Journal of Medicinal Chemistry* 57.12 (2014), pp. 4977–5010 (cit. on p. 17).

[29] Eugene N. Muratov, Jürgen Bajorath, Robert P. Sheridan, et al. “QSAR without borders”. In: *Chemical Society Reviews* 49 (11 2020), pp. 3525–3564 (cit. on p. 17).

[30] Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. “Molecular representations in AI-driven drug discovery: a review and practical guide”. In: *Journal of Cheminformatics* 12.1 (Sept. 2020) (cit. on p. 17).

[31] Noel M O’Boyle, Michael Banck, Craig A James, et al. “Open Babel: An open chemical toolbox”. In: *Journal of Cheminformatics* 3.1 (2011), pp. 1–14 (cit. on p. 17).

[32] Patrick J. Ropp, Jacob O. Spiegel, Jennifer L. Walker, et al. “Gypsum-DL: an open-source program for preparing small-molecule libraries for structure-based virtual screening”. In: *Journal of Cheminformatics* 11.1 (May 2019) (cit. on p. 17).

[33] Patrick J. Ropp, Jesse C. Kaminsky, Sara Yablonski, and Jacob D. Durrant. “Dimorphite-DL: an open-source program for enumerating the ionisation states of drug-like small molecules”. In: *Journal of Cheminformatics* 11.1 (Feb. 2019) (cit. on p. 17).

[34] rDock Development Team. *rDock Reference Guide*. `http://rdock.sourceforge.net/wp-content/uploads/2015/08/rDock_User_Guide.pdf`. Online; accessed 25 April 2022. 2015 (cit. on p. 18).

[35] Daniel Cappel, Steven Jerome, Gerhard Hessler, and Hans Matter. “Impact of Different Automated Binding Pose Generation Approaches on Relative Binding Free Energy Simulations”. In: *Journal of Chemical Information and Modeling* 60.3 (Jan. 2020), pp. 1432–1444 (cit. on p. 18).

[36] Hugo Guterres and Wonpil Im. “Improving Protein-Ligand Docking Results with High-Throughput Molecular Dynamics Simulations”. In: *Journal of Chemical Information and Modeling* 60.4 (Mar. 2020), pp. 2189–2198 (cit. on p. 18).

[37] Andrew T. McNutt, Paul Francoeur, Rishal Aggarwal, et al. “GNINA 1.0: molecular docking with deep learning”. In: *Journal of Cheminformatics* 13.1 (June 2021) (cit. on p. 18).

[38] Jack Scantlebury, Nathan Brown, Frank Von Delft, and Charlotte M. Deane. “Data Set Augmentation Allows Deep Learning-Based Virtual Screening to Better Generalise to Unseen Target Classes and Highlight Important Binding Interactions”. In: *Journal of Chemical Information and Modeling* 60.8 (July 2020), pp. 3722–3730 (cit. on pp. 18, 87).

[39] Douglas L. Strout and Gustavo E. Scuseria. “A quantitative study of the scaling properties of the Hartree–Fock method.” In: *Journal of Chemical Physics* 102.21 (1995), pp. 8448–8452. eprint: `https://doi.org/10.1063/1.468836` (cit. on p. 19).

[40] Roland R. Netz and William A. Eaton. "Estimating computational limits on theoretical descriptions of biological cells". In: *Proceedings of the National Academy of Sciences* 118.6 (Jan. 2021) (cit. on p. 19).

[41] Ron O. Dror, Robert M. Dirks, J.P. Grossman, Huafeng Xu, and David E. Shaw. "Biomolecular Simulation: A Computational Microscope for Molecular Biology." In: *Annual Review of Biophysics* 41.1 (2012), pp. 429–452 (cit. on p. 19).

[42] Michael P. Allen. "Introduction to molecular dynamics simulation," in: *Computational soft matter: from synthetic polymers to proteins*. Ed. by N. Attig, K. Binder, H. Grubmüller, and K. Kremer. Jülich: John von Neumann Institute for Computing, 2004, pp. 1–27 (cit. on p. 19).

[43] Laura Orellana. "Large-Scale Conformational Changes and Protein Function: Breaking the in silico Barrier". In: *Frontiers in Molecular Biosciences* 6 (Nov. 2019) (cit. on p. 20).

[44] David E. Shaw, Ron O. Dror, John K. Salmon, et al. "Millisecond-Scale Molecular Dynamics Simulations on Anton". In: *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. SC '09. Portland, Oregon: Association for Computing Machinery, 2009 (cit. on p. 20).

[45] Vincent A Voelz, Gregory R Bowman, Kyle Beauchamp, and Vijay S Pande. "Molecular simulation of ab initio protein folding for a millisecond folder NTL9 (1- 39)". In: *Journal of the American Chemical Society* 132.5 (2010), pp. 1526–1528 (cit. on p. 20).

[46] Outi M. H. Salo-Ahen, Ida Alanko, Rajendra Bhadane, et al. "Molecular Dynamics Simulations in Drug Discovery and Pharmaceutical Development". In: *Processes* 9.1 (2021) (cit. on p. 20).

[47] Alessandro Laio and Michele Parrinello. "Escaping free-energy minima". In: *Proceedings of the National Academy of Sciences* 99.20 (2002), pp. 12562–12566 (cit. on p. 20).

[48] Susanna K Lüdemann, Oliviero Carugo, and Rebecca C Wade. "Substrate access to cytochrome P450cam: A comparison of a thermal motion pathway analysis with molecular dynamics simulation data". In: *Molecular Modeling Annual* 3.8 (1997), pp. 369–374 (cit. on p. 20).

[49] Luca Mollica, Sergio Decherchi, Syeda Rehana Zia, et al. "Kinetics of protein-ligand unbinding via smoothed potential molecular dynamics simulations". In: *Scientific Reports* 5.1 (2015), pp. 1–12 (cit. on p. 20).

[50] Alan E Mark, Wilfred F van Gunsteren, and Herman JC Berendsen. "Calculation of relative free energy via indirect pathways". In: *Journal of Chemical Physics* 94.5 (1991), pp. 3808–3816 (cit. on p. 20).

[51] Sergei Izrailev, Sergey Stepaniants, Manel Balsera, Yoshi Oono, and Klaus Schulten. "Molecular dynamics study of unbinding of the avidin-biotin complex". In: *Biophysical Journal* 72.4 (1997), pp. 1568–1581 (cit. on p. 20).

[52] Sergio Ruiz-Carmona, Peter Schmidtke, F Javier Luque, et al. "Dynamic undocking and the quasi-bound state as tools for drug discovery". In: *Nature Chemistry* 9.3 (2017), pp. 201–206 (cit. on p. 21).

[53] Steffen Wolf and Gerhard Stock. “Targeted molecular dynamics calculations of free energy profiles using a nonequilibrium friction correction”. In: *Journal of Chemical Theory and Computation* 14.12 (2018), pp. 6175–6182 (cit. on p. 21).

[54] J Schlitter, M Engels, and P Krüger. “Targeted molecular dynamics: a new approach for searching pathways of conformational transitions”. In: *Journal of Molecular Graphics* 12.2 (1994), pp. 84–89 (cit. on p. 21).

[55] Giuseppe Deganutti, Stefano Moro, and Christopher A. Reynolds. “A Supervised Molecular Dynamics Approach to Unbiased Ligand–Protein Unbinding”. In: *Journal of Chemical Information and Modeling* 60.3 (Mar. 2020), pp. 1804–1817 (cit. on p. 21).

[56] The Galaxy Community. “The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update”. In: *Nucleic Acids Research* (Apr. 2022). gkac247. eprint: `https://academic.oup.com/nar/advance-article-pdf/doi/10.1093/nar/gkac247/43406600/gkac247.pdf` (cit. on p. 22).

[57] Björn Grüning, Ryan Dale, Andreas Sjödin, et al. “Bioconda: sustainable and comprehensive software distribution for the life sciences”. In: *Nature Methods* 15.7 (2018), pp. 475–476 (cit. on pp. 22, 23).

[58] conda-forge community. *The conda-forge Project: Community-based Software Distribution Built on the conda Package Format and Ecosystem*. July 2015 (cit. on pp. 22, 23).

[59] Gregory M Kurtzer, Vanessa Sochat, and Michael W Bauer. “Singularity: Scientific containers for mobility of compute”. In: *PLOS One* 12.5 (2017), e0177459 (cit. on p. 22).

[60] Peter W. Hildebrand, Alexander S. Rose, and Johanna K.S. Tiemann. “Bringing Molecular Dynamics Simulation Data into View”. In: *Trends in Biochemical Sciences* 44.11 (Nov. 2019), pp. 902–913 (cit. on p. 22).

[61] Clare Sloggett, Nuwan Goonasekera, and Enis Afgan. “BioBlend: automating pipeline analyses within Galaxy and CloudMan”. In: *Bioinformatics* 29.13 (2013), pp. 1685–1686 (cit. on p. 23).

[62] Felipe da Veiga Leprevost, Björn A Grüning, Saulo Alves Aflitos, et al. “BioContainers: an open-source and community-driven framework for software standardisation”. In: *Bioinformatics* 33.16 (2017), pp. 2580–2582 (cit. on p. 23).

[63] Ola Spjuth, Marco Capuccini, Matteo Carone, et al. “Approaches for containerised scientific workflows in cloud environments with applications in life science”. In: *F1000Research* 10 (June 2021), p. 513 (cit. on p. 24).

[64] Jakob Blomer, Predrag Buncic, and Thomas Fuhrmann. “CernVM-FS: Delivering Scientific Software to Globally Distributed Computing Resources”. In: *Proceedings of the First International Workshop on Network-Aware Data Management*. NDM ’11. Seattle, Washington, USA: Association for Computing Machinery, 2011, 49–56 (cit. on p. 24).

[65] Jingwen Bai, Chakradhar Bandla, Jiaxin Guo, et al. “BioContainers Registry: Searching Bioinformatics and Proteomics Tools, Packages, and Containers”. In: *Journal of Proteome Research* 20.4 (Feb. 2021), pp. 2056–2061 (cit. on p. 24).

[66] Omnia developers. *Omnia MD conda recipes*. `https://github.com/omnia-md/conda-recipes`. Online; accessed 25 April 2022. 2021 (cit. on p. 26).

[67] Peter Eastman, Jason Swails, John D. Chodera, et al. “OpenMM 7: Rapid development of high performance algorithms for molecular dynamics”. In: *PLOS Computational Biology* 13.7 (July 2017). Ed. by Robert Gentleman, e1005659 (cit. on p. 26).

[68] Mark James Abraham, Teemu Murtola, Roland Schulz, et al. “GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers.” In: *SoftwareX* 1-2 (2015), pp. 19–25 (cit. on pp. 26, 28).

[69] Cyril Dominguez, Rolf Boelens, and Alexandre MJJ Bonvin. “HADDOCK: a protein-protein docking approach based on biochemical or biophysical information”. In: *Journal of the American Chemical Society* 125.7 (2003), pp. 1731–1737 (cit. on p. 26).

[70] Vytautas Gapsys, Servaas Michielssens, Daniel Seeliger, and Bert L. de Groot. “pmx: Automated protein structure and topology generation for alchemical perturbations”. In: *Journal of Computational Chemistry* 36.5 (2015), pp. 348–354 (cit. on p. 26).

[71] Pau Andrio, Adam Hospital, Javier Conejero, et al. “BioExcel Building Blocks, a software library for interoperable biomolecular simulation workflows”. In: *Scientific Data* 6.1 (2019), pp. 1–8 (cit. on p. 26).

[72] Michael R Crusoe, Sanne Abeln, Alexandru Iosup, et al. “Methods included: Standardizing computational reuse and portability with the common workflow language”. In: *arXiv preprint arXiv:2105.07028* (2021) (cit. on p. 26).

[73] Sunghwan Kim, Paul A Thiessen, Evan E Bolton, et al. “PubChem substance and compound databases”. In: *Nucleic Acids Research* 44.D1 (2015), pp. D1202–D1213 (cit. on p. 27).

[74] Anna Gaulton, Anne Hersey, Michał Nowotka, et al. “The ChEMBL database in 2017”. In: *Nucleic Acids Research* 45.D1 (2016), pp. D945–D954 (cit. on p. 27).

[75] Teague Sterling and John J Irwin. “ZINC 15–ligand discovery for everyone”. In: *Journal of Chemical Information and Modeling* 55.11 (2015), pp. 2324–2337 (cit. on p. 27).

[76] Qiang Gu, Anup Kumar, Simon Bray, et al. “Galaxy-ML: An accessible, reproducible, and scalable machine learning toolkit for biomedicine”. In: *PLoS Computational Biology* 17.6 (2021), e1009014 (cit. on p. 27).

[77] Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, et al. “CHARMM: A program for macromolecular energy, minimization, and dynamics calculations”. In: *Journal of Computational Chemistry* 4.2 (1983), pp. 187–217 (cit. on p. 28).

[78] Mark T Nelson, William Humphrey, Attila Gursoy, et al. “NAMD: a parallel, object-oriented molecular dynamics program”. In: *The International Journal of Supercomputer Applications and High Performance Computing* 10.4 (1996), pp. 251–268 (cit. on p. 28).

[79] Naveen Michaud-Agrawal, Elizabeth J. Denning, Thomas B. Woolf, and Oliver Beckstein. “MDAnalysis: A toolkit for the analysis of molecular dynamics simulations.” In: *Journal of Computational Chemistry* 32.10 (2011), pp. 2319–2327. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.21787` (cit. on pp. 28, 70).

[80] Barry J Grant, Ana PC Rodrigues, Karim M ElSawy, J Andrew McCammon, and Leo SD Caves. “Bio3d: an R package for the comparative analysis of protein structures”. In: *Bioinformatics* 22.21 (2006), pp. 2695–2696 (cit. on p. 28).

[81] GROMACS developers. *GROMACS command line reference*. `https://manual.gromacs.org/documentation/current/user-guide/cmdline.html`. Online; accessed 25 April 2022. 2022 (cit. on p. 28).

[82] Simon A. Bray, Xavier Lucas, Anup Kumar, and Björn A. Grüning. “The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform”. In: *Journal of Cheminformatics* 12.1 (June 2020) (cit. on p. 29).

[83] David S Wishart, Yannick D Feunang, An C Guo, et al. “DrugBank 5.0: a major update to the DrugBank database for 2018”. In: *Nucleic Acids Research* 46.D1 (2017), pp. D1074–D1082 (cit. on p. 29).

[84] Ying Hong Li, Chun Yan Yu, Xiao Xu Li, et al. “Therapeutic Target Database update 2018: enriched resource for facilitating bench-to-clinic research of targeted therapeutics”. In: *Nucleic Acids Research* 46.D1 (2017), pp. D1121–D1127 (cit. on p. 30).

[85] Darko Butina. “Unsupervised data base clustering based on daylight’s fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets”. In: *Journal of Chemical Information and Computer Sciences* 39.4 (1999), pp. 747–750 (cit. on p. 30).

[86] Jerome Eberhardt, Diogo Santos-Martins, Andreas F Tillack, and Stefano Forli. “AutoDock Vina 1.2. 0: New docking methods, expanded force field, and python bindings”. In: *Journal of Chemical Information and Modeling* 61.8 (2021), pp. 3891–3898 (cit. on p. 30).

[87] Sergio Ruiz-Carmona, Daniel Alvarez-Garcia, Nicolas Foloppe, et al. “rDock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids”. In: *PLOS Computational Biology* 10.4 (2014), e1003571 (cit. on p. 30).

[88] Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. “Fpocket: an open source platform for ligand pocket detection”. In: *BMC Bioinformatics* 10.1 (2009), pp. 1–11 (cit. on p. 30).

[89] Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. “Mordred: a molecular descriptor calculator”. In: *Journal of Cheminformatics* 10.1 (Feb. 2018) (cit. on p. 31).

[90] K Dechering, C Boersma, and S Mosselman. “Estrogen receptors alpha and beta: two receptors of a kind”. In: *Current Medicinal Chemistry* 7.5 (2000), pp. 561–576 (cit. on p. 31).

[91] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 31).

[92] Bérénice Batut, Saskia Hiltemann, Andrea Bagnacani, et al. “Community-driven data analysis training for biology”. In: *Cell Systems* 6.6 (2018), pp. 752–758 (cit. on p. 32).

[93] B.W. Matthews, S.J. Remington, M.G. Grütter, and W.F. Anderson. “Relation between hen egg white lysozyme and bacteriophage T4 lysozyme: Evolutionary implications”. In: *Journal of Molecular Biology* 147.4 (Apr. 1981), pp. 545–558 (cit. on p. 59).

[94] Susan Marqusee, Manuel Llinás, Blake Gillespie, and Frederick W. Dahlquist. In: *Nature Structural Biology* 6.11 (Nov. 1999), pp. 1072–1078 (cit. on p. 59).

[95] Brian W. Matthews. “Structural and genetic analysis of the folding and function of T4 lysozyme”. In: *The FASEB Journal* 10.1 (Jan. 1996), pp. 35–41 (cit. on p. 59).

[96] A. E. Eriksson, W. A. Baase, X.-J. Zhang, et al. “Response of a Protein Structure to Cavity-Creating Mutations and Its Relation to the Hydrophobic Effect”. In: *Science* 255.5041 (1992), pp. 178–183 (cit. on p. 59).

[97] A. E. Eriksson, W. A. Baase, J. A. Wozniak, and B. W. Matthews. “A cavity-containing mutant of T4 lysozyme is stabilised by buried benzene”. In: *Nature* 355.6358 (Jan. 1992), pp. 371–373 (cit. on p. 60).

[98] David L. Mobley and Michael K. Gilson. “Predicting Binding Free Energies: Frontiers and Benchmarks”. In: *Annual Review of Biophysics* 46.1 (May 2017), pp. 531–558 (cit. on p. 60).

[99] William Humphrey, Andrew Dalke, and Klaus Schulten. “VMD: Visual molecular dynamics”. In: *J. Mol. Graphics* 14.1 (Feb. 1996), pp. 33–38 (cit. on pp. 60, 66, 67, 78, 79).

[100] Riccardo Capelli, Paolo Carloni, and Michele Parrinello. “Exhaustive Search of Ligand Binding Pathways via Volume-Based Metadynamics”. In: *Journal of Physical Chemistry Letters* 10.12 (June 2019), pp. 3495–3499 (cit. on pp. 61, 62).

[101] Ariane Nunes-Alves, Daniel M. Zuckerman, and Guilherme Menegon Arantes. “Escape of a Small Molecule from Inside T4 Lysozyme by Multiple Pathways”. In: *Biophysical Journal* 114.5 (Mar. 2018), pp. 1058–1066 (cit. on p. 61).

[102] Ai Niitsu, Suyong Re, Hiraku Oshima, Motoshi Kamiya, and Yuji Sugita. “De Novo Prediction of Binders and Nonbinders for T4 Lysozyme by gREST Simulations”. In: *Journal of Chemical Information and Modeling* 59.9 (Aug. 2019), pp. 3879–3888 (cit. on p. 61).

[103] Jakub Rydzewski and Omar Valsson. “Finding multiple reaction pathways of ligand unbinding”. In: *Journal of Chemical Physics* 150.22 (June 2019), p. 221101 (cit. on p. 61).

[104] Ariane Nunes-Alves, Daria B Kokh, and Rebecca C Wade. “Recent progress in molecular simulation methods for drug binding kinetics”. In: *Current Opinion in Structural Biology* 64 (Oct. 2020), pp. 126–133 (cit. on p. 61).

[105] Ariane Nunes-Alves, Daria B. Kokh, and Rebecca C. Wade. “Recent progress in molecular simulation methods for drug binding kinetics”. In: (2020) (cit. on p. 61).

[106] William L Jorgensen, Jayaraman Chandrasekhar, Jeffry D Madura, Roger W Impey, and Michael L Klein. “Comparison of simple potential functions for simulating liquid water”. In: *Journal of Chemical Physics* 79.2 (1983), pp. 926–935 (cit. on p. 65).

[107] Alan W Sousa da Silva and Wim F Vranken. “ACPYPE-Antechamber python parser interface”. In: *BMC Research Notes* 5.1 (2012), pp. 1–8 (cit. on p. 65).

[108] D.A. Case et al. *AMBER 2018*. http://ambermd.org/index.php/. 2018 (cit. on p. 65).

[109] Junmei Wang, Wei Wang, Peter A. Kollman, and David A. Case. “Automatic atom type and bond type perception in molecular mechanical calculations.” In: *Journal of Molecular Graphics and Modelling* 25.2 (2006), pp. 247–260 (cit. on p. 65).

[110] Tom Darden, Darrin York, and Lee Pedersen. “Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems.” In: *Journal of Chemical Physics* 98.12 (1993), pp. 10089–10092 (cit. on p. 68).

[111] Berk Hess, Henk Bekker, Herman J. C. Berendsen, and Johannes G. E. M. Fraaije. “LINCS: A linear constraint solver for molecular simulations.” In: *Journal of Computational Chemistry* 18.12 (1998), pp. 1463–1472 (cit. on p. 68).

[112] Shuichi Nosé. “A unified formulation of the constant temperature molecular dynamics methods.” In: *Journal of Chemical Physics* 81.1 (1984), pp. 511–519 (cit. on p. 68).

[113] M. Parrinello and A. Rahman. “Polymorphic transitions in single crystals: A new molecular dynamics method.” In: *Journal of Applied Physics* 52.12 (1981), pp. 7182–7190 (cit. on p. 68).

[114] Matthias Ernst, Florian Sittel, and G. Stock. “Contact- and distance-based principal component analysis of protein dynamics”. In: *Journal of Chemical Physics* 143 (2015), p. 244114 (cit. on pp. 68, 70).

[115] P. H. A. Sneath and R. R. Sokal. *Numerical taxonomy. The principles and practice of numerical classification.* San Francisco, USA: W.H. Freeman, 1973 (cit. on p. 69).

[116] Travis J. Wheeler and John D. Kececioglu. “Multiple alignment by aligning alignments.” In: *Bioinformatics* 23.13 (2007), pp. i559–i568. eprint: /oup/backfile/content_public/journal/bioinformatics/23/13/10.1093_bioinformatics_btm226/1/btm226.pdf (cit. on p. 69).

[117] Daniel H. Huson and David Bryant. “Application of Phylogenetic Networks in Evolutionary Studies.” In: *Molecular Biology and Evolution* 23.2 (2006), pp. 254–267. eprint: /oup/backfile/content_public/journal/mbe/23/2/10.1093/molbev/msj030/2/msj030.pdf (cit. on p. 69).

[118] Daniel H. Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic Networks: Concepts, Algorithms and Applications.* Cambridge, United Kingdom: Cambridge University Press, 2010 (cit. on p. 69).

[119] Simon Bray, Victor Tanzel, and Steffen Wolf. "Ligand Unbinding Pathway and Mechanism Analysis Assisted by Machine Learning and Graph Methods". In: *Journal of Chemical Information and Modeling* (2022) (cit. on p. 69).

[120] Florian Sittel, Thomas Filk, and Gerhard Stock. "Principal component analysis on a torus: Theory and application to protein dynamics". In: *Journal of Chemical Physics* 147.24 (Dec. 2017), p. 244101 (cit. on p. 70).

[121] Steffen Wolf, Benjamin Lickert, Simon Bray, and Gerhard Stock. "Multisecond ligand dissociation dynamics from atomistic simulations". In: *Nature Communications* 11.1 (June 2020) (cit. on p. 70).

[122] Peter Hänggi, Peter Talkner, and Michal Borkovec. "Reaction-rate theory: fifty years after Kramers". In: *Reviews of modern physics* 62.2 (1990), p. 251 (cit. on p. 78).

[123] Victoria A Feher, Enoch P Baldwin, and Frederick W Dahlquist. "Access of ligands to cavities within the core of a protein is rapid". In: *Nature structural biology* 3.6 (1996), pp. 516–521 (cit. on p. 80).

[124] Na Zhu, Dingyu Zhang, Wenling Wang, et al. "A novel coronavirus from patients with pneumonia in China, 2019". In: *New England journal of medicine* (2020) (cit. on p. 83).

[125] COVID Moonshot contributors. *COVID Moonshot*. `https://postera.ai/moonshot`. Online; accessed 25 April 2022. 2022 (cit. on p. 83).

[126] Zhenming Jin, Xiaoyu Du, Yechun Xu, et al. "Structure of Mpro from SARS-CoV-2 and discovery of its inhibitors". In: *Nature* 582.7811 (2020), pp. 289–293 (cit. on p. 83).

[127] Alice Douangamath, Daren Fearon, Paul Gehrtz, et al. "Crystallographic and electrophilic fragment screening of the SARS-CoV-2 main protease". In: *Nature communications* 11.1 (2020), pp. 1–11 (cit. on p. 84).

[128] Dannon Baker, Marius van den Beek, Daniel Blankenberg, et al. "No more business as usual: Agile and effective responses to emerging pathogen threats require open data and open analytics". In: *PLOS Pathogens* 16.8 (Aug. 2020), pp. 1–8 (cit. on p. 84).

[129] Simon Bray, Tim Dudgeon, Rachael Skyner, et al. "Galaxy workflows for fragment-based virtual screening: a case study on the SARS-CoV-2 main protease". In: *Journal of Cheminformatics* 14.1 (Apr. 2022) (cit. on p. 84).

[130] Fragalysis developers. *Fragalysis, https://diamondlightsource.atlassian.net/wiki/spaces/FRAG/overview*. `https://diamondlightsource.atlassian.net/wiki/spaces/FRAG/overview`. 2022 (cit. on p. 86).

[131] Susan Leung, Michael Bodkin, Frank von Delft, Paul Brennan, and Garrett Morris. "SuCOS is Better than RMSD for Evaluating Fragment Elaboration and Docking Poses". In: (May 2019) (cit. on p. 87).

[132] Paolo Di Tommaso, Maria Chatzou, Evan W Floden, et al. "Nextflow enables reproducible computational workflows". In: *Nature biotechnology* 35.4 (2017), pp. 316–319 (cit. on p. 105).

[133] Johannes Köster and Sven Rahmann. “Snakemake—a scalable bioinformatics workflow engine”. In: *Bioinformatics* 28.19 (2012), pp. 2520–2522 (cit. on p. 105).

[134] Bioconda developers. *BiocondaBot*. `https://github.com/BiocondaBot`. Online; accessed 25 April 2022. 2021 (cit. on p. 109).

List of Figures

List of Tables

Declaration

I declare that this thesis is an original report of my research, has been written by me and has not been submitted for any previous degree. Collaborative contributions have been indicated clearly and acknowledged.

Freiburg im Breisgau, September 25, 2023

Simon Bray