Haptic-driven, interactive drug design: implementing a GPU-based approach to evaluate the induced fit effect.

Athanasios Anthopoulos[a,b], **Gaia Pasqualetto**[a], **Ian Grimstead**[b] **and Andrea Brancale**[a*]

[a]School of Pharmacy and Pharmaceutical Sciences, Cardiff University, Cardiff, CF10 3NB, UK

[b]School of Computer Science, Cardiff University, Cardiff, UK, CF24 3AA

* brancalea@cardiff.ac.uk

Abstract: In this paper, we describe a hybrid meta-heuristics method of energy minimization and conformational sampling and its application into our haptic-driven molecular modelling simulator. The proposed method has been designed to suit real-time molecular docking simulations, where the time-lapse between two successive ligand poses is relatively small. In these situations the energy minimization problem becomes increasingly complex and chaotic. The algorithm is tuned to take advantage of recent advances in GPU computing with asynchronous kernel execution, which has allowed us to include full protein flexibility in the real-time interactive, haptic-driven simulations. Finally, in this paper we will also discuss the implementation of such high-performance computing approach in our software, discussing the results of our initial validation studies, highlighting the advantages and limitations of such interactive methodology.

1. Introduction

The assessment of protein flexibility upon ligand binding remains one of the most challenging aspects of computer-aided drug design. Indeed, in the last decade, several methodologies that attempt to mimic the induced-fit effect have been developed and implemented in several software packages, from the simple exploration of rotamers of predefined amino acid side chains to the prediction of backbone flexibility using short molecular dynamics runs.[1] In all cases, molecular conformations are updated using the appropriate algorithm, depending on the degree of flexibility that the application offers.[2] The objective of finding stable (or preferred) conformations for the system requires locating those conformations that lie on minimum points on the energy surface.[1] However, this is a complex multi-dimensional problem. For a system with N atoms, its potential energy is a function of 3N Cartesian co-ordinates. Therefore, the molecular energy surface is rather chaotic and searching for the global minimum cannot always be feasible. Failing that, exploring the potential energy surface is the way to find a set of optimal solutions (local minimums) where the best solution can be chosen. The search for optimal solutions can be facilitated with the use of meta-heuristics.[3]

Meta-heuristic methods cover a wide spectrum of algorithmic techniques. Evolutionary algorithms (EA) and genetic algorithms (GA) are methods able to iteratively generate a set of solutions. Evolutionary algorithms are amongst the most popular meta-heuristics and attempt to solve the optimization problem by generating a population of possible solutions from a parent (mutation). From the generated population the fittest is chosen to generate the new pool of solutions and so forth. Genetic algorithms adopt a similar approach, with the addition of the concept of crossover between two solutions to create the new generation.[4] These solution generations will be gradually improving with respect to the strategy adopted to move across the energy landscape.

The exploration of potential energy surfaces requires the adoption of uphill moves in order to recover from local minima. Such strategies for uphill move selection include simulated annealing (SA), Monte-Carlo (MC), iterated local search (ILS), guided local search (GLS) and many more. Simulated annealing uses a temperature variable to generate a probability function for accepting solutions that go uphill on the energy surface.[5] Monte Carlo methods adopt a similar approach to escape local minima by accepting uphill moves based on a probability function such as the Metropolis criterion.[6] ILS reaches optimal solutions by iteratively performing local search to reach a local minimum;[7] the way it escapes a local minimum is by perturbing the discovered solution, in hope that the perturbation mechanism will bring the system into a well with a lower minimum than the parent one. GLS escapes from local minima by gradually increasing the global minimum energy value. This way, the system will escape from a well if it is trapped, by taking an uphill move close to the minimum, which will result into a new landscape with the hope that a better minimum solution will be discovered.

Various meta-heuristics have been used by many recent molecular docking applications. Autodock-Vina used a hybrid ILS strategy using the Metropolis criterion to accept optimal solutions and predict binding affinity of protein-ligand systems.[8] Variations of the genetic algorithm have been implemented in various published works to predict molecular structures.[9,10] Soares used an evolutionary strategy to solve this problem[11] and Cai and Shao used a hybrid method of EA combined with SA for structure prediction.[12] ParadisEO approached the docking problem with a parallel GA in order to leverage additional processing power and improve its performance and efficiency.[13]

In our group we are developing an interactive molecular docking program, which allows the user to place, in real time, a small molecule in a biological target, while feeling the molecular interactions it establishes through a haptic device.[14,15] In our original system the target structure was kept rigid, while the ligand was kept flexible and the system underwent continuous energy minimisation, via a steepest descent algorithm. Recently we have explored the possibility of including a degree of protein flexibility in the interactive simulation, by taking advantage of the computational power of GPGPUs (General-Purpose Graphics Processing Unit).[16] The main challenge in our approach is that if we want to achieve a level of interactivity that provides the user with a natural experience, our algorithm should be able to obtain a stable, low energy, protein/ligand conformation within a restricted timeframe. In our software, the ligand receives a signal to move at fixed time intervals t = 33 ms, which matches the rendering rate on screen. When the signal is received the ligand atoms are translated into their new position. On the completion of ligand translation, the aim is to provide the system (ligand and protein) with the best atomic structure when the time-lapse t expires and before the new haptic pointer signals to translate the ligand to its next position (minimization cycle).

In this paper, we present a hybrid meta-heuristic method of exploring molecular energy surfaces with a focus of solving this problem as a set of independent protein ligand poses generated in a restricted timeframe.

2. Results and Discussion

2.1 CUDA Streams

The recent generation of Kepler cards (GK110) has introduced new features in GPU computing, including the ability to run up to 32 kernels asynchronously achieving execution overlaps very close to that of a parallel execution. This introduces an additional dimension of parallelism in CUDA and initiates the need of new optimization techniques in CUDA-parallel programming. To be more specific, considering this new feature, an algorithm designer should not treat each kernel individually any more, but as part of a wider context able to utilize the graphics card more efficiently. However, there are some rules and constraints on the way that GPUs parallelize kernel execution: at the present time, NVIDIA GPUs are able to run 16 scheduled blocks per streaming multiprocessor (SM). In the case of a K20 card, that gives a total of 208 blocks able to run concurrently at any time in its 13 SMs. From a resources perspective, the shared memory is limited at 48kb, so asynchronously scheduled kernels in each stream must not exceed the this limit. If it does, then the parallel execution is restricted to the number of streams whose total shared memory usage is below the 48kb limit. Regarding register usage, similar restrictions apply and if the total number of registers used per thread exceeds the card's limit (255 for GK110s) then some register overspill might take place and the slower performing local memory might be used. Finally, overlapping streams can only write on independent memory addresses; this is not a major problem as arrays can be copied in multiple buffers for each stream to operate on independently.

2.2 Potential Energy and Force Calculations

The energy (1) and force gradients (2) computations to facilitate the minima search and conformation updates for our proposed method were performed using the MMFF94s force-field and implemented in CUDA.

$$\mathrm{E} = \sum \mathrm{E}_{\mathrm{bonded}} + \sum \mathrm{E}_{\mathrm{non-bonded}} \qquad (1)$$

$$\vec{\mathrm{F}_{\mathrm{i}}} = -\nabla \mathrm{E}_{\mathrm{i}} \qquad (2)$$

Regarding the non-bonded interactions, atoms were geometrically mapped (binned) into a 3D load-balanced irregular grid.[16,17] Calculations then took place with each CUDA block responsible for processing the atomic interactions between each home cell and its neighbours.[16]

Once the force vector is available for each atom i, a scalar value (step-value) needs to be found in order to update the system into the new conformation. The conformational update is performed for each atom in the system in vector space as follows:

$$\overrightarrow{\mathrm{C'}_{\mathrm{i}}} = \vec{\mathrm{C}_{\mathrm{i}}} + \lambda \vec{\mathrm{F}_{\mathrm{i}}} \qquad (3)$$

Where $\overrightarrow{\mathrm{C'}_{\mathrm{i}}}$ is the update position of an atom i and $\vec{\mathrm{C}_{\mathrm{i}}}$ is its position before the update.

2.3 The Greedy Algorithm Implementation

Given that each minimization cycle must be performed within a 33ms timeframe, the initial plan was to find a local minimum and terminate searching in the hope that the minimum would correspond to a stable molecular conformation. For this reason we used a simple steepest descent (SD) step minimization approach. Its characteristic is that it takes only downhill moves (the greedy approach), which results in converging into the first local minimum it finds. The conformation sampling that we obtained from this method was slow and sometimes erroneous; it only helped us understand some additional aspects of the problem we were trying to solve.

An example conformation produced by the algorithm is presented in Figure 1 and it is obvious that the atom geometry is wrong. The atoms in the histidine aromatic ring should be co-planar, as the urea group on the ligand. The test system was relatively small (<2,500) atoms and there were no performance restrictions. The SD algorithm was converging before t expired and the system would quickly optimize its configuration, within a few minimization cycles, to reach often an erroneous geometry as demonstrated in Figure 1. Further improvements on the system's configuration would be either really slow or not evident and the conformation in the pocket area would always be non-ideal. As a first thought one could suspect that this would be an accuracy problem on the force-field calculations. However, probing the ligand into a different area, inducing further conformational updates, would quickly restore the previously distorted area and cause the same problems in the area of latest interaction.

From the above it is obvious that for the type of protein ligand simulations performed by our application, finding the first potential energy minimum on the molecular energy landscape, would not guarantee a reasonable conformation around the area of interaction between the two molecules. That might be because the rest of the protein could be well minimized and no scalar value for Equation 3 would exist, able to move all atoms in vector space such that the system would reach a lower energy value. What is also worth noting is that keeping the ligand stalled in its position in order to allow the system to slowly recover would not always work either. That gave the impression that every new minimization cycle would pick up from almost where it left off and a new minimum could not be found. As a result further conformational updates would either not be performed or they would be too few with a very small lambda value (i.e. $\lambda = 10^{-6}$) which would have a quite insignificant impact on the system's configuration.

In our attempts to design an efficient energy surface algorithm we also considered Monte Carlo and simulated annealing methods. However, given the strict time constraints and the finite amount of steps as a result, perturbations relying on a probability would expose our solution to the risk of wasting moves inside a single valley. This means that during a 33 ms cycle these techniques would under-sample the potential energy surface, compared to faster sampling techniques like iterative local search. Both methods would be good candidates to the problem of finding the best conformation for a protein ligand pose on a non-interactive application without strict time constraints, but probably this is not the case in our scenario.

The first attempt to solve this problem was to perform a “shuffle” technique. This means that at the beginning of each minimization cycle and before searching for a local minimum, all atoms in the system would move along their force vector trajectories using a certain value. Experimenting and fine-tuning with different λ values showed that a good value could be found in the 10^{-3} magnitude. In order to avoid big atom moves, a constraint was placed for the maximum value of displacement. This technique improved the situation, but raised concerns for the overall quality of the system’s conformation as a result of the frequent shuffling.

In a second attempt, we decided to introduce an uphill move approach, by accepting a random small uphill step as a transient state to recover the system from a local minimum. In this way we could explore the potential energy landscapes for other valleys in the hope that a lower energy value can be found. This technique improved the local restoration defect evident in previous experiments. The potential energy surface (PES) exploration would not always result in a lower energy configuration, but the configurations visualized within the interaction area between the two molecules improved significantly. This is due to the fact that exploring the energy landscape in this way triggered a higher number of conformation updates. This then caused the atoms within the interaction area to move along their force vectors a number of times and eventually form a stable conformation, with correct bond lengths and angles on both molecules.

These experiments led us to reformulate our objectives to seek for a method able to search the potential energy landscape for a good minimum and also restore the area of interaction between the two molecules in a single minimization cycle. The new algorithm should be able to evaluate candidate solutions in efficient manner. It should also ensure that within each minimization cycle there would be enough conformational updates to restore the area of interaction.

The resulted implementation uses an evolutionary algorithm to generate a set of candidate solutions to the problem (system conformations). The candidates are evaluated by calculating their potential energy asynchronously using CUDA streams, which allows for a degree of performance gain. The potential energy value of each candidate is used as an indicator of fitness. The fit candidate is the one whose potential energy value is lower to the latest record of potential energy minimum (old minimum). If no child solution has a value lower than the old minimum, then the fittest child is chosen with criteria set by our uphill move strategy. The fittest child becomes the parent of the new solutions generation and the system's configuration is updated adopting its co-ordinates. Finally after the conformational update, the force vectors for the newly updated system are calculated.

2.4 Fast Surface Exploration and Induced Local Restoration

Initially our algorithm would find its first valley and aim downhill until it reaches a minimum. On reaching a minimum, the parent solution would be unable to generate a population with a candidate solution whose energy value is less than the old minimum. At this point a strategy for escaping the local minima is needed, which would be to immediately try a different valley in hope of finding a better minimum and induce stable conformational updates at the same time. The perturbation strategy of ILS (Iterated Local Search) is ideal for quick jumping from one valley to another by accepting an uphill move when the system lies at a local minimum. This is achieved by choosing a scalar λ value that is able to perform a shuffle operation in a similar way as explained earlier.

ILS initially gave promising results, however, there were two main problems with it. The first is that it would occasionally show a visual pulsating effect, where atoms would seem to perform a fast oscillation. That was because the time-lapse of the algorithm would expire immediately after a perturbation and the system configuration on exiting the algorithm would not be optimal. This problem could be easily solved with a ghost particles buffer holding atom co-ordinates after a perturbation move. The post-perturbation generations would inherit and update co-ordinates on the ghost buffer up until a new minimum is reached lower than the one associated with the real co-ordinates array. This solved the pulsating visual effect, but initiated the second problem that we described, where ILS would often find a good minimum within the first few moves, and the background search following the last visited minimum would not be able to find a better solution. This would lead into the aforementioned problems of too few conformational updates and hence inefficient minimization cycles.

Guided local search is ideal for inducing updates on the principle that the old minimum is being raised by a small percentage at every solution evaluation (penalty function). Eventually, the algorithm can escape from the well, but will spend some time in it waiting for its penalty function to reach the appropriate levels. Experiments showed that this method underperformed compared to ILS for our simulations as shown in the results section.

Our final solution came as an amalgamation of the two approaches. We introduced a penalty value p and an energy function E, which is updated at every step i: $E(i + 1) = E(i) \cdot (1 + p)$. The penalty value is usually in the magnitude range of 0.001 – 0.01, which means that on every iteration the old minimum bar is slightly raised so that it can allow for an update of the system's real co-ordinates. This modification aims to keep the fast track of valley exploration that ILS can offer and introduce enhanced conformational update abilities to our approach. We will refer to this algorithm as IGLS (Iterated Guided Local Search).

In essence Iterated Guided Local Search achieves the following: it accepts updates for energy values close to a recently visited lowest energy configuration, for subsequent solution populations. Failing this, when subsequent solution generations do not produce an update, it prevents the algorithm from wasting too many generations without performing an update, which in turn benefits the visual effect of our application as well as the interacting molecules configuration, especially within their interaction zone.

The implementation of the above technique is performed by using two energy variables: current_lowest and the global_lowest. Every time a new minimum is reached, both the current_lowest and the global_lowest take its value. When a local minimum is reached and the new generation cannot provide a candidate with a lower energy value, perturbation takes place for a new well to be explored and the current_lowest takes the corresponding energy value. While in the new valley, the global_lowest value is raised at every energy evaluation by a small factor (such as 0.5%), while the current_lowest keeps reaching lower values, until we reach the new local minimum. Now at this point there are two possible outcomes: the first is that the new local minimum (current_lowest), is lower than the global_lowest and a conformation update is being performed. Otherwise, a new perturbation move is being performed. However, the probability of finding a new minimum this way is directly proportional to the number of energy evaluations performed. This way, a greater ratio of contributing moves compared to the total number of moves is achieved, which in turn assists into solving the local restoration problem.

This algorithm allows for quick exploration of molecular energy surfaces, by encouraging more conformational updates than a purely ILS strategy would perform. The visual result is significantly improved, as there is rapid local restoration on the impact area.

2.5 Asynchronous Execution

The performance of the above algorithm is important for our application as the potential energy landscape exploration is constrained to run for a small timeframe t. The more moves we are able to perform in the landscape, the better the quality of the resulting solution will be. In addition, the more parent solutions the algorithm generates, the higher the probability of discovering lower energy minimums. From the above, we understand that performance is vital for our simulations. Initially, our force and energy kernels as well as a binning method to an irregular grid approach were designed to minimize execution time on GK104 chipsets. After the introduction of GK110, concurrent streaming capabilities were enhanced, giving the option of running up to 32 kernels asynchronously, when the criteria listed in the background section were met.

For this reason, we designed an algorithm to take advantage of asynchronous kernel execution using CUDA-streams (Hyper-Q according to NVIDIA terminology) and improve the performance of the existing CUDA functions. In order to achieve peak performance, all the aforementioned kernels (energy, force, binning) were refactored in order to meet the criteria for concurrent execution and without sacrificing their individual performance. On the current generation of graphics cards (GK110), we can now evaluate fit of a whole generation of solutions asynchronously and almost in parallel for smaller protein-ligand systems. This means that the first energy evaluation kernel hides the latency of the remaining s-1 evaluation kernels scheduled to run asynchronously in s streams within a single move (a population evaluation series, followed by a force calculation for the fittest child), where s<=16. However, if the conditions mentioned in the background section are not met, then the number of asynchronous streams running concurrently at any time drops as demonstrated in Figure 2.

In addition, we can hide the latency of binning the atoms into a 3D irregular grid for the cell-list generation in the force gradients kernel. This means that binning overheads are no longer an issue and binning can be safely performed at every single move, with a small skin valueδ' that caters for the biggest displacement that we anticipate during each move (cutofflist = cut-off + δ')

2.6 Performance of the Algorithm

The performance of our CUDA-streams implemented hybrid approach has been benchmarked against a synchronous execution version in CUDA of the above algorithm. In order to perform a fair comparison, benchmarks were scheduled such that both versions would perform the exact same amount of energy and force gradient kernel calls. The results are listed in Figure 3 for 4—16 concurrent streams. From the figures we can observe that there are performance gains fluctuating from 0.2X - 4X depending solely on the system size and number of streams. Regarding shared memory usage, both our non-bonded energy and force gradient kernels are using 9kB whereas the bonded energy and forces kernels do not use shared memory. That means that our concurrent execution on energy evaluations at any time is restricted to five kernels as a sixth kernel would overtake the 48kB shared memory per SM limit.

Our algorithm achieves five concurrent energy evaluations for system sizes up to 10,000 atoms. This is because the energy kernels divide execution into blocks of 256 threads, where each thread is associated with one atom. This equates into a maximum of 40 blocks per stream, which is good enough to saturate the limit of five concurrent streams given from shared memory restrictions (5 x 40 < 208 concurrent blocks limit). Beyond that limit, the number of blocks needed to process the energy kernels rises above 208 and that means that CUDA cannot handle the total number of blocks scheduled for asynchronous executions, hence concurrency efficiency starts deteriorating.

Finally, for systems above 50,000 atoms, concurrency almost vanishes on the energy kernels as the number of blocks needed to process one system alone is close to the architecture's limit. However, concurrency in the forces-binning stream pair still holds as our binning method has been designed using a series of fast executing kernels able to run using only a few blocks. In addition, bonded forces kernels require fewer thread blocks and hence can contribute towards some asynchronous execution too. This is why even in the case of large systems (50,000 – 65,000 atoms) there is still a small performance gain.

Tables 1a-b demonstrate a comparison of the elapsed time of execution for the calculations demonstrated in Figure 3 between CUDA-synchronous and CUDA-asynchronous executions. In addition, it appends execution times for the serial implementation on a workstation equipped with a Tesla K-20 GPGPU card and Intel Xeon E5-4620 CPUs running at 2.20 GHz. The serial execution uses the OpenBabel library for the MMFF94s calculations.

2.7 Heuristic Ability of the Algorithm

Figures 4a-d provides a schematic overview of the four algorithms evaluated in this report. The graphs depict the steps taken by each algorithm in order to complete a 33ms computational cycle. In order to demonstrate how IGLS solves our problem, the starting configuration was chosen after a run of 1,000 steps of the greedy algorithm. This way, the starting conformation lies at or very near a local minimum and Figures 4 a-c demonstrate how each of the three proposed techniques escape it and explore the landscape. For these calculations, the parameters used are now described.

Eight equally spaced values are chosen in the range 0.1- 0.000001 and allocated to each computational stream; perturbation for both GLS and ILGS is happening choosing the child solution with index four. There has been some experimentation with different parameters for the perturbation step, but results are similar. The essence of perturbation is to escape the current valley and explore the rest of the landscape and we have no knowledge where each child solution will bring us. The penalty value was set to 0.05 for both GLS and IGLS . In IGLS it is applied every time a perturbation takes place and in GLS every time a new global minimum is not reached. A good example of the benefits of our approach is demonstrated in Figures 4 a and b where ILS and IGLS are compared head to head. In steps three and four both approaches are perturbed and in step five the proposed approach was able to accept a very good solution by slightly relaxing the eGlob constraint (global_lowest) with the aid of the penalty function. The reason why GLS was discarded is because it can spend too much time in the same well waiting for the penalty function to raise over the latest discovered energy levels. We believe that the proposed method inherits the merits of both ILS and GLS for the solution of our problem. Finally, the greedy algorithm had converged before step 1 as we already explained, hence we see a straight line depicting the best child solution repeatedly found on each circle (probably for λ=0.000001).

Another set of experiments was designed to show the fluctuation of energy values using the four algorithms in comparison. In order to perform this evaluation, we ran two sets of experiments. The first set listed in Figures 5 a-d demonstrates the heuristic ability of our approach for ligands positioned close enough to a protein in order to trigger interactions. The second set of experiments listed in Figures 6 a-d is similar to the previous set, with the only difference that it performs the same number of minimization steps for proteins in their initial pdb file structure.

In the simulation results presented in Figures 5a and 5c we can see that the hybrid method and the ILS clearly outperform the GLS and the greedy approach. It is also interesting to see that in all four figures the curves of the hybrid and ILS methods having a similar trend, with the hybrid method following a more turbulent trajectory due to its penalty function. Figure 5b shows an interesting case as we can see that ILS is trapped at a minimum, which it cannot escape from. From Table 1 we can see that for the experiment with 3F9E.pdb structure, ILS could only reach four minima and hence perform four updates. This explains why this method cannot perform well on our simulations, as this is a common scenario. Four updates are usually not enough to bring the system back to a stable conformation. Our method on the other hand, using the penalty function can escape such a minimum and carries on exploring different energy landscapes, finding new minimums and updating the system into better conformations. Regarding GLS on its own, as we can see it always underperforms compared to the hybrid method and it only performs better than the ILS at the 3F9E experiment, where ILS gets stuck in a well.

The four graphs in Figure 6 present similar behavior. Our hybrid method, together with ILS are the best performing ones as they always have the ability to discover deeper valleys in comparison to the other two methods.

From these results, it is evident that the hybrid approach is the most suitable for the type of simulations that our software performs for its ability to find a very good potential energy minimum by inducing more conformational updates than it would following a pure ILS strategy. Also, the GLS element in it does not really mean that the resulting global minimum would be lower than the corresponding ILS, nor is the opposite true. The hybrid method seems promising in solving other optimization problems too due to its ability to change over different energy landscapes and evaluate several different minima at a small temporary potential energy cost.

Overall, the improved hybrid algorithm was fast and responsive in our simulations. There were no evident distortions in any of the molecules structure and interactions between ligand and protein were modelled accurately in real time. This algorithm is capable of simulating systems of up to 30,000 atoms, with the interactivity being inversely proportional to the system size. For larger systems of 16, 000 – 30, 000 atoms, the user has to stall the ligand inside a pocket for the minimization algorithm to take effect. The overall conformation quality can improve with the addition of a long range electrostatics routine. Finally, as the algorithm relies almost solely on floating point operations and future GPGPU generations promise even better FLOP capabilities, we could expect our approach to perform even better on forthcoming cards.

2.8 Algorithm Practical Applications

To further validate and assess our system we have also carried out a set of simulations that aimed to mimic the actual use of the software. To evaluate the induced fit effect, we have selected a series of known biological target for which a series of crystal structures are available (Table 3) as *apo* form and co-crystallised with one or more ligands.

Our approach was to extract a ligand from a specific complex, then place it in the *apo* form of the corresponding protein using the haptic-driven simulator. The results obtained from the docking were then compared with the structure of the complex from which the ligand was extracted, measuring both the root mean square deviation (RMSD) of the ligand and the protein residues of the binding pocket (superposition for the different proteins was performed on the whole structure using the backbone CA as reference).[31] Furthermore, the pocket residues were also compared with the corresponding amino acids in the *apo* form and their RMSD calculated. In this manner, we could estimate of how much the protein had reacted to the ligand binding and if the induced-fit effect was comparable to the actual crystallographic structure of the complex.

The protein we have chosen presents three scenarios: CDK2 presents a fairly rigid binding pocket and only a few minor side chain movements are visible between the *apo* form and the ligand/protein complexes. The two chosen allosteric pockets of the HCV NS5b present a more evident induced-fit effect and, in some cases, there is also a clear protein backbone movement;

the HIV reverse transcriptase presents one of the most dramatic effects of induced-fit, as the binding pocket of the non nucleoside reverse transcriptase inhibitors (NNRTIs) is not even visible in the *apo* form. All the simulations were performed on a 2009 quadcore MacPro, running Ubuntu 12.04, equipped with a NVIDIA Geforce 680 and a Phantom Omni haptic device. As we consider the idea of developing a natural and intuitive interface at the core of our software development efforts, we have asked an undergraduate Medicinal Chemistry student to perform the simulations.

2.8.1 CDK2 results.

For CDK2 we have run 8 simulations, one for each ligand reported in Table 3. In terms of ligand placement, the haptic-driven simulation performed relatively well; the ligand RMSD values ranged between 0.89Å and 2.23Å. The binding site residues RMSD also produced some very interesting results: Figure 7 shows two examples of the results obtained and it is possible to see how the protein has moved in the simulation. Interestingly, the biggest movements are seen on the residues that are indeed the most flexible when the *apo* and the complex structures are compared.

Figure 8 shows how the different residues have changed and if their movement is toward the conformation present in the complex structure. In this case, it is possible to see a more complex scenario, where some residues move towards the corresponding position in the complex structure (indicated in green) and others that move away from that ideal position (indicated in red).

However, we should point out that the intervals parameters represented in Figure 8 are not absolute values but they represent how far away the haptic generated structure is to the crystallised structure, relative to the RMSD of the crystallised structure vs the *apo* structure. As most of the residues virtually retain the same conformation in all the different structures, their actual absolute RMSD-*apo/cryst* values are very small. Hence, any minor, favourable or unfavourable, movement generated from the haptic-driven simulation will be flagged as significant. An exception is represented by Lys33, which moves considerably from the *apo* structure to the 1JVP crystal structure (RMSD of 2.45Å). In the haptic driven simulation, the residue move toward the position present in the crystal giving a RMSD-haptic/cryst of 1.75Å (Figure 9). In this case it is clearly evident the induced-fit effect generated by the haptic-driven ligand placement.

2.8.2 HCV NS5b results.

The two pockets selected for this target (Palm and Thumb-2) present a more obvious induced-fit effect, compared to what has been observed with CDK2. Indeed, the results obtained are more informative (Figure 10) as significant conformational changes from the *apo* structure to both the haptic generated structure and the crystal complex are evident for a good number of residues.

The data presented in Figure 10 suggests that there is a general tendency of the binding site residues to move in the direction of their respective complex conformation. In one specific case, even a relevant backbone movement is present (Figure 11). Interestingly, some residues move away from the ideal final position and this seems to be happening for two reasons: the placement of specific ligands is not accurate (the ligands RMSD range between 1.10Å and 3.10Å); some specific residues should undergo a considerable conformational change and the algorithm is not yet able to explore the full rotamer space available to the residue.

2.8.3 HIV RT results.

The HIV reverse transcriptase presents a very different scenario: the NNRTIs pocket is not structured or visible in the *apo* form. The extent of the induced fit effect, in this case, is really remarkable. The haptic-driven placement results are shown in Figure 12 and they clearly demonstrate the complexity of these simulations. The user can push the ligand inside the pocket and all the RMSD values shows a positive move.

To a certain extent, this is not surprising, considering how different the *apo* structure and the crystal complex are. Indeed, we can see that although the haptic-driven simulation is able to induce a substantial rearrangement in the protein to accommodate the ligand in a reasonable manner, the extent of the protein structural change upon ligand binding in reality is so extensive that, once again, the algorithm cannot explore efficiently such conformational space.

3. Conclusions

In this paper we have reported a hybrid evolutionary strategy for exploring molecular energy landscapes in a small elapsed time period. Our algorithm was tuned to take advantage of the extra parallelism that CUDA streams can provide as the forcefield used was also coded in CUDA. The proposed method works well for conformational sampling in situations where the exit criterion for the meta-heuristic is the expiration of a very small timeframe. We have also tested the algorithm in a possible actual usage scenario, by implementing it in our haptic-driven molecular modelling simulator. Using a series of crystal structures, we have examined how the fully flexible haptic-driven simulation was able to mimic an induced-fit effect.

Overall, we believe the results obtained are positive and very encouraging. The haptic-driven simulations occur in a very natural and intuitive fashion. Furthermore, our results shows that we can induce meaningful and appropriate conformational changes into a protein by placing a ligand using an interactive, real time approach. The data obtained also shows some limitations of this method. Although the algorithm performs extremely well, it is possibly too conservative when sampling the conformational space of the protein residues (of course, a more powerful GPGPU card would explore more conformational states). This should not be considered always a negative aspect, as, for example, this more cautious exploration would preserve some important structural information in a binding pocket in those cases when the user applies a strong force to the ligand through the haptic device. However, as in the case of HIV RT, sometime the changes are indeed dramatic and they would require an algorithm able to sample a considerably bigger conformational space to obtain an accurate prediction.

In comparison with a classical manual molecular docking methodology, which often is performed in three separate stages (placement, energy minimization and evaluation), the integrated and interactive nature of our methodology allows the researcher also to explore other aspects of the binding process. For example, the user could investigate and evaluate, to a certain extent, the results generated by approaching the desired active site from different directions, as the actual path taken by the ligand could affect the protein conformational changes. Clearly, an accurate estimate of this process would require a significantly higher computational power than the one available to our system, which currently can only provide a small insight in this part of the binding eventissue, as the results obtained fro the HIV RT suggests.

However, even if we believe our approach represents a step forward in addressing several important aspect of protein/ligand binding, there are still several issues to be addressed. In particular, efficiently exploring the ligand conformational space within a very short timeframe and in an interactive environment remains a challenge. In our system, the results for flexible ligands (>2 rotable bonds) could be very different based on the starting conformation of the ligand. Solvation, as in many other docking algorithms, is also difficult to evaluate. It is true that an implicit model could be used, but often the information provided by explicit water molecules could be essential to understand the accurate binding of a ligand.

In conclusion, we believe the results presented here are very promising and encouraging. We are now taking the next step in the validation our system and the interactive approach, by comparing the haptic-driven simulator to other commercial and non-commercial flexible docking packages.

4. References

1 M. A. Lill, *Biochemistry,* 2011, **50**, 6157.

2 W. Sinko, S. Lindert and J. A. McCammon, *Chem. Biol. Drug Des.,* 2013, **81**, 41.

3 C. Blum and A. Roli, *Acm Comput Surv*, 2003, **3**, 268.

4 D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Reading, Mass: Addison-Wesley Pub. Co, 1989.

5 S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, *Science*, 1983, **220**, 671.

6 N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, *J. Chem. Phys.*, 1953, **21**, 1087.

7 R. Battiti, *Reactive search and intelligent optimization*, 1st ed. New York: Springer, 2008.

8 O. Trott and A. J. Olson, *J. Comput. Chem.,* 2009, **31**, 455.

9 J. Fuhrmann, A. Rurainski, H.-P. Lenhof and D. Neumann, *J. Comput. Chem.*, 2010, **31**, 1911.

10 S. M. Long, T. T. Tran, P. Adams, P. Darwen and M. L. Smythe, *J. Comput. Chem.*, 2011, **32**, 1541.

11 C. R. S. Brasil, A. C. B. Delbem and F. L. B. da Silva, *J. Comput. Chem.*, 2013, **34**, 1719.

12 W. Cai and Xand Shao, *J. Comput. Chem*., 2002, **23**, 427.

13 R. Murty and D. Okunbor, *Parallel Comput.,* 1999, **25**, 217.

14 N. Zonta,I. J. Grimstead, N. J. Avis and A. Brancale, *J. Mol. Model*., 2009, **15**, 193. 2009.

15 A. Ricci, A. Anthopoulos, A. Massarotti, I. Grimstead, and A. Brancale*, Future Med. Chem*., 2012, **4**, 1219.

16 A. Anthopoulos, I. Grimstead, and A. Brancale, *J. Comput. Chem.*, 2013, **34**, 2249.

17 S. Pall and B. Hess, *Comput. Phys. Commun.,* 2013, **184**, 2641.

18 Schulze-Gahmen, U., De Bondt, H.L., Kim, S.H. *J.Med.Chem.***1996** *39*: 4540-4546

19 A. E. Gibson, C. E. Arris, J. , Bentley, F. T. Boyle, N. J. Curtin, T. G. Davies, J. A. Endicott, B. T. Golding, S. Grant, R. J. Griffin, P. Jewsbury, L. N. Johnson, V. Mesguiche, D. R. Newell, M. E. Noble, J. A. Tucker and H. J. Whitfield, *J. Med. Chem.,* 2002, **45**, 3381

20 P. Furet, T. Meyer, A. Strauss, S. Raccuglia and J. M. Rondeau, J.M. *Bioorg. Med. Chem.*

Lett. 2002**, 12**, 221.

21 T. O. Fischmann, A. Hruza, J. S. Duca, L. Ramanathan, T. Mayhood, W. T. Windsor, H. V. Le, T. J. Guzi, M. P. Dwyer, K. Paruch, R. J. Doll, E. Lees, D. Parry, W. Seghezzi and V. Madison, *Biopolymers,* 2008, **89**, 372.

22 Y. N. Kang and J. A. Stuckey, *To be Published*

23 D. O'Farrell, R. Trowbridge, D. Rowlands and J. Jager, *J. Mol. Biol.,* 2003*,* **326**, 1025.

24 F. Velazquez, S. Venkatraman, C. A. Lesburg, J. Duca, S. B. Rosenblum, J. A. Kozlowski and F. G. Njoroge, *Org. Lett*., 2012*,* **14**, 556.

25 G. N. Anilkumar, O. Selyutin, S. B. Rosenblum, Q. Zeng, Y. Jiang, T. Y. Chan, H. Pu, L. Wang, F. Bennett, K. X. Chen, C. A. Lesburg, J. Duca, S. Gavalas, Y. Huang, P. Pinto, M. Sannigrahi, F. Velazquez, S. Venkatraman, B. Vibulbhan, S. Agrawal, E. Ferrari, C. K. Jiang, H. C. Huang, N. Y. Shih, F. G. Njoroge and J. A. Kozlowski, *Bioorg. Med. Chem. Lett.,* 2012*,* **22**, 713.

26 K. Vandyck, M. D. Cummings, O. Nyanguile, C. W. Boutton, S. Vendeville, D. McGowan, B. Devogelaere, K. Amssoms, S. Last, K. Rombauts, A. Tahri, P. Lory, L. Hu, D. A. Beauchamp, K. Simmen and P. Raboisson, *J. Med. Chem.,* 2009*,* **52**, 4099.

27 T. A. Stammers, R. Coulombe, J. Rancourt, B. Thavonekham, G. Fazal, S. Goulet, A. Jakalian, D. Wernic, Y. Tsantrizos, M. A. Poupart, M. Bos, G. McKercher, L. Thauvette, G. Kukolj and P. L. Beaulieu, *Bioorg. Med. Chem. Lett.,* 2013, **23**, 2585.

28 P. L. Beaulieu, R. Coulombe, J. Duan, G. Fazal, C. Godbout, O. Hucke, A. Jakalian, M. A. Joly, O. Lepage, M. Llinas-Brunet, J. Naud, M. Poirier, N. Rioux, B. Thavonekham, G. Kukolj and T. A. Stammers, *Bioorg. Med. Chem. Lett.,* 2013*,* **23**, 4132.

29 R. Esnouf, J. Ren, C. Ross, Y. Jones, D. Stammers and D. Stuart, *Nat. Struct. Biol.,* 1995*,***2**, 303.

30 J. Ren, R. Esnouf, E. Garman, D. Somers, C. Ross, I. Kirby, J. Keeling, G. Darby, Y. Jones and D. Stuart, *Nat. Struct. Biol.,* 1995*,* **2**, 293.

31 Molecular Operating Environment 2012.10. Chemical Computing Group, Montreal, Canada. http://www.chemcomp.com

4-Streams (a)

Proteins	K atoms	synchronous	hyper-Q	Serial
1UGM.pdb	2.103	1.8	0.87	127.5
3RDD.pdb	2.786	1.976	1.115	235
3F9E.pdb	4.74	2.508	1.378	647.5
3FAU.pdb	5.917	3.126	1.677	1017.5
2O0O.pdb	7.165	3.074	1.662	1447.5
3O05.pdb	13.053	4.216	2.368	7430
2EAR.pdb	15.528	4.14	2.826	7450
1TBG.pdb	26.927	5.46	4.74	32277.5

12-Streams (b)

Protein	K atoms	synchronous	hyper-Q	serial
1UGM.pdb	2.103	1.272	0.393	89.16667
3RDD.pdb	2.786	1.502	0.511	165
3F9E.pdb	4.74	1.862	0.74	455.8333
3FAU.pdb	5.917	2.307	0.868	719.1667
2O0O.pdb	7.165	2.248	0.941	1015.833
3O05.pdb	13.053	3.022	1.486	5870
2EAR.pdb	15.528	2.999	1.8	5423.333
1TBG.pdb	26.927	3.559	3.062	25385.83

Table 1 a-b Results for CUDA-synchronous, CUDA-asynchronous and serial executions.

Table 2a. Number of conformational updates for each protein for the four different algorithmic approaches in Figures 5a-d.

	Hybrid	ILS	GLS	Greedy
1UGM.pdb	27	24	20	15
3VB3.pdb	26	20	22	11
3F9E.pdb	18	4	17	5
3OOP.pdb	26	20	18	10

Table 2b*. Number of conformational updates for each protein for the four different algorithmic approaches in Figures 6a-d.*

	Hybrid	ILS	GLS	Greedy
3RDD	23	20	25	13
3FAU	25	20	25	15
2O0O	29	22	19	18
3O0P	27	20	25	12

Table 3. Crystal structures used in the haptic-driven simulations

Protein	PDB	Ligand Name
CDK2	1HCL[18]	-
	1GZ8[19]	MBP
	1JVP[20]	LIG
	2R3F[21]	SC8
	2R3H[21]	SCE
	2R3I[21]	SCF
	2R3R[21]	6SC
	4EK4[22]	1CK
	4FKL[22]	CK2
HCV NS5B	1NB4[23]	-
	3UPH[24]	0C1
	3U4O[25]	08E
	3GNW[26]	XNC
	4J02[27]	1JE
	4J06[27]	1JG
	4JJU[28]	1MB

Protein	PDB	Ligand Name
CDK2	1HCL[18]	-
	1GZ8[19]	MBP
	1JVP[20]	LIG
	2R3F[21]	SC8
HIV RT	1RTJ[29]	-
	1VRT[30]	NVP Fragment

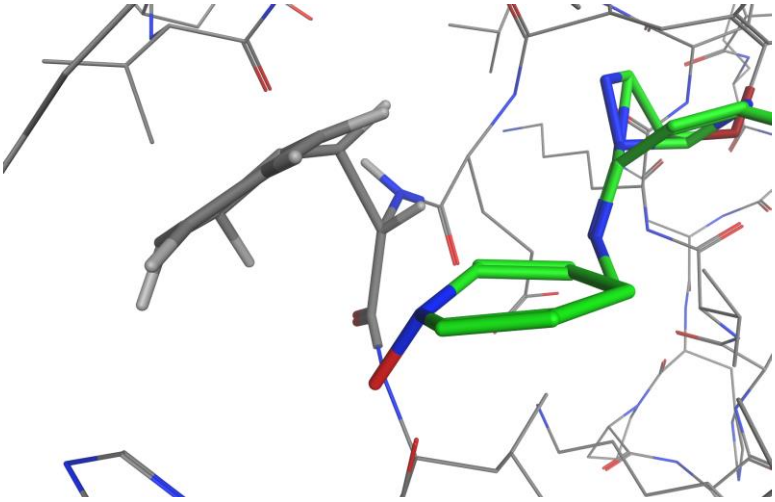


Figure 1. A snapshot of ligand approaching a protein using a greedy local search approach. The algorithm is continuously improving after the initial impact, but settling at erroneous configurations (evident geometric distortion on the protein and the ligand)

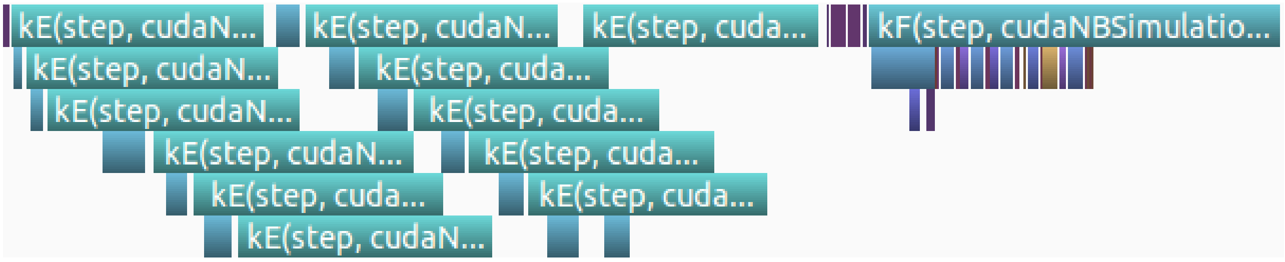


Figure 2. Visual profiling data for a move. Asynchronous execution of s streams where the total block and shared memory conditions are not met. Overlapping is divided into stream groups that meet the hardware's block and shared memory restrictions. The kE blocks represent the non-bonded energy kernels, and the corresponding ones under the kF block are the bonded forces and the 3D grid binning kernels.

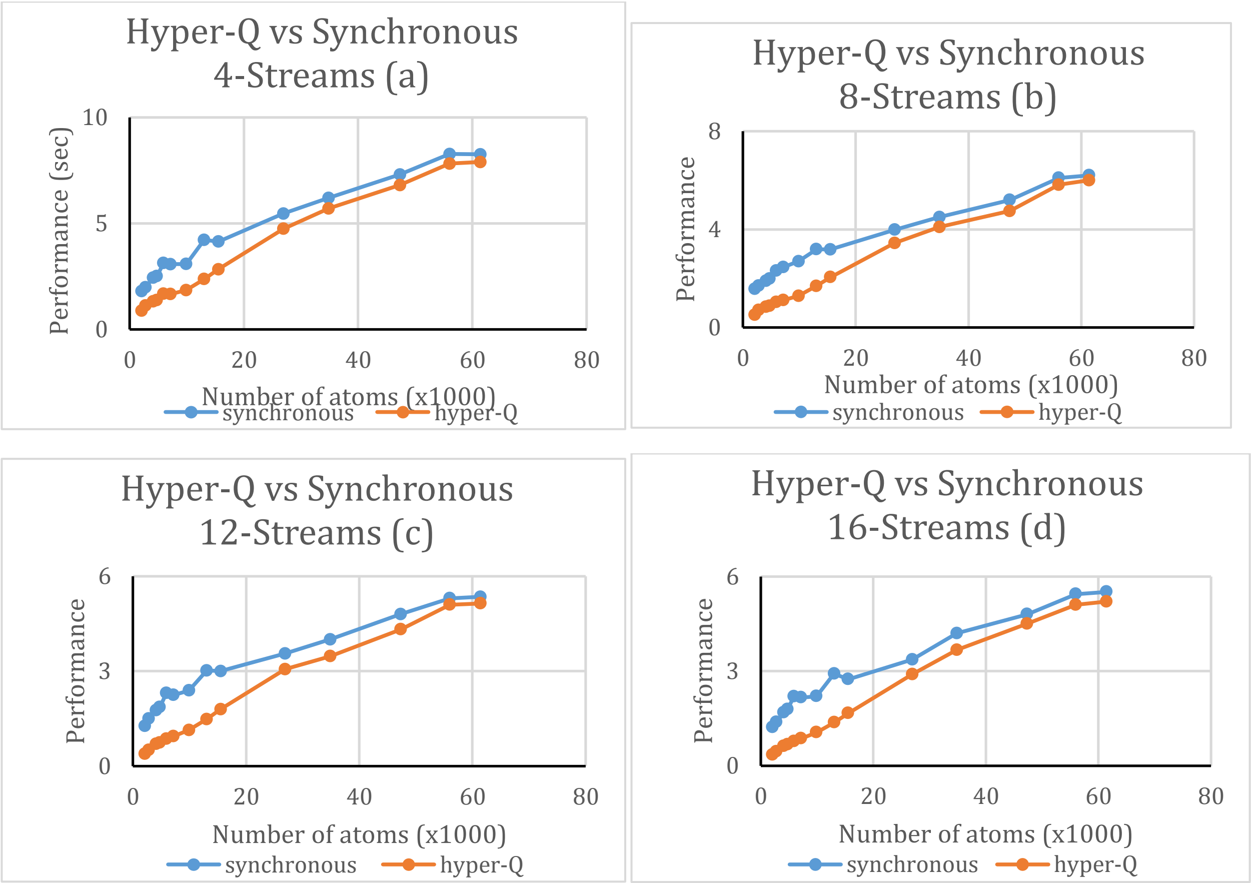
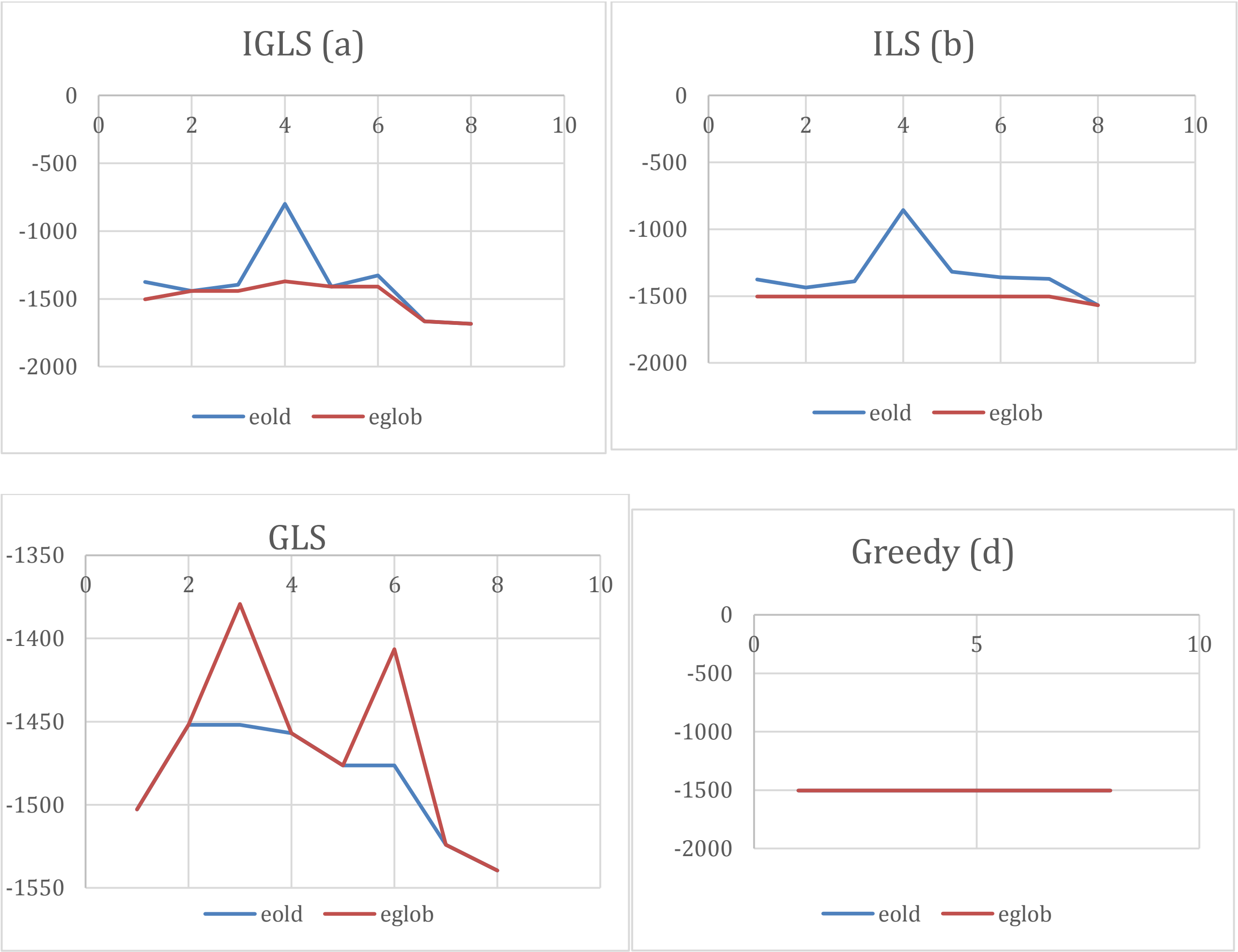
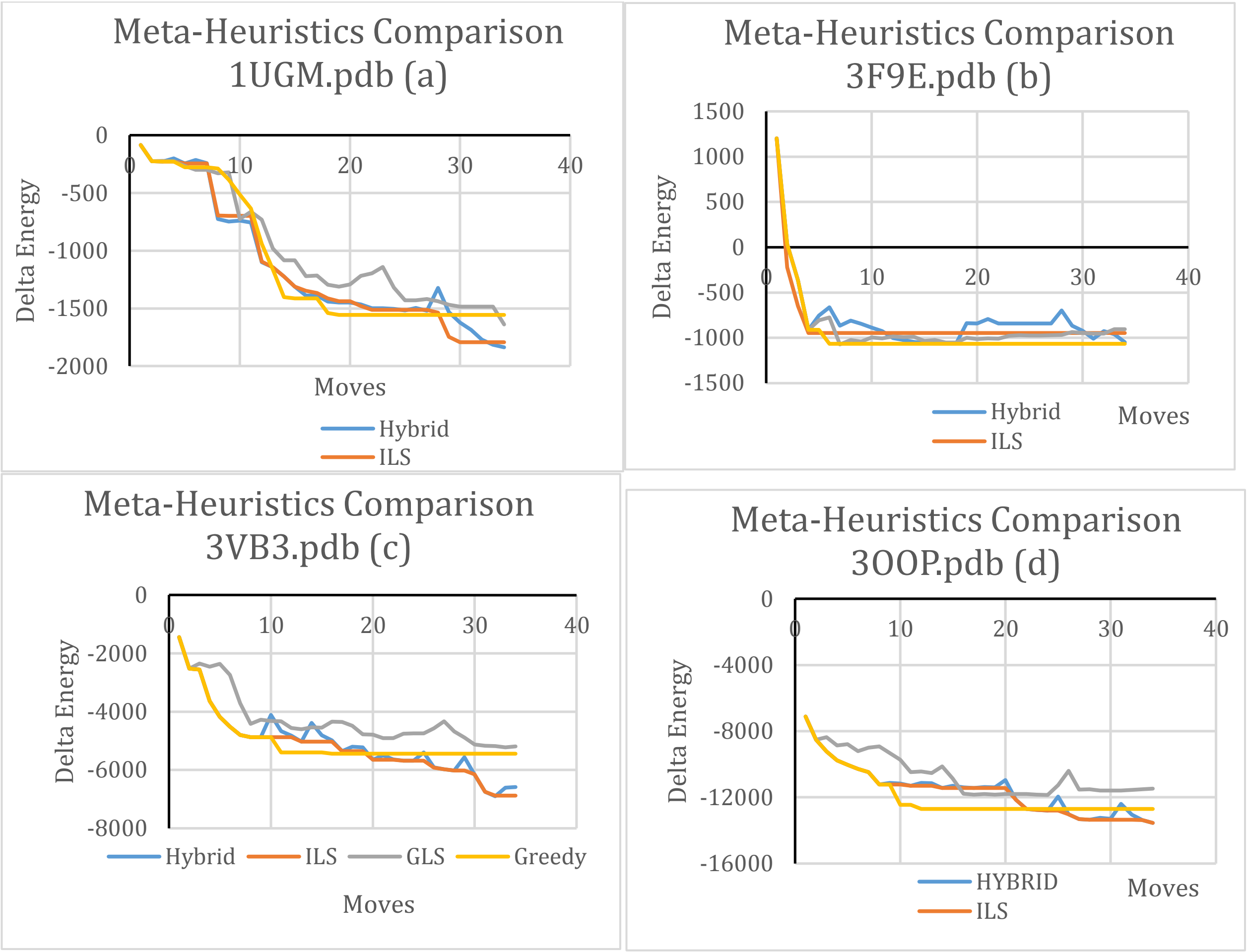


Figure 3a-d. Performance benchmarks between CUDA-asynchronous and CUDA-synchronous stream versions. The same amount of energy kernels (1,000) and Force gradient kernels (1,000/nstreams) are evaluated from both approaches in each graph.



Figures 4 a-d. A schematic overview figure of the four algorithms compared in this report, depicting how a calculation circle of t=33ms is subdivided into 8 steps of a group of energy kernel evaluation streams for the 1UGM.pdb protein file. The system is already minimized with a local search method prior to running the experiment.



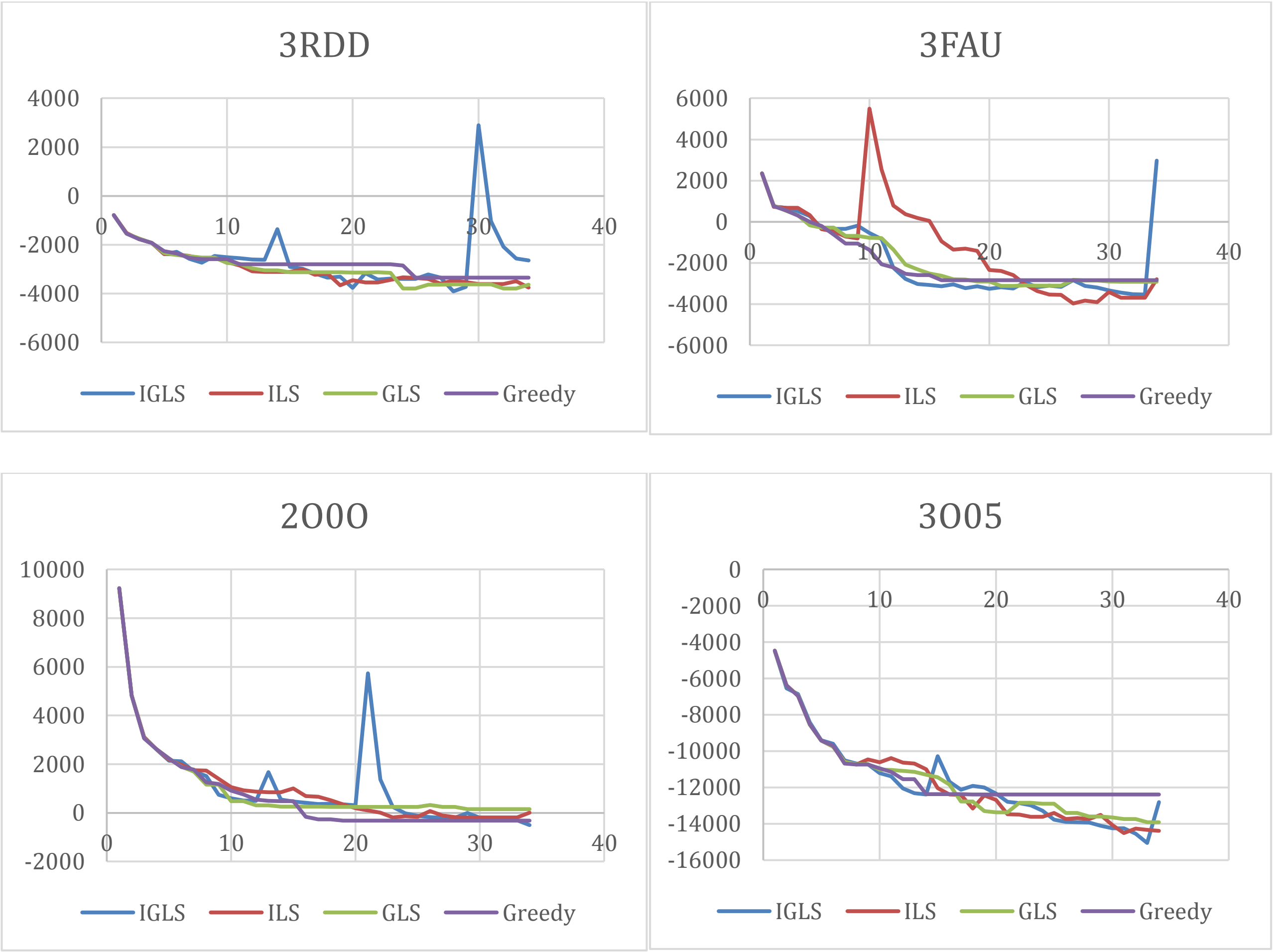
Meta-Heuristics Comparison 1UGM.pdb (a)

Meta-Heuristics Comparison 3F9E.pdb (b)

Meta-Heuristics Comparison 3VB3.pdb (c)

Meta-Heuristics Comparison 3OOP.pdb (d)

Figures 5 a-d. Comparison of the 4 different approaches with the ligand placed into a pocket.



Figures 6 a-d. Comparison of the 4 different approaches with four un-minimized proteins.

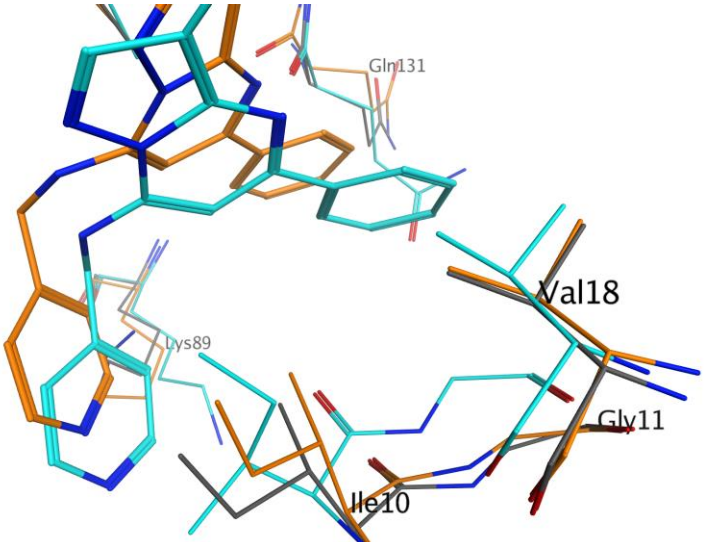


Figure 7. Snapshot the SFC ligand manually docked in the CDK2 as examples of residue movements. The *apo* structure represented in grey; the crystallised complex represented in cyan; the structure obtained from the simulation is indicated in orange.

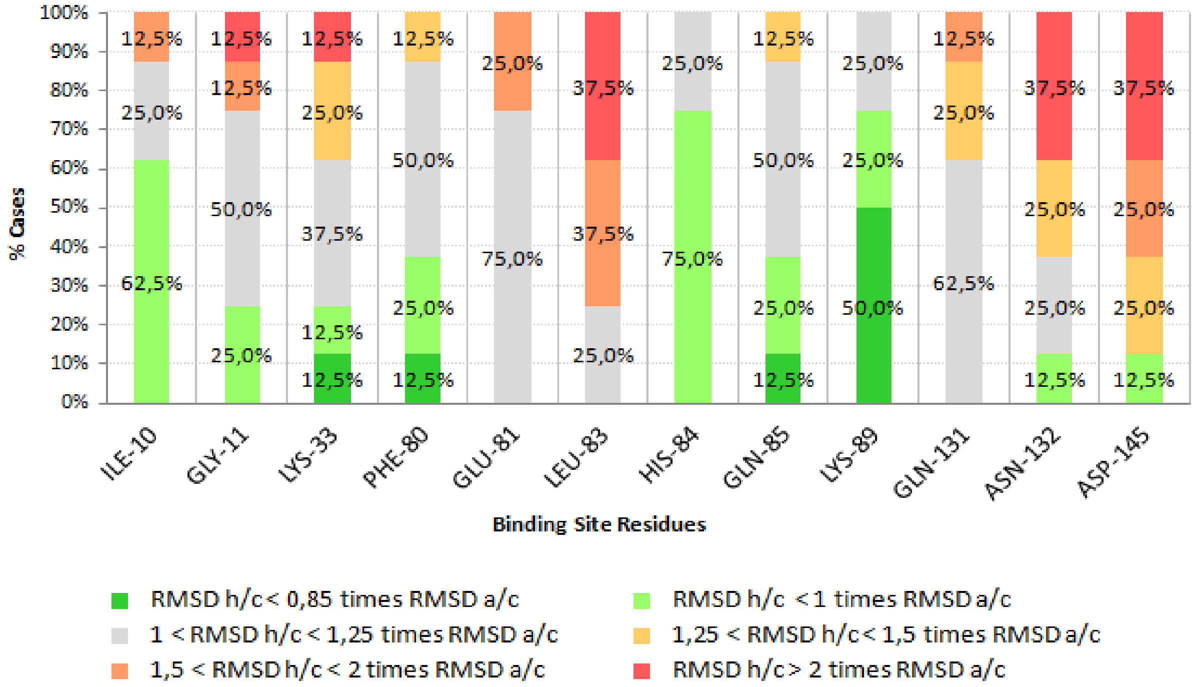


Figure 8. Visual representation of how CDK2 residues had moved during the simulation. The percentage of structures (8 structures – 100%) whose residue RMSD-*cryst/haptic* (referred as RMSD c/h in the legend) is lower than correspondent residue RMSD-*apo/cryst* (referred as RMSD a/c) are shown in green while higher ones are shown in red.

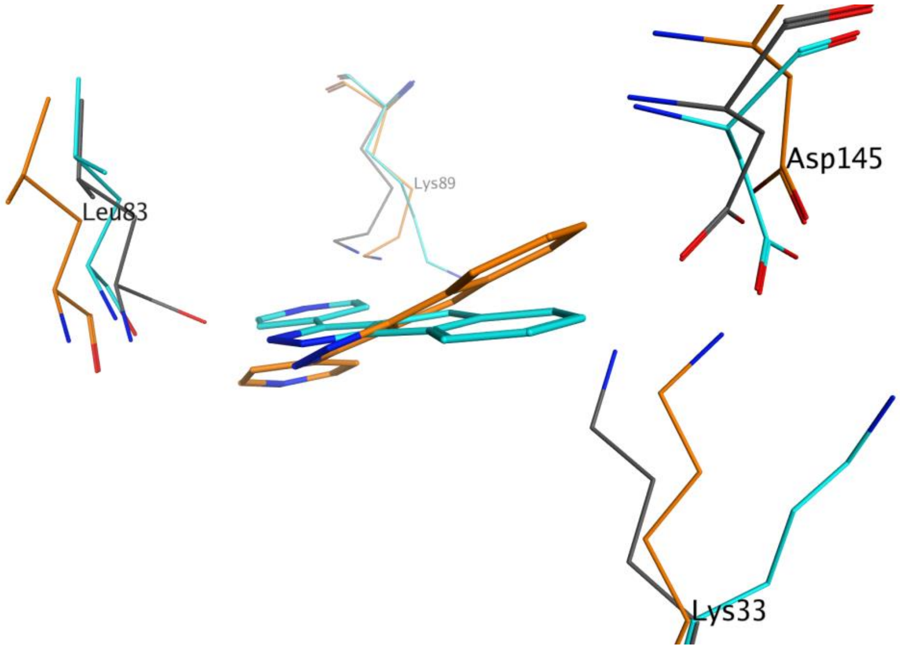


Figure 9: Snapshot of ligand “LIG” in the CDK2 binding site. Lys33 moves toward the position present in the crystallized structure. The *apo* structure represented in grey; the crystallised complex represented in cyan; the structure obtained from the simulation is indicated in orange.

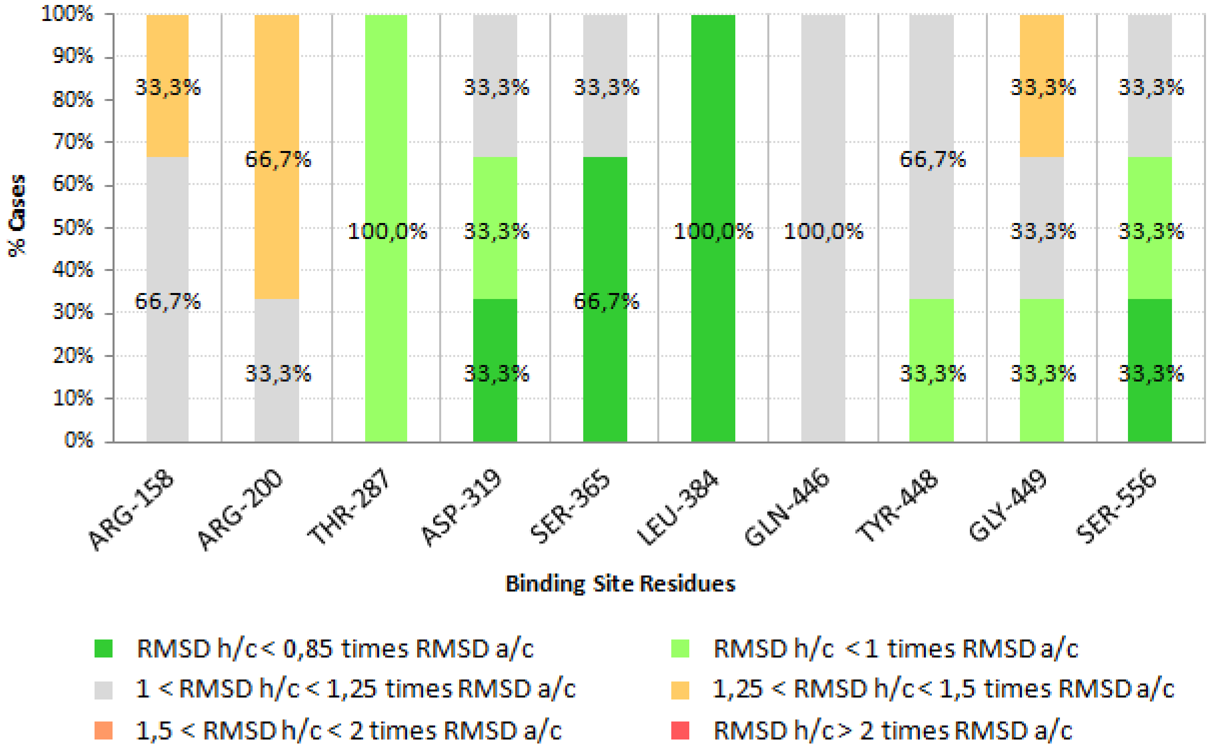


Figure 10. Visual representation of how HCV NS5B residues had moved during the simulation. The percentage of structures whose residue RMSD-*cryst/haptic* (referred as RMSD c/h in the legend) is lower than correspondent residue RMSD-*apo/cryst* (referred as RMSD a/c) are shown in green while higher ones are shown in red.

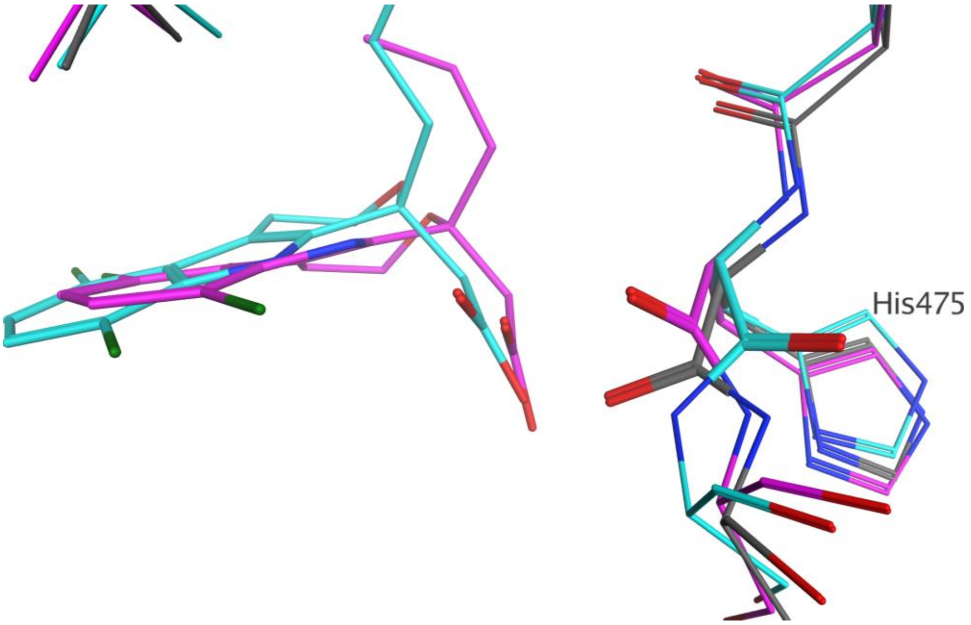


Figure 11. Snapshot of 1JE in the HCV NS5b binding pocket. His475 backbone movement is highlighted. The *apo* structure represented in grey; the crystallised complex represented in cyan; the structure obtained from the simulation is indicated in purple.

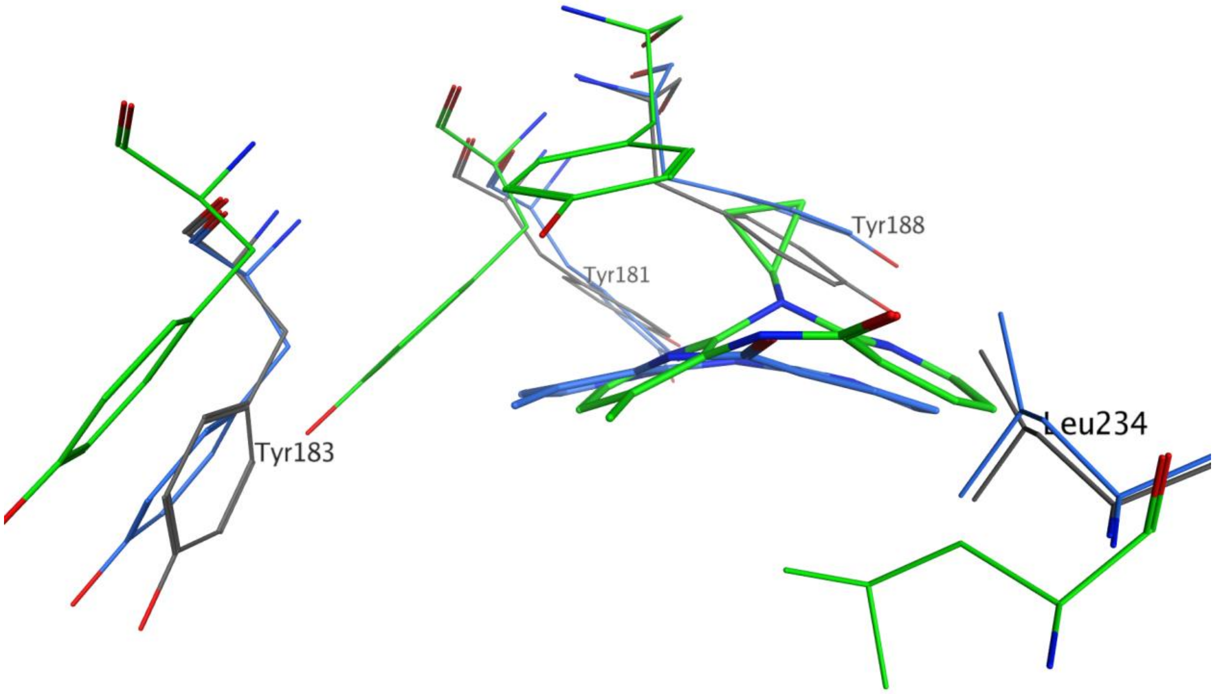


Figure 12. Snapshot of ligand NVP in the HIV RT. The *apo* structure represented in grey; the crystallised complex represented in green; the structure obtained from the simulation is indicated in blue.