CHAPTER 1

Computational Chemistry and Molecular Modelling Basics

SAMUEL GENHEDEN,[a] ANNA REYMER,[a]
PATRICIA SAENZ-MÉNDEZ[a,b] AND LEIF A. ERIKSSON*[a]

[a] Department of Chemistry and Molecular Biology, University of Gothenburg, 405 30 Göteborg, Sweden; [b] Computational Chemistry and Biology Group, Facultad de Química, UdelaR, 11800 Montevideo, Uruguay
*Email: leif.eriksson@chem.gu.se

1.1 Introduction

The use of computers for predicting the structures and properties of biomolecules has closely paralleled computer development since the 1950s, and has been one of the core areas of theoretical or computational chemistry for the past 30 years. Initially, the focus was on force-field based methodologies for studying the structures, dynamics and interactions of biomolecules as such, and the development of accurate models for the main biological solvent, water. With the emergence of accurate quantum chemical techniques suitable for studying (from a quantum chemistry perspective) large systems, density functional theory entered the stage in the 1990s as the key approach for investigating enzymatic mechanisms or properties and reactions of small, but biologically relevant, molecules. The combined use of these tools, so-called QM/MM and lately QM/MM-MD techniques enables precise descriptions of biological phenomena and reactions.

With the exponential increase in data to be analysed, obtained through the introduction of automated whole genome and protein sequencing

Chemical Biology No. 3
Computational Tools for Chemical Biology
Edited by Sonsoles Martín-Santamaría
© The Royal Society of Chemistry 2018
Published by the Royal Society of Chemistry, www.rsc.org

techniques, the field of bioinformatics rapidly emerged in the early 2000s from the pioneering laborious mapping and comparison of protein and gene sequences in molecular biology, *via* an intense phase, which to a large extent can be viewed as 'database mining' and the development of efficient computer based algorithms, into a science of its own, which today has reached a high level of maturity and sophistication. Tools in bioinformatics are nowadays used with great success in structural biology, computational chemistry, genetics, molecular biology, the pharmaceutical industry, pharmacology and more. The aspects of bioinformatics included herein focus on protein structure determination (often referred to as homology modelling), and the tools of database screening and prediction used in drug design.

In this chapter, a brief outline of simulation techniques are given, focusing on the interface between biology and medicinal chemistry; that is molecular mechanics/molecular dynamics to explore the evolution of a system, homology modelling to determine protein structures, and the use of bioinformatics tools such as molecular docking and pharmacophores in drug design. The aim is to provide a brief introduction to a vast and rapidly growing field. In subsequent chapters, more specialised applications are presented, that build upon the foundations given herein. The chapter is in no way intended to be an exhaustive coverage of the entire area of biomolecular simulations, and we have deliberately avoided the inclusion of quantum chemical methods.

The interested reader wishing to dig deeper into the basics of computational modelling is referred to any of the many excellent textbooks available.[1–11]

1.2 Techniques in Biomolecular Simulations

1.2.1 Molecular Mechanics and Force Fields

The palette of computational chemistry methods has become increasingly versatile. Starting from quantum chemistry, where molecular orbitals and electrons occupying these are described, allows us to calculate any physical or chemical property that directly depends on the electron distribution; reaching all the way to coarse-grained molecular dynamics simulations, where groups of atoms described as beads interacting by laws of Newtonian mechanics, providing valuable insights into the complexity of biological processes on a bigger, cellular level scale. For comparison, a feasible size of a system treated by quantum chemistry calculations, even today, does not exceed a few hundred atoms, whereas the empirical methods, *e.g.* molecular mechanics (MM), can easily handle several hundred thousand atoms, and in case of a coarse-grained approach—several million atoms. Thus, the latter class of methods has become popular among researchers dealing with biomacromolecular systems, which exist and function in aqueous solutions or lipid environments. The surrounding environment could take up to 90% of

all atoms in a model system, and its presence is crucial for the correct representation of living matter. The giant leap in system size is possible due to reasonable simplicity of the MM potential energy functional. The potential energy is calculated by adding up the energy terms that describe interactions between bonded atoms (bonds, angles and torsions) and terms that describe the non-bonded interactions, such as van der Waals and electrostatic interactions (eqn (1.1)).

$$V(r^N) = \sum_{\text{all bonds}} k_l (l - l_0)^2 + \sum_{\text{all angles}} k_\theta (\theta - \theta_0)^2 + \sum_{\text{all torsions}} \frac{1}{2} V_n [1 + \cos(n\omega - \gamma)] + \sum_{j=1}^{N} \sum_{i=j+1}^{N} \left\{ \varepsilon_{ij} \left[\left(\frac{r_{0ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{0ij}}{r_{ij}} \right)^{6} \right] + \frac{q_j q_i}{4\pi\varepsilon_0 r_{ij}} \right\} \tag{1.1}$$

The bonded terms represent the stretching of bonds (l), bending of valence angles (θ) and rotation of torsional angles (ω); *cf.* Figure 1.1. Three force constants: k_l, k_θ and V_n characterise the energetic cost relative to the equilibrium value, needed to increase the value of a bond length (l_0), angle (θ_0) or rotation around a torsion angle. The torsion term represents a periodic rotation of a dihedral angle with periodicity n and phase γ. The non-bonded energy is the sum of repulsion, attraction and electrostatics between non-bonded atoms. The parameter ε_{ij} is related to the well-depth of Lennard-Jones (LJ) potential, r_{0ij} is the distance at which the LJ potential has its minimum. q_i is the partial atomic charge, ε_0 is the vacuum permittivity, and

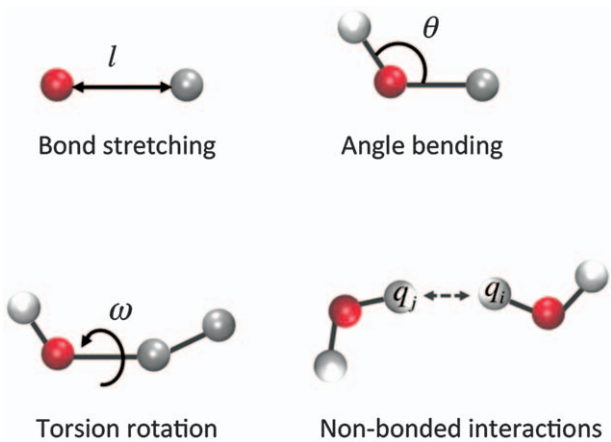

Figure 1.1 In molecular mechanics, molecular systems are treated by means of classical physics: atoms are represented as charged spheres, which have bonded (bond stretch, angle bend and torsional angle rotation) and non-bonded interactions (van der Waals and electrostatics).

r_{ij} is the distance between atom i and atom j. The LJ and Coulomb potentials describe the short-range non-bonded interactions. The evaluations of the long-range electrostatic interactions can be difficult and was often ignored beyond a specific cut-off distance resulting in approximations in a calculation. With the introduction of Ewald summation and particle mesh Ewald (PME) method long-range electrostatic calculations have become significantly more accurate.[12,13]

The simplicity of the potential energy functional form means, on the one hand, fast and easy calculations, and on the other hand that the accuracy of the empirical methods is highly dependent on the set of empirically derived parameters describing atoms and their interactions. These parameters are either derived from *ab initio* or semi-empirical quantum chemistry calculations on small model systems or by fitting to experimental data, *e.g.* X-ray and electron diffraction, NMR and IR spectroscopy. The potential energy functional form and the empirically derived parameters can be both referred to as a force field.

There are a number of empirical force fields families available, having different degrees of complexity, and oriented to treat different kinds of systems. The most popular ones designed for biological macromolecules are AMBER,[14,15] CHARMM,[16] and GROMOS.[17] Other force fields, such as OPLS[18] and COMPASS[19] were originally developed to simulate condensed matter; GAFF[20] a force field developed to simulate organic compounds together with bio-macromolecules; and GLYCAM[21] a force field specifically developed for carbohydrates. Both GAFF and GLYCAM are compatible with AMBER. These force fields vary slightly as to the functional form of the potential energy functional, mainly in the non-bonded terms, as well as values of specific atomic parameters. For more details the reader is referred to a recent review on current advances in empirical force fields for biomolecules.[22] For coarse-grained systems, the most commonly used force field is MARTINI,[23,24] which has been parameterised for lipids, proteins, carbohydrates and nucleic acids. Recently, a tool was developed to parameterise small molecules automatically.[25] The MARTINI model is based on a four-to-one mapping, implying that about four heavy atoms are coarse-grained to a single bead. The beads interact predominantly by Lennard-Jones parameters together with harmonic bonds and angles.[23] Other coarse-grained models commonly used are GROMOS[26] and Elba.[27]

1.2.2 Basic Simulation Techniques

To explore the energy landscape described by the molecular mechanics force field, *i.e.* to sample molecular conformations, a simulation is required. This is also the route to relate the microscopic movements and positions of the atoms to the macroscopic or thermodynamic quantities that can be measured experimentally.[28] There are two major simulation methods to sample biomolecular systems: molecular dynamics (MD) and Monte Carlo (MC) (Figure 1.2).

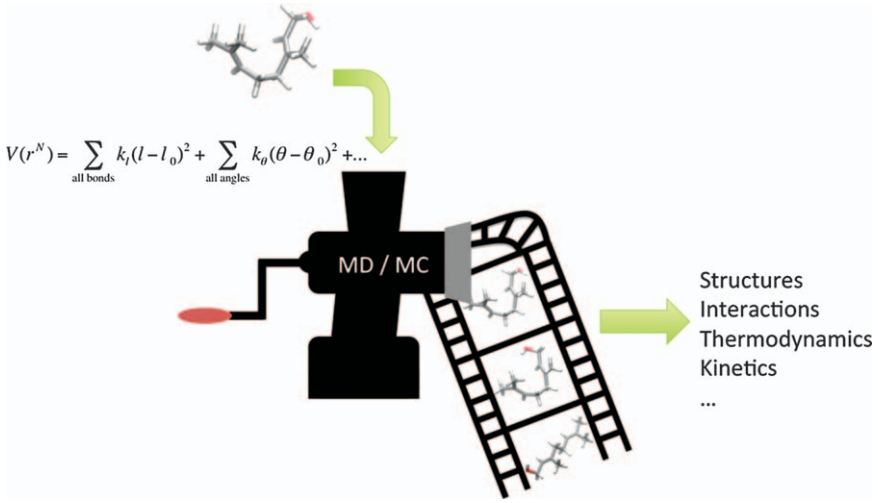

Figure 1.2 Simulation techniques such as MD and MC use a force field and a starting geometry to generate a set of molecular structures or conformers, which can be analysed to recover *e.g.* structures, interactions and thermodynamic and kinetic quantities.

1.2.2.1 Molecular Dynamics

Molecular dynamics is based on Newton's second law of motion, which relates the force, **F**, acted upon an atom to its acceleration, **a**, *i.e.* the second derivative of the position, **q**, with respect to time, t (eqn (1.2))

$$\mathbf{F} = m\mathbf{a} = m\frac{\mathrm{d}^2\mathbf{q}}{\mathrm{d}t^2} \tag{1.2}$$

where m is the mass of the atom. In a molecular simulation, time is discretised and the position after a small, finite time, Δt can be computed using a simple Taylor expansion (eqn (1.3))

$$\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \frac{\mathrm{d}\mathbf{q}(t)}{\mathrm{d}t}\Delta t + \frac{\mathrm{d}^2\mathbf{q}(t)}{\mathrm{d}t^2}\frac{\Delta t^2}{2} + \cdots \tag{1.3}$$

and hence it is easy to see that the position $\mathbf{q}(t)$, velocity $\mathrm{d}\mathbf{q}(t)/\mathrm{d}t$ and acceleration $\mathrm{d}^2\mathbf{q}(t)/\mathrm{d}t^2$ are sufficient for propagation of the molecular system. The acceleration can be computed from eqn (1.2) and the force **F** is obtained by differentiating the energy of the system.[29]

An MD simulation is setup by assigning initial velocities and positions to all atoms in the system. The velocities are usually randomly assigned, whereas the positions are typically taken from *e.g.* a crystal structure or idealised geometries. Thereafter, the force acting on each atom is calculated, giving the direction of movement. The atoms are moved in this direction, giving new forces on each atom, and the procedure is then repeated a

number of steps. There are several numerical recipes describing how this integration of motion is done practically, *e.g.* leapfrog, Verlet or velocity-Verlet. They chiefly differ in the numerical stability and whether they in addition to propagating positions also propagate the velocities.[29]

A major limitation to an efficient sampling with MD is the discrete time step, Δt. It is desirable to choose a longer time step, which would give longer simulations with less computational resources. However, Δt is limited by the fastest motion in the simulated system. For an atomistic system, the fastest motion is the bond vibration between a hydrogen and a carbon atom, which limits Δt to about 1 fs. Therefore, these bonds are typically constrained in the simulations, allowing a 2 fs time step. An alternative is to increase the mass of the hydrogen atoms, effectively slowing down the bond vibration.[30] In coarse-grained simulations, a much larger time step is possible, typically between 10 and 40 fs, depending on the model.[23,27]

A simulation obeying Newton's second law of motion can be shown to sample a thermodynamic ensemble with constant number of particles, volume and total energy (kinetic + potential). However, experiments are usually performed at constant temperature and either constant pressure or volume. To sample such an ensemble, the equations of motion have to be modified. In the case of constant temperature, a thermostat is required and there are many such algorithms. Common approaches include (1) modifying the velocities (*e.g.* weak-coupling), (2) introducing fictitious particles in an extended system (*e.g.* Nosé–Hoover), or (3) introducing friction (*e.g.* Langevin dynamics). An extensive discussion of different thermostats is however beyond the scope of this introductory chapter and interested readers are referred to a review on the subject.[31] Similarly to temperature, constant pressure can be introduced by a barostat that modifies the volume of the simulated system. Common approaches include (1) scaling the box dimensions (*e.g.* weak-coupling), (2) introducing fictitious particles (*e.g.* Parinello-Rahman), or (3) introducing a piston. Some thermostats and barostats are better suited for systems far from equilibrium, whereas others are better for production simulations.

1.2.2.2 Monte Carlo Simulations

The other major sampling method, Monte Carlo (MC), is a statistical technique where new conformations are generated by a random walk in phase space by assigning random displacements to the internal degrees of freedom, *i.e.* bonds, angles and torsions. Naturally, all conformations are not equally likely and therefore, the sampling is biased such that conformations are generated with a probability prescribed by the thermodynamic ensemble of interest. The overwhelmingly most common way to accomplish this is by performing a Metropolis–Hastings test.[32,33] In a Metropolis MC simulation, a new conformation is accepted with the probability, p (eqn (1.4))

$$p = \min[1, \exp(-\Delta U/kT)] \tag{1.4}$$

where ΔU is the energy difference between the new and old conformation, k is the Boltzmann constant and T the absolute temperature. In practice, the energy of the new and old conformation is compared and if the new conformation is lower in energy it is retained for the next step. If it is higher in energy, the Boltzmann factor $\exp(-\Delta U/kT)$ is compared to a uniform random number between 0 and 1, and if the Boltzmann factor is higher than the random number the new conformation is kept.

An MC simulation consists of a number of moves, which is a recipe on how to sample specific degrees of freedom. This can be a simple translation move where the center of mass of a molecule is displaced, a rotation about a torsion angle or a complicated, concerted move of several protein backbone atoms. A move is selected randomly followed by a Metropolis test of the new conformation and the procedure is repeated for a number of steps.

The Metropolis test illustrated above gives a canonical ensemble, *i.e.* constant number of particles, volume and temperature. However, it can be modified to allow for a volume change such that constant pressure is simulated. Furthermore, entire molecules, *e.g.* water, can be inserted and removed during the simulation, leading to a grand canonical ensemble.[34] Thus, MC simulations are more versatile than MD simulations, but are heavily dependent on the construction of efficient moves. In addition, since MC only depends on the positions of the atoms, dynamic information is lacking, and MC cannot be used to *e.g.* estimate transport properties or diffusion constants.

1.2.2.3 Boundary Conditions

An important aspect of both MD and MC is the choice of boundary conditions.[28] Typically, a molecule is solvated by a finite water shell, or inserted in a lipid bilayer, leaving some of the atoms facing vacuum. This is not good physical description of a biological system, and a well-used solution is to extend the system periodically in all three directions to represent a pseudo-infinite system, effectively removing the vacuum. Periodic boundary conditions can be used with various geometries, a cubic box, a rhombic dodecahedron, or a truncated octahedron. The latter scheme is common in simulations of biological macromolecules solvated in water, since it allows the least number of solvent molecules in the system and thus speeds up the computation. Although periodic boundary conditions are the most common choice, there are other solutions in use, *e.g.* spherical boundaries with addition of restraints.[35]

1.2.2.4 Enhanced Sampling Techniques

As mentioned earlier, efficient sampling is one of the major limitations of both MD and MC. MD trajectories might not reach all relevant conformations, for example short-lived transient states connected with a biological function. This problem can be addressed by employing enhanced sampling

algorithms, such as metadynamics, steered MD or replica exchange MD. Often, the aim of such enhanced samplings is to build a more complete energy surface and/or to obtain free energy profiles or potential of mean force (PMF) data. Some of the more advanced features are covered in subsequent chapters of this book; we also refer the reader to recent reviews on enhanced sampling techniques.[36,37]

1.2.3 Basic Data Analysis

After a simulation has been completed, it needs to be analysed to extract relevant information about the system of interest. This can be quite challenging and depends very much on the type of a simulated system. Here, a few common strategies for analyses will be outlined.

1.2.3.1 Proteins

It is common to estimate the equilibrium of a protein simulation by computing the root mean squared deviation (RMSD; eqn (1.5)) of the backbone atoms compared to the starting conformation,

$$\mathrm{RMSD} = \sqrt{\frac{1}{N}\sum_{N}(\mathbf{r}_i(t_0) - \mathbf{r}_i(t))^2} \tag{1.5}$$

where N is the number of atoms, $\mathbf{r}_i(t)$ the position of atom i, at time t. Prior to the analysis the protein need to be fitted onto the starting structure to remove the overall translation and rotation. Although this is a straightforward analysis and gives an indication of local equilibrium, it is a far too simple method to assess the global convergence of the simulation. It is also possible to compute a pair-wise RMSD between each snapshot in a simulation. This could for instance be used in order to evaluate how efficient the sampling has been, or if the simulation has become stuck in a local energy well (Figure 1.3). Whereas the RMSD provides an overall estimate for the entire protein and an approach to assess the degree of movement of individual residues is to compute the root mean squared fluctuation (RMSF; eqn (1.6)), which is simply the variance of the position of an atom:

$$\mathrm{RMSF} = \frac{1}{T}\sum_{T}(\mathbf{r}(t) - \bar{\mathbf{r}})^2 \tag{1.6}$$

where T is the total time of the simulation (or number of snapshots) and $\bar{\mathbf{r}}$ is the average position. The RMSF can be related to the B-factor used in crystallography by multiplying by $8/3\pi$. The analysis can be done on a per residue-basis, where all the atoms of a residue is included in the average and can for instance be used to assess the movement of sidechains. Alternatively, one can include only C_α atoms in the analysis to assess the backbone movement.

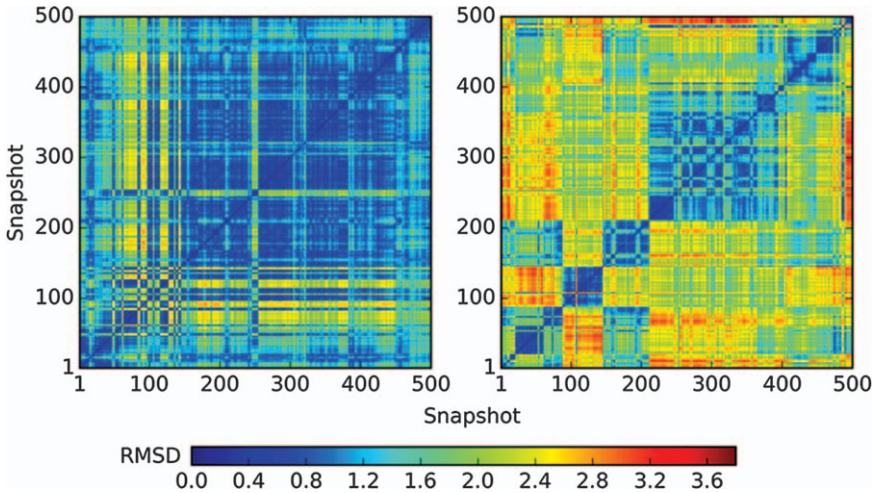

Figure 1.3 The pairwise RMSD between each snapshot in two MC simulations of a small drug-like molecule. The simulation to the left is stuck in a local energy-well as shown by the small RMSD between the snapshots, whereas the simulation to the right displays larger RMSD, indicating more diverse sampling.

To assess the overall compactness of a protein, the radius of gyration can be computed from eqn (1.7):

$$R_g^2 = \frac{1}{M}\sum_i m_i(r_i - R_c)^2 \tag{1.7}$$

where M is the total mass of the protein, m_i the mass of atom i and R_c the mass center. This is a simple analysis to determine if the protein is compact or extended. In order to obtain more specific analysis on the protein structure, one can analyse the secondary structure. It is possible to classify each amino acid to determine if it is part of a helix, a beta sheet or a loop. This is useful to monitor during the simulation in order to detect large conformational changes, *i.e.* loss of secondary structure. More complicated motions can be investigated with a principal component analysis (PCA). This is a statistical technique that reduces the dimensionality of problem; instead of looking at all $3N$ coordinates in a simulation, PCA reduces this to a few principal components that describes the major movements. The principal components are computed from the eigenvalues of a covariance matrix, describing the covariance of the positions of selected atoms. For a protein, typically the C_α atoms are analysed. The principal component can be projected onto the simulated system and visualised, enabling straightforward inspection of the major motions. This could for instance be a breathing movement of two protein domains or the outward movement of a loop area upon binding of a small molecule.

1.2.3.2 Nucleic Acids

The conformational space of DNA or RNA is quite diverse and dynamic, reflecting their ability to change depending on the physicochemical properties of the surrounding environment, local sequence motif, and interactions with other molecules. Thus, apart from RMSD and analysis of the inter and intramolecular networks of contacts, assessment of nucleic acids simulations is directed towards capturing such conformational interplay. The geometry of DNA, and to some extent RNA, can be described in terms of helical parameters (pitch and diameter of the helix), groove parameters (depth and width), furanose ring conformation, six torsional angles of the backbone, rotational (tip and inclination) and translational base pair parameters, six intra-base parameters (buckle, propeller, opening, shear, stretch, and stagger), and six inter-base parameters (tilt, roll and twist, shift, slide, and rise). The most popular programs that analyse nucleic acids simulations in terms of the mentioned degrees of freedom include Curves+ and Canal,[38] and 3DNA.[39] As nucleic acids exist and function as salts, the behaviour of the surrounding counterions is an integral part of the analysis, and could now be done with, *e.g.*, the program Canion.[40]

1.2.3.3 Membranes

To assess the equilibration of a membrane simulation, several simple geometric properties are typically calculated such as the area and volume per lipid as well as the thickness of the membrane.[41] The area and volume can be calculated straightforwardly from the box dimensions and by assuming the water density. There are several kinds of thicknesses that can be computed, but a simple one is to measure the distance between the peaks of the density of the phosphate atoms (or equivalent). It is also common to calculate an order parameter of the fatty acyl chain (eqn (1.8))

$$S = \frac{1}{2}\left\langle 3\cos^2\theta - 1 \right\rangle \tag{1.8}$$

where θ is the angle between the bilayer normal and a carbon–deuterium bond in the acyl chain, and the brackets indicate an average over the simulation. For coarse-grained simulations, the bond between two neighboring atoms replaces the carbon–deuterium bond. It is therefore not correct to compare order parameters from atomistic and coarse-grained simulations. In both cases, the order parameter gives information on the phase of the membrane, *i.e.* if it is in the fluid or liquid-ordered phase.

1.2.3.4 Small Molecules

When performing simulations on small molecules (either as solvated entities, in a larger 'bulk' system, or as part of *e.g.* a biomolecular complex), an interesting analysis to perform is clustering. This will provide information

on different kinds of conformations the molecule attains during a simulation and can be used to both assess if the simulation is stuck in a local conformation and to investigate the probability of different conformations. There is a plethora of different clustering methods available and it outside the scope of this text to discuss them at any length; the interested reader is referred to the literature.[42]

1.2.4 Software

Over the last two decades with the development of new simulation algorithms and new technologies in hardware platform design, molecular simulations have dramatically increased in size, length and system complexity. The appearance of a variety of standardised molecular modelling software packages, including GROMACS,[43] Amber,[44] CHARMM,[45] GROMOS,[46] and NAMD,[47] has transformed the field of computational chemistry by commoditising molecular simulations and making it accessible to a broader group of researchers. All these packages have complementary strengths and profiles, with GROMACS and NAMD being two of the most popular. Considering GROMACS and NAMD as only MD engines there is no dramatic difference as to their performance, both work with a variety of force fields, and have GPU acceleration implemented. However, small differences should be mentioned, such as the possibility to perform QM/MM simulation in GROMACS, or NAMD's extensibility to user-written scripts. Both packages are distributed free of charge with source code. Moreover, for NAMD, there are downloadable binaries for a variety of platforms. This can be useful for a beginner in computational chemistry, as compilation of MD software might not always be straightforward. Both GROMACS and NAMD are parallel molecular dynamics engines, designed for high-performance simulations of large biomolecular systems, with GROMACS being better for simulations of smaller systems on medium-size supercomputers. To achieve the best performance for a particular system on a particular supercomputer we recommend initial benchmarking. For both GROMACS and NAMD a wide variety of tutorials are available. External software packages, like PLUMED,[48] can provide additional functionality, such as enhanced molecular dynamics techniques mentioned above.

Various pieces of software are used for visualisation and analysis of molecular dynamics trajectories. Among the most popular and freely accessible tools are molecular modelling programs VMD[49] and USCF Chimera.[50] VMD (visual molecular dynamics) is a specialised molecular visualisation program for displaying, animating, and analysing molecular dynamics trajectories, extensively used with any MD software. USCF Chimera, on the other hand, is a highly extensible program for interactive visualisation and analysis of molecular structures and related data, including density maps, supramolecular assemblies, sequence alignments, docking results, trajectories, and conformational ensembles. Both programs could be used to create professional illustrations. VMD and USCF Chimera can also perform basic structural analysis, but for more extensive assessment of trajectories, such as

clustering or modifications of system topologies, AmberTools and, in particular, cpptraj[51] is recommended.

1.2.5 Examples

Molecular simulations have been used in numerous applications, to obtain the structure and dynamics of many biomolecules in order to elucidate biochemical functions, processes and pathways. In principle, any molecular system can be simulated, and two typical examples are shown in Figure 1.4. Although many of the simulations have studied individual macromolecules solvated in water or a model membrane, there has also been some effort in looking at larger assemblies. Already a decade ago, Schulten and co-workers used a massively parallel supercomputer to study the complete satellite tobacco mosaic virus, described with an all-atom force field.[52] They were able to accumulate 50 ns of simulation time of a system containing roughly 1 million atoms and could conclude that the virus capside became unstable upon removal of the core RNA molecules. Using a coarse-grained model, Sansom and co-workers simulated the influenza A virion, a significantly larger system.[53] By using a CG model, they could simulate at the microsecond timescale and in addition investigate alterations to the membrane envelope and sensitivity to temperature.

Long-time scales, such as micro- or milliseconds, are generally not accessible when using an all-atom force field. The exception is if special-purpose hardware is used as in the work from Shaw and co-workers. They reported the first continuous millisecond simulation of a protein described with an all-atom force field when they studied the folded structure of BPTI.[54] Several distinct conformational states were found, separated by large kinetic barriers. Shaw and co-workers has also used simulation techniques to find flaws in current protein force fields, which could not previously be detected due to the typically short simulations. An alternative to running long,

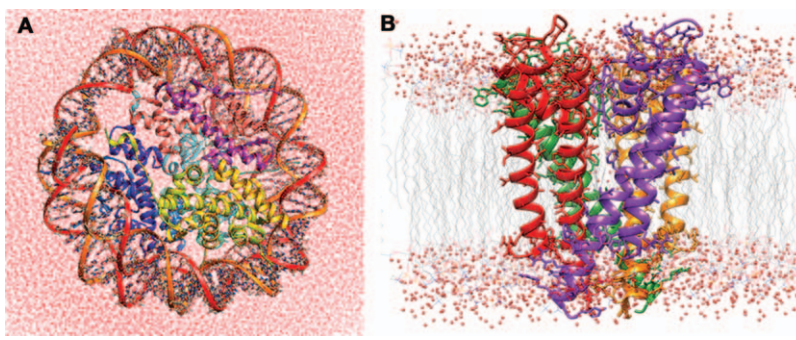

Figure 1.4 Examples of complex systems that can be simulated by all atom MD: (A) nucleosome particle (PDB id: 5B2I) in 20 Å octahedral box of explicit water, total 205 084 atoms; (B) potassium channel (PDB id: 1K4C) in 100×100 Å POPC membrane patch, total 54 000 atoms.

continuous trajectories that has become popular lately is to assembly many short simulations using Markov state models.[55] They have for instance been used to study protein folding and how this immensely complicated process is affected by solvent effect and electrostatics.[56]

The above-mentioned examples represent extremes, both in terms of system size and length of the simulations. Molecular simulations of much more modest dimensions are routinely used to gain insight into biological processes and to complement wet-lab experiments. We find a very fruitful application of simulations in the field of enzymology.[57] Here the substrate, a few active site residues and coordinating waters and ions are described with quantum mechanics, whereas the rest of the system is described with a molecular mechanics force field. Such simulations have been used in numerous applications to for instance elucidate enzymatic mechanism, understanding the nature of the catalytic power and enzyme design.[57,58] Another common application of molecular simulation is the estimation of binding free energies of small molecules, *e.g.* drugs, to their targets. Jorgensen and co-workers routinely use such simulations to aid in their drug design pipeline.[59] This is done by systematically introducing small chemical groups such as a hydroxyl or methyl group in a lead compound and evaluating its contribution with alchemical free energy simulation.

The above illustrations provide a few examples of what is possible with molecular simulations, with many more provided in the subsequent chapters of this book.

1.3 Protein Structure Prediction

Protein structure prediction is often listed among activities within the bioinformatics area, and essentially covers approaches enabling us to go from primary sequence (be it nucleic or amino acids), *via* secondary and tertiary structure, to quaternary structure and possibly also function of the resulting protein. This follows the central assumption that a proteins primary sequence and the inherent properties of the amino acid side chains dictate the final folded three-dimensional structure. Besides the above predictions, which are generally obtained through knowledge-based potentials or algorithms, or by comparing to already existing structures of systems with similar amino acid sequence, analysis of the quality of the resulting model is an essential part of protein structure prediction. We will in this section go through the different steps involved in structure prediction, including tools for analysis and some of the available software and web based solutions.

1.3.1 Sequence Alignment and Secondary Structure Prediction

Assuming we know the primary structure; that is, the amino acid sequence of our protein (or, if we have the DNA sequence, have translated this to the

corresponding amino acids, and assuming there are no post-translational modifications that will alter the sequence), the next step is normally to search a database of protein structures for homologous sequences to which we can compare our query through sequence alignment. Normally, the coordinate repository of X-ray, NMR and EM structures of the RCSB protein databank from Brookhaven National Laboratory (www.rcsb.org), or any of its sister sites (PDBe, PDBj, BMRB), or a related database including refined protein structures, is used. To date the protein databank contains over 110 000 solved protein structures, with an annual growth of around 8500–9000 new entries, as well as structures of DNA, RNA and protein–nucleic acid complexes.

The search against databases to identify homologous sequences is normally performed using BLAST (Basic Logical Alignment Search Tool)[60] or FASTA.[61] Nowadays BLAST, which comes in several variants depending on type of algorithm, sequence and database, is the more common. A BLAST search systematically compares three-letter segments of the query sequence, referred to as words, to the database of templates step by step in a heuristic approach. For example, a sequence AHKRV is searched as the words AHK, HKR, KRV; this initial search is referred to as 'seeding'. Comparison of words from the query sequence with words the database of known sequences is made both based on identity (each residue is matched perfectly), and similarity (similar function/property/size, but not identical), and a total score is calculated using a scoring matrix such as BLOSUM62 (BLOck Substitution Matrix).[62] For example, according to BLOSUM62, an arginine matched by another arginine is given the value +5, arginine *vs.* lysine is +2, and arginine *vs.* cysteine is −2. Also, other scoring matrices and approaches exist, such as identical scoring matrix, minimal mutation distance matrix and point-accepted mutation (PAM).

After the 3-letter word search is done, the word length is extended to nearest and next-nearest neighbours, and possible alternative alignments assessed and scored. An example of the latter is shown in Figure 1.5A assuming a query sequence AHRKCCVGA to be matched against the template sequence AGRKKCVGGA, where different parts given as gaps or insertions provide different scores and result in shorter segments or additional loops of the modelled protein.

If we have several templates to compare against, we must, in addition to the possible alignments as above, also consider which of those alternative yet slightly mismatching sequences that fits the best; *e.g.* assuming we again have our query sequence AHRKCCVGA, is AHRK***SV***CVG***G***A or AHR***AC***K***V***CVGA a better template (*cf.* Figure 1.5B)? In the case of multiple sequences to which we compare our query, this is referred to as multiple sequence alignment, of which the most common methods are the iterative PSI-BLAST[63,64] and CLUSTALW.[65] Sequence alignment approaches are also commonly used to explore similarity of a certain protein between different species, to identify conserved residues and motifs, and similar.

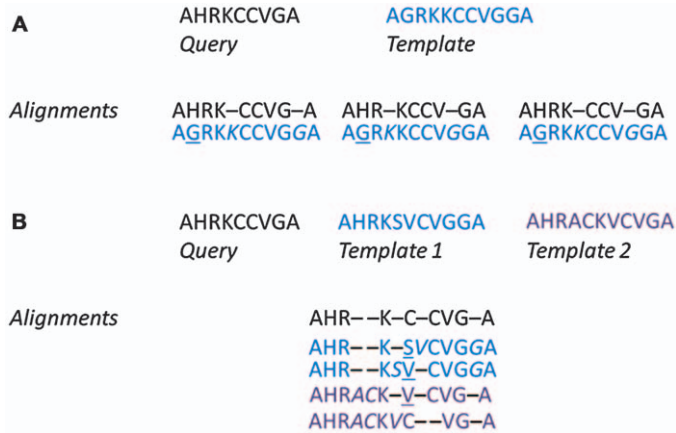

Figure 1.5 Examples of different alignments to same sequence (A), and multiple sequence alignment (B).

Once the best-scoring template(s) have been determined, we next compute the most likely secondary structure elements of the query sequence, again by stepwise comparing 3- and 5-letter segments and their likelihood of forming any of the common motifs. Each amino acid is scored (often using a scale from 0 to 9) based on probability to attain a certain structural motif in its local environment, and the prediction is compared to the template structure(s) that have been selected from the (PSI-)BLAST search. Secondary structure predictions are commonly displayed or mapped graphically in some way, *e.g.* red bar for helix, yellow arrow for β-sheet (*cf.* Figure 1.6A as an example). The aim is to determine which parts of our query sequence that locally are more likely to form α-helices, which sections that have a propensity for formation of β-sheets, where loops or coil-regions will be. This, again, is used to match the similarity against the obtained templates, and assist in the folding predictions. In addition, predictions are also frequently made based on the properties of amino acid sidechains, of which segments or parts of the sequence that are more hydrophobic or more hydrophilic. The rationale for this in accurate structure determination becomes obvious if we *e.g.* compare a globular protein present in the cytosol (hydrophilic surface and hydrophobic interior) with a membrane spanning ion channel (hydrophobic residues on the outside, interacting with the membrane lipids, and hydrophobic residues in the interior, lining the pore).

1.3.2 Comparative Modelling Approaches

Having determined suitable templates for our query sequence, and most likely secondary structure elements, *i.e.* the primary fold of the sequence, we next organise or pack our query structure according to the templates in order to generate a tertiary structure model. This is referred to as comparative

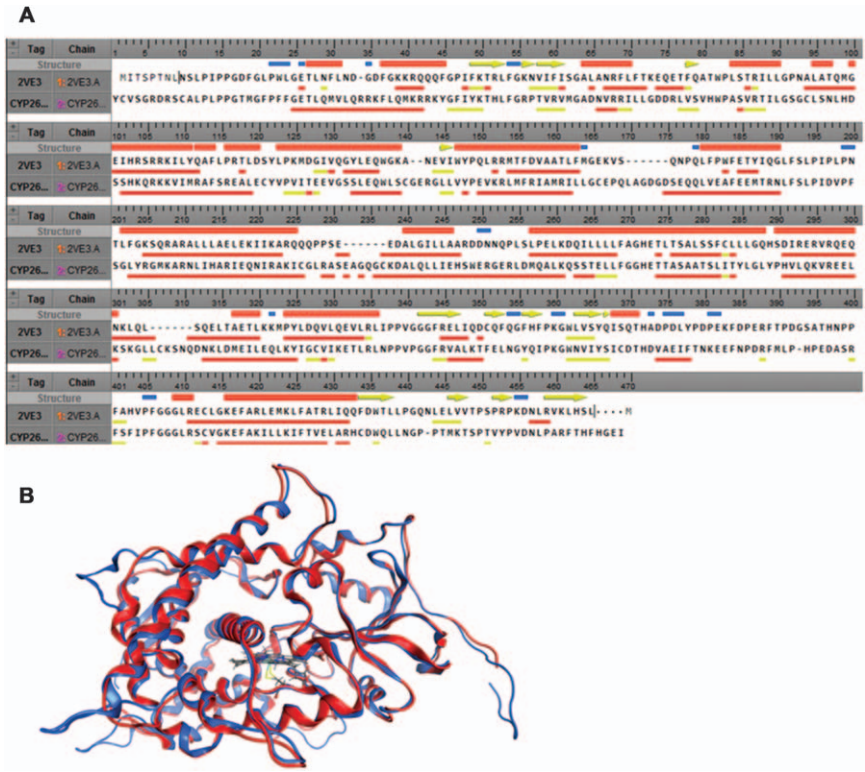

Figure 1.6 (A) Sequence alignment of template CYP120A1 (PDB ID 2VE3) and amino acid sequence of CYP26A1 based on this template. Top row in each segment displays consensus secondary structure (red for α-helix, yellow for β-sheet and blue for loops, and below each sequence the analysed or predicted secondary structure is displayed. (B) Superposed homology model of CYP26A1 (blue) and its template CYP120A1 (red).

protein modelling, and relies on the fact (as far as we can conclude from the currently available determined protein structures) that, although the number of possible proteins is essentially infinite, the number of folds is limited to approximately 2000 different 'types'. That is, provided the sequence similarity (or ~identity) between two proteins is sufficient, the two structures will in all likelihood have essentially the same backbone topology in the aligned regions.

Normally one separates the field of comparative protein modelling into homology modelling and threading/fold recognition. In homology modelling, we assume that if two sequences are so closely related that they can be satisfactorily aligned, they will also attain the same three-dimensional structure. This approach views the problem from the standpoint that folds are more evolutionarily conserved than the actual sequences. Clearly, the more identical the sequences, the better the model—if two sequences share 70% identity—the accuracy of the modelled structure is claimed to be

similar to that of a crystal structure with resolution (Cα RMSD) in the range of 1–2 Å; at 25% identity, the structure corresponds to a resolution of 2–4 Å.

If there are no apparent homologous proteins identified, secondary structure prediction is necessary and compared to the template database, where after the sequence is 'threaded' according to the fold of the best-matching recognised template(s). This is also referred to as 3D–1D fold recognition, as it links a primary sequence to a three-dimensional structure. 3D–1D alignment is sometimes also included as an intermediate step in the above homology modelling.

There are a number of variants to the above, such as fragment assembly and segment matching; which in essence means that smaller parts of the sequence are modelled separately and then combined into a full protein structure.

In the unfortunate event that no reasonable templates can be identified, the 'last resort' is referred to as *ab initio* structure prediction. In this case the secondary structure elements are assembled stochastically, normally using a Monte Carlo type of algorithm, combined with refinement and (possibly) shorter simulations in order to generate a large number of potential three-dimensional models which are assessed and either discarded or improved further in successive iterations. One must, however, be very cautious when it comes to the interpretation of protein structures generated entirely without prior knowledge or templates, as the uncertainty in predicting the appropriate spatial arrangement between secondary structure elements is very high.

Regions of high flexibility, such as loops or the C- and N-termini, are normally poorly resolved or missing in X-ray crystal structures, and loop modelling is hence one of the approaches by which homology modelling can be used in order to improve a protein structure. Care should, however, be taken if the modelled loop is longer than ten amino acids.

Once a model is available, side chains need be optimised (packed) properly. This is done by successively evaluating the energetics for different rotameric states of the sidechains, either by actual energy calculations or using rotamer libraries, and determining the lowest possible (*i.e.* most stable) overall configuration. Finally, the model is generally subjected to some form of energy relaxation or minimisation. In Figure 1.6B, an example of a homology model is shown, and superposed to its template. The overall Cα RMSD of 1.75 Å over 433 residues, with 31.5% sequence identity between query and template. As seen, the agreement for the ordered secondary structure regions is very high, whereas the main deviations are noted in the loops and termini.

Several programs also have the capability to generate models from different templates, and merge the best-matching local segments thereof to construct a hybrid multiple template model. In general, one main template is in those cases used for the core structure, and replacing smaller fragments that are less accurately determined, such as loops or stretches where the sequence similarity to the 'core template' is particularly low.

1.3.3 Function Prediction

Predicting the explicit role or function of a protein, or different parts thereof, is more difficult, and is compared to the above protein structure determination still at a very early stage of development. In essence, protein function prediction relies heavily on identifying homologous regions either by sequence motifs or by 3D structure alignment, to identify possible domains and, by analogy with the identified templates, their specific roles. However, herein lies also the aspect of paralogs—proteins that have evolved from a common ancestor into structurally very conserved entities but where the functional role is entirely different—which makes the task even more difficult. It lies beyond the scope of the current introductory chapter to also cover such aspects, although we do mention some servers and other tools in Section 1.3.5 below.

1.3.4 Analysing the Quality of the Modelled Structure

Once a structure has been modelled, it is crucial to also assess its quality. Some quality checks are already embedded in the routines employed in the model development (such as BLAST *E*-value, BLOSUM score, side chain dihedrals and packing score), but it is recommended that a thorough assessment is made using some of the many servers and programs available. Some of the key tools are included herein.

The first assessment to be made is to produce a Ramachandran plot of the obtained structure, which displays the values of the ϕ and ψ angles of the protein backbone, and which provides a picture of the stereochemical quality of the amino acids. The RAMPAGE server (http://mordred.bioc.cam.ac.uk/~rapper/rampage.php) is one of the main tools for this. A large number of outliers (*i.e.* values of the backbone torsional angles that do not fall into the allowed or generously allowed regions) indicates that the model has significant problems.

Secondly, the folding reliability can be evaluated using the Verify3D server (http://services.mbi.ucla.edu/Verify_3D). This assessment evaluates the likelihood that a particular residue in a particular sequence context part-takes in the predicted 3D fold, and provides an estimate of correct *vs.* incorrect folding for each amino acid. Scores below zero indicate serious folding problems, and 80% of the residues of a protein should attain values $\geq$0.2 in the 3D–1D profile for the model to have acceptable folding reliability.

The absolute quality of a model can be obtained using the QMEAN *Z*-score server (http://swissmodel.expasy.org/qmean/cgi/index.cgi), which calculates the quality of the model by combining six different structural descriptors such as secondary structure elements, solvent accessibility and torsional angles. The normalised mean value is then compared against the corresponding vales of a non-redundant set of high-resolution experimental structures of similar size ($\pm$10%) solved through X-ray crystallography.

Also, many other quality and property checks can be performed besides the basic tools listed above. Examples are determination of druggable sites and their relative scores, relating to docking simulations; see Chapter 1.4,

using *e.g.* ProteinsPlus, http://proteinsplus.zbh.uni-hamburg.de (recently changed from DoGSiteScorer), and metal binding site interactions, if present, through the sever CheckMyMetal (CMM), http://csgid.org/csgid/metal_sites.

1.3.5 Software and Web Based Servers

There are several programs available for protein structure determination, both as standalone codes to be installed on the users own computer or cluster, and as web based servers. The current compilation is not intended to be exhaustive, but meant to provide a sample of different options available; most (but not all) of the programs listed are free for academic users. Each program has its pros and cons, and the reader is advised to read up on the different codes and approaches first, and to preferably test more than one code in order to build up experience in what functions the best for his or her particular needs. Lastly, a slight word of caution: Although there is a plethora of web based programs available, one must always remember that submitting your computation to someone else's computer (server) means you have no control over the results, including aspects pertaining to safety/security.

I-TASSER—server and downloadable (http://zhanglab.ccmb.med.umich.edu/I-TASSER).[66]
LOMETS—meta-server combining nine different programs (http://zhanglab.ccmb.med.umich.edu/LOMETS).[67]
MODELLER—a standalone program and server; several graphical interface programs are also available that use MODELLER (http://salilab.org/modeller).[68]
MOE—a standalone program, with license fee (http://www.chemcomp.com).[69]
PHYRE2—server (http://www.sbg.bio.ic.ac.uk/phyre2/html/page.cgi?id = index).[70]
PRIME—part of the Schrödinger package; standalone, with license fee (http://www.schrodinger.com/Prime).
RAPTORX—server and downloadable (http://raptorx.uchicago.edu).[71]
ROBETTA—server, and as part of the downloadable Rosetta3 package (https://web.archive.org/web/20150819163428/http://www.robetta.org).[72]
SWISS-MODEL—server (http://swissmodel.expasy.org).[73]
YASARA—standalone, minor license fee (http://www.yasara.org).[74]

For protein function or domain prediction, fewer programs are available as yet. However, we refer the interested reader to any of the following servers:

PFAM—Protein Families Database (http://pfam.xfam.org).[75]
dcGO—(http://supfam.org/SUPERFAMILY/dcGO).[76]
PROSITE—Database of protein domains, families and functional sites (http://prosite.expasy.org).[77]

1.4 Computer-based Drug Design

Computer-based drug design (CBDD) or computer-aided drug design (CADD) refer to the application of different computational methodologies and algorithms for developing bioactive compounds. It is currently an independent discipline within computational chemistry, mainly because it focuses on predicting/designing the next potential bioactive molecule to synthesise and test. It is well known that drug research and development is not only time-consuming, but also very expensive. It has been estimated that developing a new drug from idea to market would take *ca.* 14 years with an associated cost from 800 million to 2 billion dollars.[78,79] In fact, the overall cost is increasing every year, mainly for specialised drugs for smaller patients populations.[80] This emphasises the benefits of applying computational tools in the early stages of drug discovery, thereby reducing the cost, the required time and the inherent risks such as late-stage failure.[81] The well-known 'fail fast, fail early' pharmaceutical mantra is the goal.[82,83] While high-throughput screening (HTS) of large compound libraries is still the major source for discovering new hits in drug design, CBDD is currently playing a key role in the search for novel bioactive compounds, both in pharmacy and academy.[84] A comparison between the two techniques has been reported in the screening for novel inhibitors of Protein Tyrosine Phosphatase-1B (PTP1B), an enzyme implicated in diabetes. The HTS of 400 000 compounds resulted in 85 hits actually inhibiting the enzyme (0.021%). On the other hand, 365 high-scoring molecules were obtained from the virtual screening, 127 of which inhibited PTP1B (34.8%). These results clearly showed that CBDD increased the hit rate over random (HTS) screening.[85] Thus, the application of computational tools allows for covering a larger part of chemical space and at the time the number of compounds that must be synthesised, is drastically reduced.

CBDD can be classified into two main classes: structure-based drug design (SBDD) and ligand-based drug design (LBDD).[86,87] SBDD is based on the knowledge of the 3D structure of the target protein, using virtual screening techniques to search for molecules having complementarities toward the selected target. For SBDD, molecular docking, virtual screening and molecular dynamics are the most important underlying methodologies.[88] LBDD does not require knowledge of a protein, instead using the information provided by known active and inactive compounds to find potential hits by similarity searches or quantitative structure–activity relationship studies (QSAR).[87] The latter is usually the selected methodology when there is no structural information available of the target system.

1.4.1 Pre-requisites for SBDD—Sampling Algorithms and Scoring Functions

Molecular docking is a methodology that attempts to predict the conformation of ligands within the receptor binding site.[89] The identification of

the 'best pose', *i.e.* the ligand internal conformation and orientation towards the receptor, involves searching the ligand conformational space (sampling or posing) and ranking of the predicted binding conformations (scoring).

1.4.1.1 Sampling Algorithms

Docking most frequently deals with ligand flexibility and in some cases with protein flexibility. Virtual screening employing a large ligand library (virtual high-throughput screening, VHTS) is a very time- and resource-consuming procedure. Therefore, usually the ligand and receptor are both treated as rigid bodies (rigid ligand and rigid receptor docking) in the initial screening. Even though the search space is restricted, large libraries can be rapidly explored and filtered. More often, molecular docking employing smaller libraries treats the ligand as flexible, while the receptor is kept fixed (flexible ligand and rigid receptor docking). Finally, incorporating receptor flexibility is the most accurate and costly methodology, and it is usually employed for refining previous docking rounds (flexible ligand and flexible receptor docking).[90,91] Treatment of ligand flexibility includes systematic, stochastic, and simulation methods.[92–94]

1.4.1.1.1 Systematic Methods. These methods account for ligand flexibility by exploring the conformational space of the molecule. After search of a ligand's degrees of freedom, the method converges to the most likely binding mode. As with all 'down-hill' methods, a systematic search can converge to a local minimum rather than the global one, a problem that can be overcome by performing several searches starting from different initial ligand conformations.[95] When exploring all possible degrees of freedom in a ligand (exhaustive search), the number of possible combinations is usually prohibitive, facing the so-called problem of combinatorial explosion. An alternative to exhaustive search is to employ incremental construction algorithms.[93] Docking programs such as DOCK,[96] FlexX[97] and Glide[98] apply an incremental construction search method, in which the ligand is first divided into fragments. One fragment is selected as anchor (usually the larger fragment) and docked in the binding site. The remaining fragments are incrementally added until the entire ligand is built.[90]

1.4.1.1.2 Stochastic Methods. Stochastic methods perform a random search of the conformational space by making random changes to the ligand or a population of ligands. Such changes include translational, rotational and internal modification of the ligand's coordinates. This strategy allows for finding the global minimum, covering also a larger conformational space. MC, genetic algorithms and tabu search are typical algorithms belonging to this class.[92,94]

MC methods make random modifications to the ligand structure and the resulting conformation is tested according to the metropolis criterion, which accepts conformations with a lower energy, and higher energy states when

the Boltzmann factor is greater than a random number between 0 and 1 (see also Section 1.2.2). Programs such as Autodock[99] and MOE[69] employ MC methods for sampling.

Genetic algorithms (GA) have their roots in the Darwin's theory of evolution and natural selection. All structural parameters are encoded in genes, and a particular pose is referred to as a 'chromosome'. The random search algorithm then generates several chromosomic mutations (*i.e.* several poses), which are in turn evaluated in terms of energy. The best adapted chromosome will be the one with the lowest energy and thus selected to be used in the next generation. The next generation is populated with poses having increased favourable structural characteristics. After several generations (several conformational search cycles), the energy minimum conformation is reached.[100] Programs implementing genetic algorithms are Autodock[101] and Gold.[102]

Tabu search (TS) is a heuristic method originally proposed by Glover in 1986.[103] The algorithm proceeds stepwise from a conformation, generating a number of moves to the current solution. The moves are scored, ranked using the energy function and examined. The method keeps a list of the previously visited solutions, and a move is considered 'tabu' if it generates a solution that is not sufficiently different from the previous ones. The algorithm calculates the root mean square deviation (RMSD) between the current move and the all previously recorded solutions. Only those movements having a RMSD smaller than a cut-off are accepted. The tabu search continues for a user-defined number of iterations.[104] Examples of programs implementing tabu search are MOE[69] and PRO_LEADS.[105]

1.4.1.1.3 Simulation Methods (MD). MD simulations (*cf.* Section 1.2.2) are also used in the context of molecular docking, allowing for representing the flexibility of both the ligand and the receptor. However, MD is not the best method for simulation of ligand–target interactions, mostly for its intrinsic difficulty to cross high-energy barriers, leading to a poor sampling. Considering that MD simulations are efficient at exploring the local hyper surface, the best approach is to use a systematic or random search in order to find the most likely conformation for the ligand, followed by MD simulations.[90] For techniques dealing with receptor flexibility in particular, see Section 1.4.2.

1.4.1.2 Scoring Functions

Molecular docking programs predict binding conformations employing sampling algorithms, and their evaluation to estimate the energy of the ligand–target interaction is crucial. To this end, scoring functions are employed aiming to rank the complexes and discriminate correct poses from incorrect ones. Therefore, the design and proper use of scoring functions is of utmost importance in SBDD. Scoring functions can be classified into four

types: force field-based, empirical, knowledge-based and consensus scoring functions.[90,92–94,106–109]

1.4.1.2.1 Force Field-based Scoring Functions. Force field-based scoring functions estimates the binding energy using classic molecular mechanics formulations, calculating the sum of the non-bonded interactions (*i.e.* electrostatic and van der Waals terms). The electrostatic terms are calculated by a Coulombic formulation using a distance-dependent dielectric function to account for charge–charge interactions, whereas the van der Waals terms are usually described by a Lennard-Jones potential function (see Section 1.2.1). The parameters of the Lennard-Jones term can modify the 'hardness' of the potential, which in turns changes the distance between the receptor and ligand atoms. The most important limitations of force field-based scoring functions include the introduction of cut-off distances for the treatment of non-bonded interactions, which reduce the accuracy in calculating long-range effects involved in binding. In addition, force field scoring functions do not estimate entropic contributions and solvation energies.[110] The results of scoring with force field-based functions can be refined through calculation of linear interaction energy (LIE),[111,112] inclusion of generalised solvation through the Born model (MM/GBSA)[113] and free-energy perturbation methods (FEP).[114–117] Programs such as Gold,[102] Dock[96] and AutoDock[101] employ force field-based scoring functions.

1.4.1.2.2 Empirical Scoring Functions. Empirical scoring functions fit parameters to reproduce experimental data, such as binding energy, originally proposed by Böhm.[118,119] The binding energy is decomposed into several weighted terms, such as hydrogen bond, hydrophobic contact terms, desolvation energy, ionic interactions and binding entropy. The coefficients of each term are calculated from a regression analysis using experimental information from a training set of ligand–protein complexes with known binding affinities. Although the empirical scoring functions are simple to evaluate, the major drawback is their dependence on the training set employed and thus the transferability of the weighted parameters.[120–124] ChemScore[120] and FlexX[97] are examples of programs using empirical scoring functions.

1.4.1.2.3 Knowledge-based Scoring Functions. These functions are designed to reproduce experimentally determined complex structures. They are based on the assumption that the more favourable interatomic distances occur with higher frequency and the algorithms model those frequency distributions as pairwise atom-type potentials. The score is calculated as a sum of the individual interactions. The functions are computationally simple allowing for screening large compound databases, but as they rely on the training set employed for deriving the parameters an

extensive use is limited.[125–128] Knowledge-based potentials have been implemented in potential of mean force (PMF)[126,129–131] and DrugScore.[132]

1.4.1.2.4 Consensus Scoring Functions. A more recent strategy is the introduction of consensus scoring functions. This scoring scheme combines different scoring functions aiming to improve single scores and increase the ligand enrichment (*i.e.* the percentage of occurrence of strong binders among high-scoring ligands). However, when the employed scoring functions are significantly correlated, the method could magnify the calculation errors, rather than attenuate them.[133–135] CScore implements consensus scoring by combining Gold, DOCK, PMF, ChemScore and FlexX scoring functions.[136]

1.4.2 Structure Based Drug Design (SBDD)

SBDD makes use of high-throughput virtual screening (HTVS) techniques to search for bioactive compounds, identifying hits out of thousands of molecules by detecting complementarities between the ligand and the biological target. Selected compounds are ranked employing different scoring functions. Eventually the selected hits are experimentally evaluated to assay the biological activity on the selected target.[137,138] SBDD consists on the following key steps: (1) preparation of the target receptor, (2) compound database selection and preparation, and (3) molecular docking (*i.e.* determination of a favourable binding pose for each compound and ranking of the docked structures).

The first step involves the preparation of the target receptor, which indeed is of uttermost importance because the representation of the active site affects the quality of ligand posing and scoring. Experimentally determined structures of many receptors are available, mostly through X-ray crystallography and NMR spectroscopy. When the receptor structure is not experimentally available, it is possible to create a model starting from the sequence and applying homology modelling (see Section 1.3). Once the receptor model has been selected it has to be prepared for molecular docking studies, usually by adding hydrogen atoms, removing water molecules unless they bear important interactions, calculating partial charges and assigning tautomerisation states.[139] The initial selection of the structure is critical in the sense that small conformational changes arising from ligand binding highly influence the results, *e.g.* when using holo-, apo-proteins or homology models as targets.[140]

The second step is the selection and preparation of the small-molecule database. Several public databases containing millions of compounds and chemical information are freely accessible.[141,142] The most common chemical databases used in VHTS are ZINC,[143,144] PubChem,[145,146] DrugBank,[147,148] ChemSpider,[149,150] ChemBank,[151,152] eMolecules,[153] ChEMBL,[154,155] ChemDB,[156,157] and Binding DB.[158,159]

Preparing the original libraries requires different aspects depending on the software employed to perform the screening, but commonly involves the correct assignment of stereochemistry, partial charges and ionisation state according to the selected pH. In addition, several filters may be applied in order to enrich according to expected physicochemical properties of the potential ligands. Usually, filtering the database according to Lipinski's rule of five is performed to ensure drug-likeness.[160–163]

The third step involves docking experiments of the prepared small-molecule database into the prepared receptor binding site, and the analysis of the resulting docked conformations (Figure 1.7).

Exploration of ligand flexibility was described in previous sections of this chapter. The biological molecules are intrinsically mobile and consequently the representation of molecular flexibility of receptors is an important aspect of SBDD.[165] Incorporating receptor flexibility is a challenge in molecular docking due to the evident computational cost of modelling multiple degrees of freedom. Methods for accounting for receptor flexibility include the use of soft potentials (soft-docking), use of rotamer libraries, inclusion of side chain flexibility, and to perform ensemble docking.[166] Soft-docking decreases the van der Waals repulsion term energy in the scoring function allowing for partial overlapping between the receptor and the ligand. This method is simple but does not include suitable flexibility.[167] Employing rotamer libraries involve searching within the library to obtain possible conformations of the residue side chains. Even though it is efficient in terms of computation, it is highly dependent on the database used and ignores backbone flexibility.[168] Including side chain flexibility consist in sampling several side chains of the receptor simultaneously with the ligand sampling using for instance genetic algorithms. The main drawback is that only selected side chains are accounted for (the others are treated as rigid) and the backbone is not considered flexible. Finally, using an ensemble of protein conformations as obtained from *e.g.* NMR experiments allows for docking

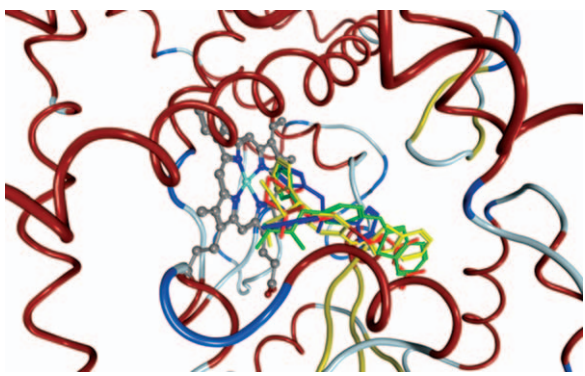

Figure 1.7 Structures of atRA (all-*trans* retinoic acid) in red, fluconazole in blue, R115866 in yellow and R116010 in green, docked in the active site of CYP26A1.[164]

the ligand into several receptor structures (*i.e.* different conformations), considering in each round a flexible ligand and the receptor as a rigid body. This method completely accounts for flexibility, but only for those protein conformations included in the sampling.[169,170] Even though several pitfalls of SBDD are well-known,[171,172] it has been successfully employed in identifying potent hits in many drug discovery studies.[173–181]

1.4.3 Ligand Based Drug Design (LBDD)

LBDD involves the analysis of ligands known to interact with the selected target. Several molecular descriptors are calculated for a set of reference compounds, *i.e.* compounds known to be active, which in turn are applied as molecular filters. Thus, the filtering is employed to select compounds sharing characteristics with the reference set. These methods do not require any information of the structure of the receptor. A distinct approach is the construction of a QSAR model predicting the biological activity from chemical structure.[182,183]

Different molecular descriptors can be calculated and the selected set depends on the biological function to be predicted. Molecular descriptors can be 2D (depending only on the topological connectivity) or 3D (depending on the geometry). 2D descriptors include physical properties such as atomic charges, polarisability, $\log P$ (logarithm of partition coefficient between *n*-octanol and water), solubility in water, volume, number of hydrogen bond donor/acceptor atoms, and molecular weight. 3D descriptors include properties such as total energy (and its components), ionisation potential, and HOMO/LUMO energy.[94,184–186]

After accounting for molecular descriptors, fingerprint techniques (similarity search) may be used to search databases for compounds similar in structure to a query (usually a lead compound).

Quantitative Structure–Activity Relationship (QSAR) models describe the mathematical relation between structural features and target response of a set of compounds.[187,188] The method involves the inclusion of active and inactive ligands, thus creating a set of mathematical descriptors. The subsequent step consists of the generation of a model establishing the relationship between those descriptors and the experimental biological activity of the compounds. Finally, the model is applied to predict the activity of compounds of interest.[94]

1.4.4 Pharmacophores

Pharmacophore models are of fundamental importance in drug design when no structural data is available. IUPAC have defined a pharmacophore as ‘the ensemble of steric and electronic features that is necessary to ensure the optimal supramolecular interactions with a specific biological target structure and to trigger (or to block) its biological response. The pharmacophore can be considered as the largest common denominator shared by a

set of active molecules'.[189] Thus, a pharmacophore is a collection of structural properties relevant for biological activity, a purely abstract concept, rather than a real molecule. Pharmacophoric descriptors include hydrogen-bond donors and acceptors, hydrophobic, aromatic, acidic, basic and ionisable groups.

A pharmacophore model can be ligand-generated, *i.e.* by superposing a set of bioactive molecules and extracting common chemical features responsible for the biological activity, or structure-based, *i.e.* by determining the main interactions between the target and the active ligands.[190] The latter can be obtained by analysing the structure of ligand–receptor complexes (either from crystal structures of from docking experiments), particularly the chemical features of the active site and the interactions with an active compound (Figure 1.8). The pharmacophore model must then fit the selected features (Figure 1.9).[191]

A more challenging problem is to generate a ligand-based pharmacophore model, and involves the following steps: (1) identifying the relevant properties, (2) superposing the molecules according to those properties, and (3) generating the pharmacophore model. The most demanding issue to address is the development of algorithms for effective molecular superposition, ensuring that a maximum number of chemical features overlap geometrically.[192,193] This in turn incorporates the problem of conformational flexibility, that can be addressed by the pre-enumerating method (multiple conformations of each molecule are included into a database), or by performing a conformational analysis during the pharmacophore modelling process as requested by the alignment algorithm (the so called on-the-fly method).[190,193] Once the ligands have been aligned, a pharmacophore feature map is extracted. A more general property definition increases the population of compounds matching the pharmacophore. This allows for identifying new compounds but also increasing the rate of false positives.[94,194]

Ligand-based pharmacophore modelling has become an essential computational strategy for drug discovery in the absence of structural information about the target. Several programs incorporate pharmacophore construction, such as Catalyst (part of Biovia Discovery Studio),[195,196] Phase[197,198] accessible by Schrödinger's graphical interface Maestro,[199] and MOE.[69,200]

1.4.5 Compound Optimisation

The last step of computational drug discovery involves the modification of the hits in order to improve the biological activity by changing the chemical structure, the hit-to-lead process. This optimisation involves increasing the drug potency (two- or three-fold), selectivity and pharmacokinetics, including absorption, distribution, metabolism, excretion and potential for toxicity (ADMET).[201,202]

In order to increase the biological potency of detected hits, similarity search employing pharmacophoric models is a valuable tool.[203]

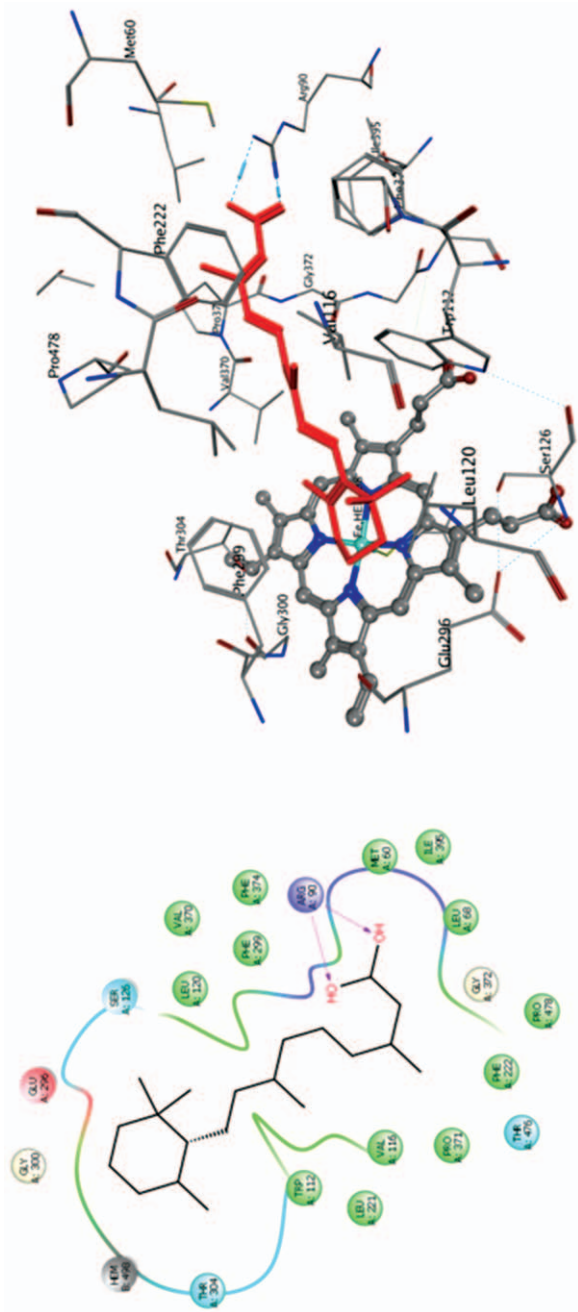

Figure 1.8 Main ligand interactions of atRA docked into the active site of CYP26A1, 2D (left) and 3D (right) diagrams.

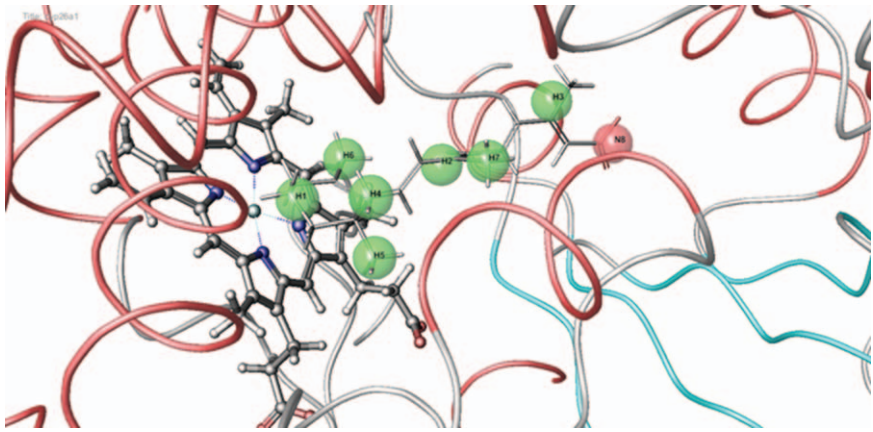

Figure 1.9 Pharmacophore model for atRA docked into the active site of CYP26A1. Green spheres indicate hydrophobic sites interactions, while the red sphere points out a required hydrogen bond acceptor in that position.

Focusing on ADMET properties, several filters can be employed to sort-out compound libraries, such as Lipinski's rule of five as mentioned in Section 1.4.2. This set of rules is related to those properties thought as necessary for good oral bioavailability and mainly targeting eukaryotic receptors.[204] However, when focusing for instance on antibiotics, the target might be located in the peptidoglycan matrix or the outer surface on the inner membrane. Then, permeation through the inner lipid membrane is not required to kill the pathogens and Lipinski's rules are simply not followed.[205] Therefore, different filtering rules may be needed depending on the particular biological target of interest. Moreover, several biologically active compounds violate more than one of the Lipinski's rules, such as atorvastatin (Lipitor®) and montelukast (Singulair®),[206] evidencing that automatic filtering might artificially remove potential leads.

Metabolic stability of drugs is a desirable property in the sense that when it is lowered, the drug diminishes its efficacy and increases the risk of generating toxic metabolites. Cytochrome P450 enzymes (CYPs) are major drug-metabolizing enzymes and the prediction of compounds that would be metabolised by, or inhibit CYPs must be assessed.[207,208] In terms of SBDD, the off-target prediction (focusing on available structures of CYPs) must be performed, aiming to determine the affinity of potential hits towards different receptors other than the main biological target. Besides accounting for metabolism, the off-target prediction may include any relevant human protein where inhibition would lead to toxic side effects.

1.4.6 Software and Web Based Servers

In previous sections of this chapter several programs were mentioned, describing the sampling algorithms, scoring functions and type of drug design

scheme included (SBDD or LBDD). Apart from those software packages, several web based servers for molecular docking and virtual screening are available. DOCK Blaster is a web server version of UCSF DOCK allowing for screening ZINC databases subsets.[209,210] SwissDock allow for ligand selection (ZINC ID, URL specification, an internal curated database, or an uploaded file).[211,212] DockThor is an online receptor-ligand docking facility allowing for uploading receptor, ligands and cofactor structures.[213,214] PharmMapper server is an integrated pharmacophore matching platform for potential target identification (off-target binding), very useful when predicting potential toxicity of developed hits.[215,216] These are some examples, many more web-based resources can be found.

Acknowledgements

Funding from the Swedish research council (LAE, AR), the Faculty of science at University of Gothenburg (LAE), the Wenner-Gren foundations (SG) and the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007–2013) under REA grant agreement no 608743 (PSM) are gratefully acknowledged.

References

1. A. Hinchliffe, *Molecular Modelling for Beginners*, Wiley & Sons, 2nd edn, 2008.
2. F. Jensen, *Introduction to Computational Chemistry*, Wiley & Sons, 2nd edn, 2007.
3. C. J. Cramer, *Essentials of Computational Chemistry: Theories and Models*, Wiley & Sons, 2nd edn, 2004.
4. A. M. Lesk, *Introduction to Bioinformatics*, Oxford University Press, 2002.
5. D. W. Mount, *Bioinformatics; Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press, 2nd edn, 2004.
6. J. Xiong, *Essential Bioinformatics*, Cambridge University Press, 2006.
7. D. Higgins and W. Taylor, *Bioinformatics: Sequence, Structure and Databanks*, Oxford University Press, 2000.
8. A. R. Leach, *Molecular Modelling: Principles and Applications*, Pearson, 2nd edn, 2001.
9. M. P. Allen, D. J. Tildesley, *Molecular Simulations of Liquids*, Oxford University Press, 1987.
10. *Drug Design: Structure- and Ligand-based Approaches*, ed. K. M. Merz Jr., D. Rigne and C. H. Reynolds, Cambridge University Press, 2010.
11. D. C. Young, *Computational Drug Design: A Guide for Computational and Medicinal Chemists*, Wiley & Sons, 2009.
12. T. A. Darden, D. York and L. Pedersen, *J. Chem. Phys.*, 1993, **98**, 10089–10092.
13. C. Sagui and T. A. Darden, *Annu. Rev. Biophys. Biomol. Struct.*, 1999, **28**, 155–179.

14. A. Pérez, I. Marchán, D. Svozil, J. Sponer, T. E. Cheatham, C. A. Laughton and M. Orozco, *Biophys. J.*, 2007, **92**, 3817–3829.
15. Y. Duan, *et al.*, *J. Comput. Chem.*, 2003, **24**, 1999–2012.
16. N. Foloppe and A. D. MacKerell, Jr., *J. Comput. Chem.*, 2000, **21**, 86–104.
17. C. Oostenbrink, A. Villa, A. E. Mark and W. F. van Gunsteren, *J. Comput. Chem.*, 2004, **25**, 1656–1676.
18. W. L. Jorgensen, D. S. Maxwell and J. Tirado-Rives, *J. Am. Chem. Soc.*, 1996, **118**, 11225–11236.
19. H. Sun, *J. Phys. Chem. B*, 1998, **102**, 7338–7364.
20. J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman and D. A. Case, *J. Comput. Chem.*, 2004, **25**, 1157–1174.
21. K. N. Kirschner, *et al.*, *J. Comput. Chem.*, 2008, **29**, 622–655.
22. L. Monticelli, D. P. Tieleman, *Methods in Molecular Biology*, Clifton, N.J., 2013, pp. 197–213.
23. S. J. Marrinck, H. J. Risselada, S. Yefimov, D. P. Tieleman and A. H. de Vries, *J. Phys. Chem. B*, 2007, **111**, 7812–7824.
24. S. J. Marrinck and D. P. Tieleman, *Chem. Soc. Rev.*, 2013, **42**, 6801.
25. T. Bereau and K. Kremer, *J. Chem. Theory Comput.*, 2015, **11**, 2783–2791.
26. S. Riniker and W. F. van Gunsteren, *J. Chem. Phys.*, 2011, **134**, 084110.
27. M. Orsi and J. W. Essex, *PLoS One*, 2011, **6**, e28637.
28. C. Mura and C. E. McAnany, *Mol. Simul.*, 2014, **40**, 732–764.
29. M. E. Tuckerman and G. J. Martyna, *J. Phys. Chem. B*, 2000, **104**, 159–178.
30. K. A. Feenstra, B. Hess and H. J. C. Berendsen, *J. Comput. Chem.*, 1999, **20**, 786–798.
31. P. H. Hünenberger, *Adv. Polym. Sci.*, 2005, **173**, 105–149.
32. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, *J. Chem. Phys.*, 1953, **21**, 1087.
33. W. K. Hastings, *Biometrika*, 1970, **57**, 97–109.
34. G. A. Ross, M. S. Bodnarchuk and J. W. Essex, *J. Am. Chem. Soc.*, 2015, **137**, 14930–14943.
35. G. King and A. Warshel, *J. Chem. Phys.*, 1989, **91**, 3647.
36. R. C. Bernardi, M. C. R. Melo and K. Schulten, *Biochim. Biophys. Acta, Gen. Subj.*, 2015, **1850**, 872–877.
37. V. Spiwok, Z. Sucur and P. Hosek, *Biotechnol. Adv.*, 2015, **33**, 1130–1140.
38. R. Lavery, M. Moakher, J. H. Maddocks, D. Petkeviciute and K. Zakrzewska, *Nucleic Acids Res.*, 2009, **37**, 5917–5927.
39. X.-J. Lu and W. K. Olson, *Nucleic Acids Res.*, 2003, **31**, 5108–5121.
40. M. Pasi, J. H. Maddocks and R. Lavery, *Nucleic Acids Res.*, 2015, **43**, 2412–2433.
41. D. Poger and A. E. Mark, *J. Chem. Theory Comput.*, 2010, **6**, 325–336.
42. J. Shao, S. W. Tanner, N. Thompson and T. E. Cheatham, *J. Chem. Theory Comput.*, 2007, **3**, 2312–2334.
43. D. van der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark and H. J. C. Berendsen, *J. Comput. Chem.*, 2005, **26**, 1701–1718.
44. D. A. Case, *et al.*, *J. Comput. Chem.*, 2005, **26**, 1668–1689.

45. B. A. Brooks, *et al.*, *J. Comput. Chem.*, 2009, **30**, 1545–1614.
46. M. Christen, *et al.*, *J. Comput. Chem.*, 2005, **26**, 1719–1751.
47. J. C. Phillips, *et al.*, *J. Comput. Chem.*, 2005, **26**, 1781–1802.
48. M. Bonomi, *et al.*, *Comput. Phys. Commun.*, 2009, **180**, 1961–1972.
49. W. Humphrey, A. Dalke and K. Schulten, *J. Mol. Graphics*, 1996, **14**, 33–38.
50. E. F. Pettersen, *et al.*, *J. Comput. Chem.*, 2004, **25**, 1605–1612.
51. D. R. Roe and T. E. Cheatham, *J. Chem. Theory Comput.*, 2013, **9**, 3084–3095.
52. P. L. Freddolino, A. S. Arkhipov, S. B. Larson, A. McPherson and K. Schulten, *Structure*, 2006, **14**, 437–449.
53. T. Reddy, *et al.*, *Structure*, 2015, **23**, 584–597.
54. D. E. Shaw, *et al.*, *Science*, 2010, **330**, 341–346.
55. V. S. Pande, K. Beuchamp and G. R. Bowman, *Methods*, 2010, **52**, 99–105.
56. G. Jayachandran, V. Vishal and V. S. Pande, *J. Chem. Phys.*, 2006, **124**, 164902.
57. M. W. van der Kamp and A. J. Mulholland, *Biochemistry*, 2013, **52**, 2708–2728.
58. H. M. Senn and W. Thiel, *Curr. Opin. Chem. Biol.*, 2007, **11**, 182–187.
59. W. L. Jorgensen, *Acc. Chem. Res.*, 2009, **42**, 724–733.
60. S. F. Altschul, W. Gish, W. Miller, E. Myers and D. J. Lipman, *J. Mol. Biol.*, 1990, **215**, 403–410.
61. D. J. Lipman and W. R. Pearson, *Science*, 1985, **227**, 1435–1441.
62. S. Henikoff and J. G. Henikoff, *Proc. Natl. Acad. Sci. U. S. A.*, 1992, **89**, 10915–10919.
63. S. F. Altschul, *et al.*, *Nucleic Acids Res.*, 1997, **25**, 3389–3402.
64. A. A. Schaffer, *et al.*, *Nucleic Acids Res.*, 2001, **29**, 2994–3005.
65. J. D. Thompson, D. G. Higgins and T. J. Gibson, *Nucleic Acids Res.*, 1994, **22**, 4673–4680.
66. J. Yang, R. Yan, A. Roy, D. Xu, J. Poisson and Y. Zhang, *Nat. Methods*, 2015, **12**, 7–8.
67. S. Wu and Y. Zhang, *Nucleic Acids Res.*, 2007, **35**, 3375–3382.
68. A. Fiser and A. Sali, *Methods Enzymol.*, 2003, **374**, 461–491.
69. Montreal, Quebec, MOE 2015:10 edn, 2016.
70. L. A. Kelley, *et al.*, *Nat. Protoc.*, 2015, **10**, 845–858.
71. M. Källberg, H. Wang, S. Wang, J. Peng, Z. Wang, H. Lu and J. Xu, *Nat. Protoc.*, 2012, **7**, 1511–1522.
72. A. Leaver-Fay, *et al.*, *Methods Enzymol.*, 2011, **487**, 545–574.
73. M. Biasini, *et al.*, *Nucleic Acids Res.*, 2014, **42**(W1), W252–W258.
74. E. Krieger and G. Vriend, *Bioinformatics*, 2014, **30**, 2981–2982.
75. R. D. Finn, *et al.*, *Nucleic Acids Res.*, 2016, **D44**, D279–D285.
76. H. Fang and J. Gough, *Nucleic Acids Res.*, 2013, **D41**, D536–D544.
77. C. J. A. Sigrist, *et al.*, *Nucleic Acids Res.*, 2013, **D41**, D344–D347.
78. Y. Tang, W. Zhu, K. Chen and H. Jiang, *Drug Discovery Today Technol.*, 2006, **3**, 307–313.

79. S. C. Basak, *Curr. Comput.-Aided Drug Des.*, 2012, **8**, 1–2.
80. Forbes, *Crisis In Pharma R&D: It Costs $2.6 Billion To Develop A New Medicine; 2.5 Times More Than in 2003*, http://www.forbes.com/sites/theapothecary/2014/11/26/crisis-in-pharma-rd-it-costs-2-6-billion-to-develop-a-new-medicine-2-5-times-more-than-in-2003/#239cbbce1641.
81. C. M. Song, S. J. Lim and J. C. Tong, *Briefings Bioinf.*, 2009, **10**, 579–591.
82. I. M. Kapetanovic, *Chem.-Biol. Interact.*, 2008, **171**, 165–176.
83. B. C. Duffy, L. Zhu, H. Decornez and D. B. Kitchen, *Bioorg. Med. Chem.*, 2012, **20**, 5324–5342.
84. J. H. Van Drie, *J. Comput.-Aided Mol. Des.*, 2007, **21**, 591–601.
85. T. N. Doman, S. L. McGovern, B. J. Witherbee, T. P. Kasten, R. Kurumbail, W. C. Stallings, D. T. Connolly and B. K. Shoichet, *J. Med. Chem.*, 2002, **45**, 2213–2221.
86. D. M. Kruger and A. Evers, *ChemMedChem*, 2010, **5**, 148–158.
87. Y. Westermaier, X. Barril and L. Scapozza, *Methods*, 2015, **71**, 44–57.
88. S. Kalyaanamoorthy and Y. P. Chen, *Drug Discovery Today*, 2011, **16**, 831–839.
89. I. Halperin, B. Ma, H. Wolfson and R. Nussinov, *Proteins*, 2002, **47**, 409–443.
90. X.-Y. Meng, H.-X. Zhang, M. Mezei and M. Cui, *Curr. Comput.-Aided Drug Des.*, 2011, **7**, 146–157.
91. J. P. Changeux and S. Edelstein, *F1000 Biol. Rep.*, 2011, **3**, 19.
92. D. B. Kitchen, H. Decornez, J. R. Furr and J. Bajorath, *Nat. Rev. Drug Discovery*, 2004, **3**, 935–949.
93. L. G. Ferreira, R. N. Dos Santos, G. Oliva and A. D. Andricopulo, *Molecules*, 2015, **20**, 13384–13421.
94. G. Sliwoski, S. Kothiwale, J. Meiler and E. W. Lowe, Jr., *Pharmacol. Rev.*, 2014, **66**, 334–395.
95. Z. Zsoldos, D. Reid, A. Simon, S. B. Sadjad and A. P. Johnson, *J. Mol. Graphics Modell.*, 2007, **26**, 198–212.
96. T. J. A. Ewing, S. Makino, A. G. Skillman and I. D. Kuntz, *J. Comput.-Aided Mol. Des.*, 2001, **15**, 411–428.
97. M. Rarey, B. Kramer, T. Lengauer and G. Klebe, *J. Mol. Biol.*, 1996, **261**, 470–489.
98. R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, D. E. Shaw, P. Francis and P. S. Shenkin, *J. Med. Chem.*, 2004, **47**, 1739–1749.
99. D. S. Goodsell and A. J. Olson, *Proteins*, 1990, **8**, 195–202.
100. E. M. Krovat, T. Steindl and T. Langer, *Curr. Comput.-Aided Drug Des.*, 2005, **1**, 93–102.
101. G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew and A. J. Olson, *J. Comput. Chem.*, 1998, **19**, 1639–1662.
102. M. L. Vendonk, J. C. Cole, M. J. Hartshorn, C. W. Murray and R. D. Taylor, *Proteins*, 2003, **52**, 609–623.
103. F. Glover, *Comput. Oper. Res.*, 1986, **13**, 533–549.

104. C. A. Baxter, C. W. Murray, D. E. Clark, D. R. Westhead and M. D. Eldridge, *Proteins*, 1998, **33**, 367–382.
105. D. R. Westhead, D. E. Clark and C. W. Murray, *J. Comput.-Aided Mol. Des.*, 1997, **11**, 209–228.
106. A. R. Leach, B. K. Shoichet and C. E. Peishof, *J. Med. Chem.*, 2006, **49**, 5851–5855.
107. P. L. Kastritis and A. M. J. J. Bonvin, *J. Proteome Res.*, 2010, **9**, 2216–2225.
108. R. T. Kroemer, *Curr. Protein Pept. Sci.*, 2007, **8**, 312–328.
109. S.-Y. Huang, S. Z. Grinter and X. Zou, *Phys. Chem. Chem. Phys.*, 2010, **12**, 12899–12908.
110. P. Englebienne and N. Moitessier, *J. Chem. Inf. Model.*, 2009, **49**, 2564–2571.
111. J. Michel, M. L. Verdonk and J. W. Essex, *J. Med. Chem.*, 2006, **49**, 7424–7439.
112. I. D. Wall, A. R. Leach, D. W. Salt, M. G. Ford and J. W. Essex, *J. Med. Chem.*, 1999, **42**, 5142–5152.
113. V. Tsui and D. Case, *Biopolymers*, 2000, **56**, 275–291.
114. P. Kollman, *Chem. Rev.*, 1993, **93**, 2395–2417.
115. J. M. Briggs, T. J. Marrone and J. A. McCammon, *Trends Cardiovasc. Med.*, 1996, **6**, 198–206.
116. H. A. Carlson and W. L. Jorgensen, *J. Phys. Chem.*, 1995, **99**, 10667–10673.
117. J. Aqvist, V. B. Luzhkov and B. O. Brandsdal, *Acc. Chem. Res.*, 2002, **35**, 358–365.
118. H. J. Böhm, *J. Comput.-Aided Mol. Des.*, 1992, **6**, 593–606.
119. H. J. Böhm, *J. Comput.-Aided Mol. Des.*, 1998, **12**, 309–323.
120. M. D. Eldridge, C. W. Murray, T. R. Auton, G. V. Paolini and R. P. Mee, *J. Comput.-Aided Mol. Des.*, 1997, **11**, 425–445.
121. G. M. Verkhivker, D. Bouzida, D. K. Gehlhaar, P. A. Rejto, S. Arthurs, A. B. Colson, S. T. Freer, V. Larson, B. A. Luty, T. Marrone and P. W. Rose, *J. Comput.-Aided Mol. Des.*, 2000, **14**, 731–751.
122. A. N. Jain, *J. Comput.-Aided Mol. Des.*, 1996, **10**, 427–440.
123. R. D. Head, M. L. Smythe, T. I. Oprea, C. L. Waller, S. M. Green and G. R. Marshall, *J. Am. Chem. Soc.*, 1996, **118**, 3959–3969.
124. C. W. Murray, T. R. Auton and M. D. Eldridge, *J. Comput.-Aided Mol. Des.*, 1998, **12**, 503–519.
125. G. Verkhivker, K. Appelt, S. T. Freer and J. E. Villafranca, *Protein Eng.*, 1995, **8**, 677–691.
126. I. Muegge and Y. C. Martin, *J. Med. Chem.*, 1999, **72**, 791–804.
127. H. Gohlke, M. Hendlich and G. Klebe, *J. Mol. Biol.*, 2000, **295**, 337–356.
128. S.-Y. Huang and X. Zou, *J. Comput. Chem.*, 2006, **27**, 1866–1875.
129. J. Shimada, A. V. Ishchenko and E. I. Shakhnovich, *Protein Sci.*, 2000, **9**, 765–775.
130. I. Muegge, *Perspect. Drug Discovery Des.*, 2000, **20**, 99–114.
131. I. Muegge, *J. Comput. Chem.*, 2001, **22**, 418–425.

132. H. F. Velec, H. Gohlke and G. Klebe, *J. Med. Chem.*, 2005, **48**, 6296–6303.
133. P. S. Charifson, J. J. Corkery, M. A. Murcko and W. P. Walters, *J. Med. Chem.*, 1999, **42**, 5100–5109.
134. M. Feher, *Drug Discovery Today*, 2006, **11**, 421–428.
135. N. M. O'Boyle, J. W. Liebeschuetz and J. C. Cole, *J. Chem. Inf. Model.*, 2009, **49**, 1871–1878.
136. R. D. Clark, A. Strizhev, J. M. Leonard, J. F. Blake and J. B. Matthew, *J. Mol. Graphics Modell.*, 2002, **20**, 281–295.
137. E. Lionta, G. Spyrou, D. K. Vassilatis and Z. Cournia, *Curr. Top. Med. Chem.*, 2014, **14**, 1923–1938.
138. R. P. Gangwal, M. V. Damre, N. R. Das, G. V. Dhoke, A. Bhadauriya, R. A. Varikoti, S. S. Sharma and A. T. Sangamwar, *J. Mol. Graphics Modell.*, 2015, **57**, 89–98.
139. A. N. Jain and A. Nicholls, *J. Comput.-Aided Mol. Des.*, 2008, **22**, 133–139.
140. S. L. McGovern and B. K. Shoichet, *J. Med. Chem.*, 2003, **46**, 2895–2907.
141. A. J. M. Barbosa and A. Del Rio, *Curr. Top. Med. Chem.*, 2012, **12**, 866–877.
142. S. O. Jonsdottir, F. S. Jorgensen and S. Brunak, *Bioinformatics*, 2005, **21**, 2145–2160.
143. ZINC, http://zinc.docking.org/.
144. J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad and R. G. Coleman, *J. Chem. Inf. Model.*, 2012, **52**, 1757–1768.
145. PubChem, https://pubchem.ncbi.nlm.nih.gov/.
146. Q. Li, T. Cheng, Y. Wang and S. H. Bryant, *Drug Discovery Today*, 2010, **15**, 1052–1057.
147. DrugBank, http://www.drugbank.ca/.
148. D. S. Wishart, *Nucleic Acids Res.*, 2006, **34**, D668–D672.
149. ChemSpider, http://www.chemspider.com/.
150. H. E. Pence and A. Williams, *J. Chem. Educ.*, 2010, **87**, 1123–1124.
151. ChemBank, http://chembank.broadinstitute.org/.
152. K. P. Seiler, G. A. George, M. P. Happ, N. E. Bodycombe, H. A. Carrinski, S. Norton, S. Brudz, J. P. Sullivan, J. Muhlich, M. Serrano, P. Ferraiolo, N. J. Tolliday, S. L. Schreiber and P. A. Clemons, *Nucleic Acids Res.*, 2007, **36**, D351–D359.
153. eMolecules, https://www.emolecules.com/.
154. ChEMBL, https://www.ebi.ac.uk/chembl/.
155. A. P. Bento, A. Gaulton, A. Hersey, L. J. Bellis, J. Chambers, M. Davies, F. A. Krüger, Y. Light, L. Mak, S. McGlinchey, M. Nowotka, G. Papadatos, R. Santos and J. P. Overington, *Nucleic Acids Res.*, 2014, **42**, D1083–D1090.
156. ChemDB, http://chemdb.ics.uci.edu/.
157. J. H. Chen, E. Linstead, S. J. Swamidass, D. Wang and P. Baldi, *Bioinformatics*, 2007, **23**, 2348–2351.
158. BindingDB, https://www.bindingdb.org/bind/index.jsp.
159. T. Liu, Y. Lin, X. Wen, R. N. Jorissen and M. K. Gilson, *Nucleic Acids Res.*, 2007, **35**, D198–D201.

160. C. A. Lipinski, F. Lombardo, B. W. Dominy and P. J. Feeney, *Adv. Drug Delivery Rev.*, 1997, **23**, 3–25.
161. C. A. Lipinski, F. Lombardo, B. W. Dominy and P. J. Feeney, *Adv. Drug Delivery Rev.*, 2001, **46**, 3–26.
162. C. A. Lipinski, F. Lombardo, B. W. Dominy and P. J. Feeney, *Adv. Drug Delivery Rev.*, 2012, **64**, 4–17.
163. C. A. Lipinski, *Adv. Drug Delivery Rev.*, 2016, 33–41.
164. M. K. A. Awadallah, T. Alshammari, L. A. Eriksson and P. Saenz-Mendez, *Molecules*, 2016, **21**, 351.
165. S. J. Teague, *Nat. Rev. Drug Discovery*, 2003, **2**, 527–541.
166. C. B-Rao, J. Subramanian and S. D. Sharma, *Drug Discovery Today*, 2009, **14**, 394–400.
167. F. Jiang and S.-H. Kim, *J. Mol. Biol.*, 1991, **219**, 79–102.
168. A. R. Leach, *J. Mol. Biol.*, 1994, **235**, 345–356.
169. R. M. A. Knegtel, I. D. Kuntz and C. M. Oshiro, *J. Mol. Biol.*, 1997, **266**, 424–440.
170. C. N. Cavasotto and R. A. Abagyan, *J. Mol. Biol.*, 2004, **337**, 209–225.
171. T. Scior, A. Bender, G. Tresadern, J. L. Medina-Franco, K. Martínez-Mayorga, T. Langer, K. Cuanalo-Contreras and D. K. Agrafiotis, *J. Chem. Inf. Model.*, 2012, **52**, 867–881.
172. J. Kirchmair, P. Markt, S. Distinto, G. Wolber and T. Langer, *J. Comput.-Aided Mol. Des.*, 2008, **22**, 213–228.
173. O. M. Becker, D. S. Dhanoa, Y. Marantz, D. Chen, S. Shachman, S. Cheruku, A. Heifetz, P. Mohanty, M. Fichman, A. Sharadendu, R. Nudelman, M. Kauffman and S. Noiman, *J. Med. Chem.*, 2006, **49**, 3116–3135.
174. F. M. Ruiz, R. Gil-Redondo, A. Morreale, A. R. Ortiz, C. Fábrega and J. Bravo, *J. Chem. Inf. Model.*, 2008, **48**, 844–854.
175. N. Li, F. Wang, S. Niu, J. Cao, K. Wu, Y. Li, N. Yin, X. Zhang, W. Zhu and Y. Yin, *BMC Microbiol.*, 2009, **9**, 129.
176. K. J. Simmons, I. Chopra and C. W. G. Fishwick, *Nat. Rev. Microbiol.*, 2010, **8**, 501–510.
177. I. Pauli, R. N. dos Santos, D. C. Rostirolla, L. K. Martinelli, R. G. Ducati, L. F. S. M. Timmers, L. A. Basso, D. S. Santos, R. V. C. Guido, A. D. Andricopulo and O. Norberto de Souza, *J. Chem. Inf. Model.*, 2013, **53**, 2390–2401.
178. Z. Miller, K.-S. Kim, D.-M. Lee, V. Kasam, S. E. Baek, K. H. Lee, Y.-Y. Zhang, L. Ao, K. Carmony, N.-R. Lee, S. Zhou, Q. Zhao, Y. Jang, H.-Y. Jeong, C.-G. Zhan, W. Lee, D.-E. Kim and K. B. Kim, *J. Med. Chem.*, 2015, **58**, 2036–2041.
179. K. Matsuno, Y. Masuda, Y. Uehara, H. Sato, A. Muroya, O. Takahashi, T. Yokotagawa, T. Furuya, T. Okawara, M. Otsuka, N. Ogo, T. Ashizawa, C. Oshita, S. Tai, H. Ishii, Y. Akiyama and A. Asai, *ACS Med. Chem. Lett.*, 2010, **1**, 371–375.
180. L. Wang, Q. Gu, X. Zheng, J. Ye, Z. Liu, J. Li, X. Hu, A. Hagler and J. Xu, *J. Chem. Inf. Model.*, 2013, **53**, 2409–2422.

181. S. Dadashpour, T. Tuylu Kucukkilinc, O. Unsal Tan, K. Ozadali, H. Irannejad and S. Emami, *Arch. Pharm.*, 2015, **348**, 179–187.
182. H. Geppert, M. Vogt and J. Bajorath, *J. Chem. Inf. Model.*, 2010, **50**, 205–216.
183. P. Ripphausen, B. Nisius and J. Bajorath, *Drug Discovery Today*, 2011, **16**, 372–376.
184. H. Eckert and J. Bajorath, *Drug Discovery Today*, 2007, **12**, 225–233.
185. P. Willet, *J. Chem. Inf. Comput. Sci.*, 1998, **38**, 983–996.
186. A. Bender and R. C. Glen, *Org. Biomol. Chem.*, 2004, **2**, 3204.
187. A. R. Katritzky and E. V. Gordeeva, *J. Chem. Inf. Comput. Sci.*, 1993, **33**, 835–857.
188. S. Zhang, *Drug Des. Discovery*, 2011, **716**, 23–38.
189. IUPAC, 1998, http://www.chem.qmul.ac.uk/iupac/medchem/ix.html#p7.
190. S.-Y. Yang, *Drug Discovery Today*, 2010, **15**, 444–450.
191. J. Zou, H.-Z. Xie, S.-Y. Yang, J.-J. Chen, J.-X. Ren and Y.-Q. Wei, *J. Mol. Graphics Modell.*, 2008, **27**, 430–438.
192. G. M. Spitzer, M. Heiss, M. Mangold, P. Markt, J. Kirchmair, G. Wolber and K. R. Liedl, *J. Chem. Inf. Model.*, 2010, **50**, 1241–1247.
193. G. Wolber, *Drug Discovery Today*, 2008, **13**, 23–29.
194. C. Acharya, A. Coop, J. E. Polli and A. D. MacKerell, *Curr. Comput.-Aided Drug Des.*, 2011, **7**, 10–22.
195. Catalyst, http://accelrys.com/products/collaborative-science/biovia-discovery-studio/pharmacophore-and-ligand-based-design.html.
196. Dassault-Systèmes, *BIOVIA, Discovery Studio Modeling Environment, San Diego*, 2016.
197. Phase, *4.4, Schrödinger LLC, New York*, 2016.
198. S. L. Dixon, A. M. Smondyrev, E. H. Knoll, S. N. Rao, D. E. Shaw and R. A. Friesner, *J. Comput.-Aided Mol. Des.*, 2006, **20**, 647–671.
199. Maestro, *10.3, Schrödinger LLC, New York*, 2016.
200. P. Labute, C. Williams, M. Feher, E. Sourial and J. M. Schmidt, *J. Med. Chem.*, 2001, **44**, 1483–1490.
201. I. Poggesi, P. S. Burton, J. T. Goodwin and M. Germani, in *Optimizing the ''Drug-Like'' Properties of Leads in Drug Discovery*, ed. R. T. Borchardt, E. H. Kerns, M. J. Hageman, D. R. Thakker and J. L. Stevens, Springer, New York, 2006, pp. 195–220.
202. C. Manly, J. Chandrasekhar, J. Ochterski, J. Hammer and B. Warfield, *Drug Discovery Today*, 2008, **13**, 99–109.
203. K. H. Bleicher, H.-J. Böhm, K. Müller and A. I. Alanine, *Nat. Rev. Drug Discovery*, 2003, **2**, 369–378.
204. M. S. Lajiness, M. Vieth and J. Erickson, *Curr. Opin. Drug Discovery Dev.*, 2004, **7**, 470–477.
205. R. O'Shea and H. E. Moser, *J. Med. Chem.*, 2008, **51**, 2871–2878.
206. R. G. Bickerton, G. V. Paolini, J. Besnard, S. Muresan and A. L. Hopkins, *Nat. Chem.*, 2012, **4**, 90–98.
207. C. J. Omicienski, R. P. Remmel and V. P. Hosagrahara, *Toxicol. Sci.*, 1999, **48**, 151–156.

208. H. van de Waterbeemd and E. Gifford, *Nat. Rev. Drug Discovery*, 2003, **2**, 192–204.
209. DockBlaster, http://blaster.docking.org/.
210. J. J. Irwin, B. K. Shoichet, M. M. Mysinger, N. Huang, F. Colizzi, P. Wassam and Y. Cao, *J. Med. Chem.*, 2009, **52**, 5712–5720.
211. SwissDock, http://www.swissdock.ch/.
212. A. Grosdidier, V. Zoete and O. Michielin, *Nucleic Acids Res.*, 2011, **39**, W270–W277.
213. DockThor, http://dockthor.lncc.br/index.php?pg=home.
214. C. S. de Magalhães, D. M. Almeida, H. J. C. Barbosa and L. E. Dardenne, *Inf. Sci.*, 2014, **289**, 206–224.
215. PharmMapper, http://lilab.ecust.edu.cn/pharmmapper/.
216. X. Liu, S. Ouyang, B. Yu, Y. Liu, K. Huang, J. Gong, S. Zheng, Z. Li, H. Li and H. Jiang, *Nucleic Acids Res.*, 2010, **38**, W609–W614.