Developing novel structural bioinformatics techniques is a particularly challenging task, not only due to the complexity of a full three-dimensional description of biomolecular systems, but also owing to the large amount of coding required to solve recurring tasks, like the reading and writing of molecular file formats, handling of efficient molecular data structures, and evaluation of established force fields. Implementing the whole range of functionality one would like to apply to current fields of interest is often infeasible. While this problem can be partially adressed by combining several external applications into a pipeline, an alternative approach, resulting in greatly reduced development times and typically more stable applications, consists in using application development frameworks, like our Biochemical ALgorithms Library **BALL** (Kohlbacher et al., Bioinformatics, 16(9):815-824). **BALL** is a freely available, open source C++ library, composed of approximately 500 classes covering a wide range of molecular data structures and algorithms. In this work, we want to exemplarily demonstrate **BALL**'s rapid prototyping capabilities by implementing and extending – in the short time span of a master's thesis – an established complex docking algorithm: the 'Stochastic Roadmap Simulation' (SRS) approach (M. S. Apaydin et al., J. Comp. Biol., (10) 257-281.)

SRS samples the ligand's path through conformational space during the binding process to estimate the kinetics of association. To this end, distinct conformations are generated by randomly perturbing the ligand's degrees of freedom. Similarity relationships between the resulting conformations are represented in a graph, where each node corresponds to one conformation. Edges in the graph represent energetically feasible transitions in configuration space between reasonably close conformations and are annotated with transition probabilities according to Boltzmann's law. The algorithm proceeds to compute the expected number of Monte Carlo steps required to move from a certain conformation to all of its neighbours and propagates this information along paths in the graph. Finally, binding rate constants are computed from the average number of steps required to enter or leave the bound state.

Implementing SRS requires a considerable number of molecular data structures and algorithms, and can hence be significantly simplified by using the **BALL** library. In addition, using **BALL** allows us to supplement the algorithm with further functionality. For example, we can easily make use of all different force fields (like Amber, MMFF94, or CHARMM) and the finite difference Poisson-Boltzmann solver implemented in **BALL** to compute internal and intermolecular energies. In our implementation, we use these capabilities to perform a short energy minimization of near hydrogen atoms, yielding improved sampling properties close to the receptor.

To provide non-specialists with an intuitive GUI, our implementation will be fully integrated into **BALL**'s molecular modelling tool **BALLView** (Moll et al., Bioinformatics 22(3):365366). Due to **BALLView**'s modular design, this can be achieved without the need to implement any GUI related functionality, or to convert between different molecular representations.

In our experience, realizing as complex a project as the implementation of the SRS approach in a master's thesis without relying on a framework like **BALL** is highly challenging at least. In contrast, using **BALL**, we were able to implement the basic functionality of SRS in a few weeks only. This allows us to develop and evaluate even complex extensions of the algorithm, like the mentioned coupling to local minimization techniques, in the course of the thesis. We thus believe that the presented project demonstrates **BALL**'s abilities to significantly reduce development times and to give easy access to advanced molecular modelling features.