# Design of iron(ii) spin-crossover compounds using *de novo* artificial evolution with fitness-function based on DommiMOE's implementation of LFMM

Kjell Nedrelid


Master Thesis in Chemistry

Department of Chemistry

University of Bergen, Norway

June 2015

**Abstract:**

This is a software development adding DommiMOE as a fitness-provider to the *de novo* artificial intelligence internally called Denoptim.

The software supports stochastic conformational search, molecular dynamics, geometry optimization and single point calculations.

Tests revealed a robust software that handled collections of invalid molecules without terminating. The software also worked with Denoptim.

When it comes to conformational searches, needing atleast 1000 steps is adviceable.

While the code for running MD works there are multiple problems with MD making this not an option at this point.

Contents:

# 1. Introduction.

*De novo* design of transition metal complexes is still limited[1, 2] compared to *de novo* drug design[3, 4]. Since *de novo* design is normally a multistep-process[4] one method to make *de novo* design more available for transition metals is by programming an interface between two (or more) softwares there each software can handle one of the steps. Programming the interface between the two softwares mentioned in the next paragraph is a part of this thesis.

For *de novo* design of transition metals an in-house developed software internally known as Denoptim handles the building of molecules[5, 6]. On the built molecules a conformational search and/or molecular dynamics (MD) followed by geometry optimization is often wanted. For this step a fast method that handles transition metals like ligand field molecular mechanics (LFMM)[7] as implemented in DommiMOE[8] is desirable.

Spin crossover (SCO)[9] has many desirable properties and while Fe(II) complexes of N-donor ligands is the most common[10] only a few of these are $Fe(II)N_6$ amines showing SCO. To my knowledge only 3 of these SCO contains only secondary amino nitrogen ligands[11].

The rest of this introduction is split into five subchapters with the first being an overview of *de novo* design and Denoptim. The second subchapter includes computational methods including LFMM. The third subchapter is about DommiMOE while the 4th. is a subchapter with spin crossover. In the last subchapter the problems and aims of this thesis is presented.

## 1.1 Short history of *de novo* design and the main features of Denoptim.

This subchapter is further split into 3 sub-chapters on drug design including common features with transition metals, transition metals including host-design and Denoptim.

### 1.1.1 Drug design, chemical space and scoring.

In drug design high-throughput screening and synthesis was used but due to high costs and low success rates[12] a more efficient technology was desirable and *de novo* drug design was introduced. In the first-generation *de novo* design each molecule was built from atoms. A disadvantage with this approach was the generation of overly complex molecules that lacked synthetic accessibility[4].

As an improvement to this fragment-based methods was introduced. This increases the synthetic accessibility and at the same time decreases the sampling-space. The size of the chemical space is much too large to be sampled fully since even limited to the "drug-like" space an estimate is of the order of $10^{60}$ molecules[13].

After building a molecule and doing geometry optimizations that will be more fully mentioned in chapter 1.2, getting a measure of how good or bad a molecule is compared to other molecules is necessary. The methods depends on the system under investigation but one or more descriptors is used to score the molecules in the

fitness function[4]. One of the problems with *de novo* design is to find a good fitness function.

### 1.1.2 Transition metals and HostDesigner

For transition metals modelling has lagged behind[14] the organic modelling. Some of the reasons for this is the larger numbers of electrons, orbitals and isomers [14]. The features of *de novo* design of transition metals is largely the same as *de novo* drug design since both are now using fragment-based approaches and relying on fitness functions. The availability of software for *de novo* transition metal design is on the other hand fairly limited since most software can't handle the extra complexity of d-orbitals and more electrons.

There does exist one software-package appart for Denoptim specifically designed to handle transition metals and this is the HostDesigner[15]. Here a "host" is compared to a "guest" and the geometrical similarities is used as the scoring [16, 17].

A few other software-packages also includes support for transition metals [2].

### 1.1.3 Denoptims main features.

"An evolutionary algorithm for *de novo* optimization of functional transition metal compunds"[5] shortened Denoptim by combining the underscored letters, is the in-house developed *de novo* software capable of building molecules. Denoptim is written in Java and relies on CDK toolkit[18], TINKER[19] and OpenBabel[20]. Denoptim is a fragment-based builder and additionally it includes evolutionary algorithm (EA). The main loop of Denoptim is shown in figure 1.1.

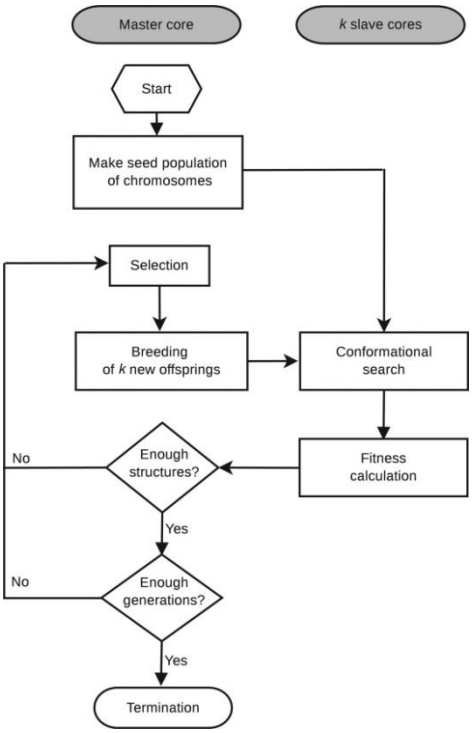


**Figure 1.1:** The overall workflow of the *de novo* evolutionary algorithm. Taken from reference [5].

Denoptim is supporting fragment-based building by internally relying on graphs and edges. A graph can contain a single atom or a complex molecule, there each graph is called a fragment and each edge is called an attachment-point (AP). Each fragment must have at least one AP but can have multiple of same type or of different types with an example shown in figure 1.2.

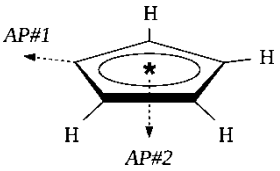


**Figure 1.2:** A possible fragment with two different AP. Taken from reference [1].

While one fragment with a set of AP is really no different from another fragment having the same AP then it comes to graphs and edges, it is still beneficial to split fragments into 3 separate groups. The first group is called scaffolds and all molecules built by Denoptim must include one scaffold. A second group is called capping and is added at the end of building to any AP still empty. The 3rd. group does not have a special name and is only called fragments.

Appart for the different groups of fragments and AP there must also be some rules to how AP on one fragment can connect to another AP. These rules is collected into a compability matrix[1] and will include rules like example AP1 can connect to AP2, AP3 and AP4 while AP2 can connect to AP1 and AP5. This makes it possible to fine-tune the type of fragments can make connections to eachother and should be consistent with the wanted chemistry.

The fragments used for building in Denoptim can either be created by utilizing SMARTS[21] cutting-rules on pre-exisiting molecules or by manually creating the fragments.

To build a molecule in Denoptim there are 4 possiblities with the first being a new molecule only built from fragments. In this instance a scaffold is first picked, whereafter there is for each AP on scaffold a random chance if another fragment is tried built on this AP and if picked for building one of the fragments with compatible AP-classes is used. A fragment can additionally include one or more AP not connected to scaffold and a new random chance is used to decide if fragments is connected to these AP. This process will continue until either all AP is used, a maximum limit for how many fragments can be built on fragments is reached, none of the fragments can be built on a given AP or the random chance for building was not fulfilled. At this point any free AP will be tried filled with fragments from the capping group.

The 3 other possibilities for building new molecules all relies on molecules already present in the population and is part of the EA. The possibilities is random growth, mutation and crossover. In crossover two molecules swaps one or more fragments and two new molecules is made while in random growth and mutation one old

molecule gives one new molecule. These processes can additionally be symmetric and examples of both symmectric and non-symmetric is shown in figure 1.3.

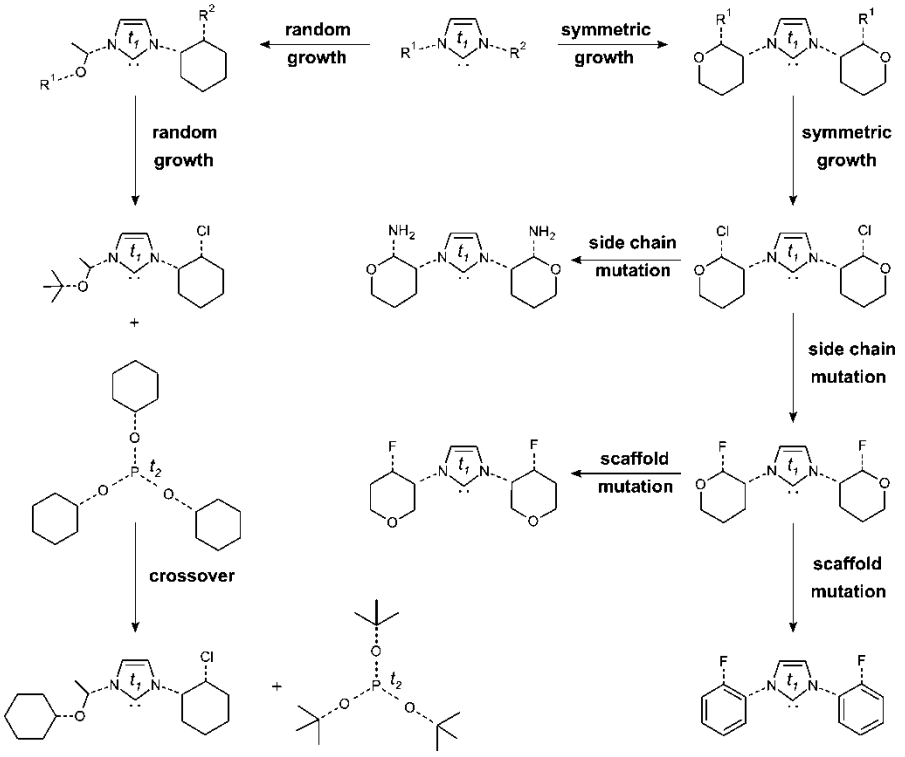


**Figure 1.3:** Examples showing crossover, grow and mutation. Taken from reference [5].

Before EA can work there must be some molecules to do random or symmetric growth, mutations and crossover on. For this reason the start of Denoptim includes a generation-zero containing a fixed number of molecules N built fully from original fragments and this can be called the original parents. In later generations either one or two parents is used to generate new molecules called children based on growth, crossover and mutations or it is possible a new molecule only based on the original fragments is built. For any molecule built from original fragments there are no limitations appart for being valid. For the 3 methods part of EA on the other hand the fitter a molecule is the larger the chance for taking part in making children. There is also two different methods here, there one being the most restrictive called "survivial of the fittest". In this scenario only the N fittest molecules is eligible for taking part in making children, with N being same as in generation-zero. This means for each generation there are always N possible parents. The other method is called "keep growing" and while the fittest molecules has the highest probability of being picked there is still a chance any molecule with a low fitness will take part in making children. This means with N in generation-zero and C children in each generation after example 10 generations the "survival of the fittest" will have N possible parents while "keep growing" will have N + 10 * C possible parents.

While this includes the main parts of Denoptim, two things are not commented on yet and this is the two boxes "Conformational search" and "Fitness calculation" as shown in figure 1.1. This is done in the next sub-chapter.

### 1.1.4 Denoptims handling of fitness script.

Conformational search and fitness calculation as shown in figure 1.1 is not directly handled by Denoptim but instead relies on external software to do these parts. Both steps is from Denoptim's point-of-view combined in a script called the fitness-script. The types of software used is up to the user appart for 3 main restrictions Denoptim has. The first restriction is the script must return any result to the file specified by Denoptim and the file must be of one of the CTfile formats with extension sdf and the older V2000-format[22]. The second is the sdf-file must either contain a number tagged as <FITNESS> in case of success or <MOL_ERROR> in case of any errors. The third is the script can not finish with any non-zero error-code unless user wants Denoptim to crash. One extra small limitation is the single number reported as <FITNESS> must be maximized meaning the higher the number the better the fitness.

For most purposes none of these limitations is a problem since the user-supplied fitness-script can normally easily start multiple other software and/or scripts if desired by the user.

### 1.1.5 Known weaknesses in Denoptim.

The current Denoptim implementation of graphs does not support ring closure[1].

Since Denoptim only looks on fragments and AP there are no limitation of putting two incompatible AP on the same molecule. While filtering out one of the chemical types of fragments and not include these as possible fragments is a possibility this is not always desirable. In some instances the fragments are only incompatible if bound to a common atom.

## 1.2 Computational methods and LFMM.

For doing geometry optimization on a structure the main possibilities are wave-based quantum mechanics (QM), density functional theory (DFT) and molecular mechanics (MM). It is also possible to use hybrid-methods like QM/MM. While all is successfully used on transition metal complexes using QM on SCO-molecules is not a good option since SCO-molecules is too large for using the most accurate QM and fast methods like Hartree-Fock is too inaccurate for SCO[23].

While QM fails DFT on the other hand is successfully used on SCO[23-25]. The speed of DFT compared to MM is on the other hand greatly in DFT's disadvantage[26]. In *de novo* design it is normally necessary to include a conformational search and with the thousands of structures created in this search DFT becomes too slow.

Using MM is therefore the fastest but the accuracy is not always the best. MM is using classical mechanics for determinations of molecular equilibration structures[27]. There are multiple possibilities for the type of functions used in MM and the parameters necessary to calculate energies in MM. The collection of functions and parameters is collected in force fields[28] where some of them are AMBER[29], CHARMM[28] and Merck[30]. The parameterizing in forcefield depends on the intended usage with the choosen parameters a compromise intended to handle same element bounded to many different elemenets. To increase the accuracy forcefields creates extra atom-types where example carbon-sp3-hybridized is one atom-type and carbon-sp2-hybridized is another.

The handling of transition-metals is in some forcefields totally lacking while in others the fit is bad. To improve this the LFMM was created to describe the ligand field stabilization energy (LFSE)[8].

More details about the functions used in MM, LFMM, LFSE and additionally the Morse function[27] is included in chapter 2.

**1.3 The molecular operating environment (MOE)[31] and DommiMOE[32].**
Some of the features of MOE is a windows-based computing, modelling and drug design environment with the emphasis on non-transition-metals. For transition-metals MOE does not look like an obvious choise but by adding LFMM to the mix with the integration of DommiMOE an environment capable of handling transition-metals is available[8].

With over 2000 functions going into all the functionality of MOE is outside the scope of this thesis but one necessary feature is the inclusion of MOE's own programming language called scientific vector language (SVL). Included in SVL are standard features like if…then, for-loops and some standard mathematical functions. In addition to this functions to load and save molecules and manipulate them in many ways are present. Also included is the possibility for a basic try…catch to handle single function-evaluations going wrong without the program crashing.

DommiMOE adds an extra menu to MOE and here the necessary LFMM-versions of conformational search, MD, geometry optimization and single point calculation is included. By loading a molecule into DommiMOE makes it easy to select any of these calculations the inability to load a database of molecules and run a geometry optimization on all molecules in database either serially or parallel is a disadvantage. Another shortcoming is DommiMOE only allows LFMM-calculations if the atom-type of the metal is correctly detected and with DommiMOE relying on MOE for this part having to manually override MOE's choise does happen.

In addition to the graphical interface MOE does include a command-line-version (CLI). By supplying a script to MOE CLI most things appart for visually inspecting molecules and/or drawing them can in my experience be done by script. A bonus here is the CLI uses one license or token while the graphical interface MOE uses 3.

### 1.4 The main features of spin crossover compounds.

When a transition metal in the d-block is approached by a ligand the outer d-orbital energy levels can be split into multiple energy levels. In an octahedral the splitting is into 3 energy-levels being 2/5 $\Delta_o$ below the original while 2 energy-levels is 3/5 $\Delta_o$ above the original level. The splitting is shown into different energy-levels is shown for both highspin and lowspin $Fe^{2+}$ in figure 1.4 and the $\Delta_o$ points to the original state that is the lowspin and is therefore the energy difference between the 6 electrons shown and the 4 emtpy d-orbitals shown for lowspin.

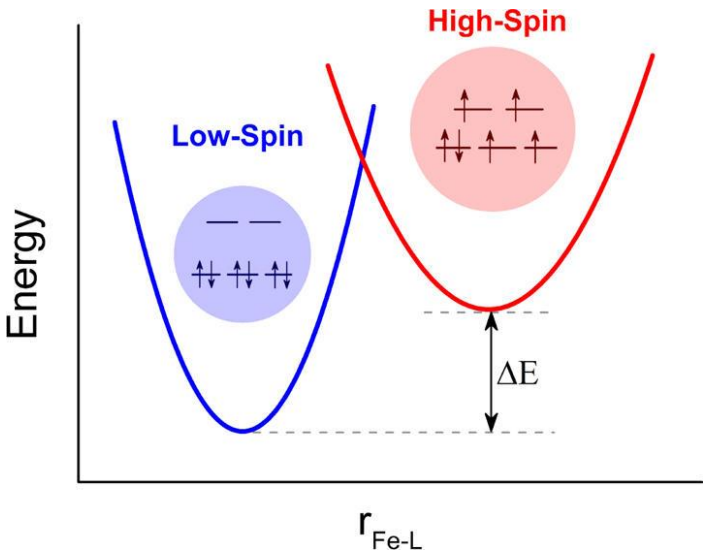


**Figure 1.4:** The figure shows the lowspin and the highspin 3d distribution of electrons in $Fe^{2+}$ octahedral ligand-field with $r_{Fe-L}$ being the distance between iron and ligand. The figure was taken from reference[33].

The splitting into energy-levels isn't limited to $Fe^{2+}$ but will happen for any transition metals with not fully empty and completely full d-orbitals. The splitting will also happen for other geometries than octahedral but for the remainder of SCO only octahedral is used.

A d-electron will want to jump from lowspin into highspin if $\Delta_o$ is not too large, it is currently paired in an orbital with another electron and by jumping to highspin it will not be paired with another electron. This limits the possibilities to $d^4$ – $d^7$ to even be considered. Additionally if $\Delta_o$ is very large the electron will want to jump back down if it gets enough energy and is therefore not counted as SCO. One last requirement is ΔE is not too large.

For any metal complex fulfilling all these requirements you have a metal with SCO. Among the currently known SCO includes $Mn^{3+}$, $Fe^{3+}$ and $Co^{2+}$ but $Fe^{2+}$ is the most common [34].

Some of the effects of SCO is immediately apparent from figure 1.4, this is highspin have longer distance between metal and ligand and therefore normally a larger volume. With atleast one electronspin changing the highspin state is paramagnetic

and is affected by magnetic fields. It is often possible to force SCO to switch between lowspin and highspin either by increasing the temperature somethat that gives more highspin or by supplying pressure something that gives more lowspin. The fraction of highspin and lowspin can follow different types of curves and can additionally be following a completely different path during heating and cooling. This last thenomen is called hysteresis and is shown in figure 1.6.

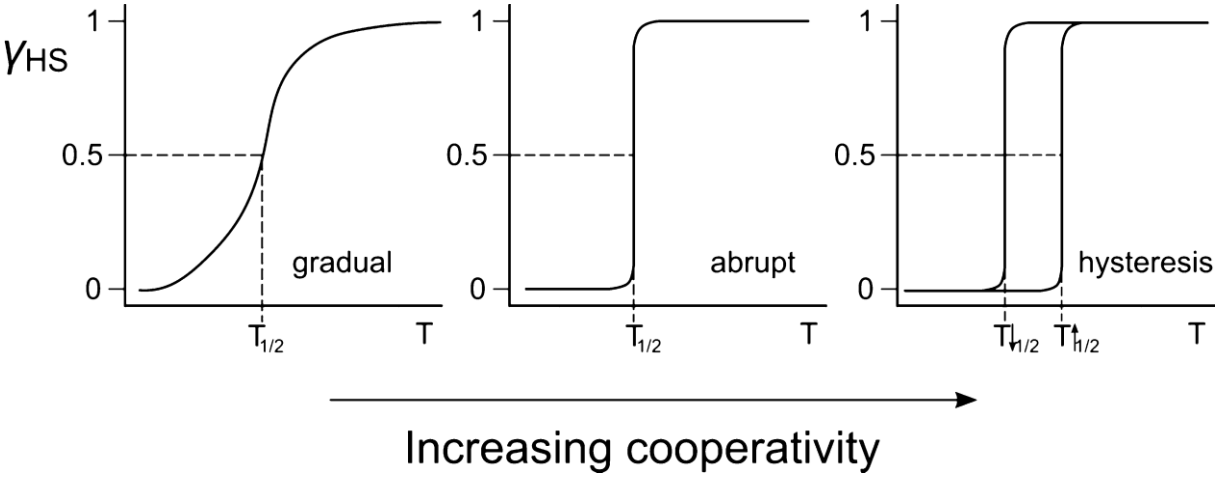


**Figure 1.5:** The figure shows 3 different types of spin crossover, where $\gamma_{HS}$ is the fraction of the molecules in a highspin state and $T_{1/2}$ is the temperature where 50% of the molecules is highspin and the other 50% is lowspin. The figure to the right shows hysteresis. This figure was taken from reference[34].

A SCO having hysteresis is a requirement for the possible usage in switches or data storage. Another feature is the light induced excited spin state trapping (LIESST)[35] as shown in figure 1.7.

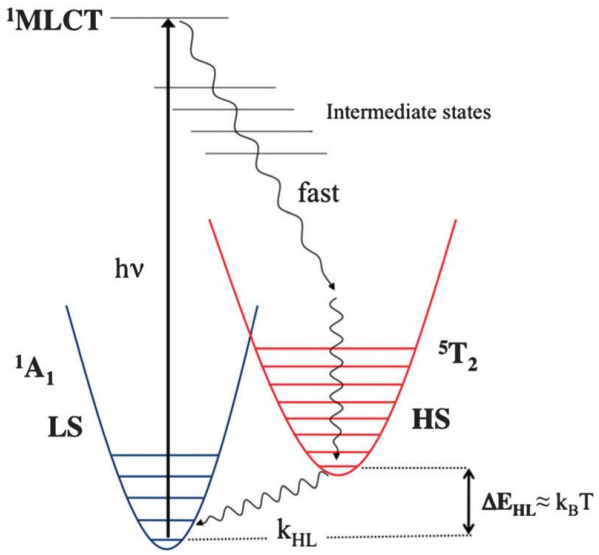


**Figure 1.6:** The figure shows the electronic structure of $Fe^{2+}$ SCO. Also included is the effects of LIESST. The figure was taken from reference[35].

Despite all the desirable features and the continuous search for SCO as of 2014 the only industrial usage of SCO is as paint[36].

### 1.5 The aims and problems.

The first of the two main aims of this thesis is to write the software to enable running 4 DommiMOE-LFMM-functions from the command line on a MOE-database of molecules. The second main aim is to write a fitness-script for Denoptim enabling using DommiMOE as the fitness-provider.

The 4 LFMM-functions has already been mentioned earlier in the introduction and they are stochastic conformational search, MD, geometry optimization and single point calculation. The fitness-script for Denoptim needs to handle the requirements mentioned in chapter 1.1.4. For the problem with atom-types mentioned in chapter 1.3 and incompatible fragments bonded to a single atom in chapter 1.1.5 a method to filter molecules is a minor aim.

While MD was originally planned to be included in Denoptim-runs, due to a problem becoming apparent inside the group only resently[37] this was not possible for this thesis.

For testing-purposes and as input for fragmenting that enable running Denoptim a set of molecules is needed. These are gathered from the Cambridge crystal structure database (CCSD)[38] last updated February 2015.

## 2. Theory:

$$E_{tot} = \sum E_{str} + \sum E_{bend} + \sum E_{tor} + \sum E_{nb} + LFSE$$

LFSE adds a new term to the total as shown in equation above.

## 3. Computational details:

For all DommiMOE-calculations the 2011-version was used. For Denoptim version 0.4.5 and specifically SVN-checkout 8500. Java 1.8.40, OpenBabel version 2.3.2. For some of the Denoptim-runs TINKER version 7.1.2 was used while for others 6.2.

For handling of large molecular databases and the non-LFMM-filtering-purposes in chapter 4.2 MOE 2014 was used.

In the Denoptim-runs the parameters being different between runs was as shown in table 3.1 while the common parameters was as shown in table 3.2

**Table 3.1:** Table of parameters for Denoptim-runs. This includes the Denoptim-parameters and the conformational search parameters for DommiMOE and the parameters used for fitness function. For the fitness function see chapter 4.1.1.

| run id | run_160 | run_201 | run_310 | run_323 | run_325 |
|---|---|---|---|---|---|
| Initial population | 100 | 100 | 100 | 100 | 100 |
| Children per gen. | 50 | 25 | 25 | 25 | 25 |
| Generations | 100 | 70 | 130 | 130 | 130 |
| Build-method | Keep growing | Sufvival fittest | Survival fittest | Keep growing | Survival fittest |
| Crossover probability | 0.5 | 0.3 | 0.3 | 0.3 | 0.3 |
| Mutation probability | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| Symmetry probability | 0 | 0.1 | 0.1 | 0.1 | 0.1 |
| Shortest conf. search | 100+200 | 100+200 | 100+200 | 100 | 100 |
| Conf. exponent | 2 | 2 | 2 | N/A | N/A |
| Longest conf. search | 1600 | 800 | 800 | 100 | 100 |
| Early term. conf. on energy difference [kJ/mol] | > 4.184 | < 4.184 | < 4.184 | < 4.184 | < 4.184 |
| Fitness function constant a | 0.9 | 1 | 1 | 1 | 1 |
| Fitness function constant b | - 3.133 | - 2 | - 1.472899 | - 2 | - 1.472899 |
| Fitness function constant c | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |

**Table 3.2:** The common Denoptim parameters used in all runs.

| Parameter | Selection |
|---|---|
| seed | 2429837395615 |
| numOfProcessors | 10 |
| selectionStrategy | TS |
| growthProbabilityScheme | 2 |
| steepSigma | 1.0 |
| middleSigma | - 2.5 |

For the scaffolds used in the different runs see figure 3.1 and 3.2. The scaffolds used in run_160 had no concept of symmetry. The only main difference is nitrogens gave one AP-class while carbon gave another AP-class. For the scaffolds in the 4 other runs on the other hand some limited symmetry is tried incorporated.

For possible fragments appart for scaffolds see figure 4.19 before cutting. For all runs all ring-hydrogens was connection points. For the runs appart for run_160 all hydrogens on hydrocarbons was also connection points.

For the compability matrices, the one for run_160 is the shortest but is 5 pages long and is for this reason not included. The compability matrix for run_201 is 15 pages and for the common one for run_310, run_323 and run_325 is 16 pages long.

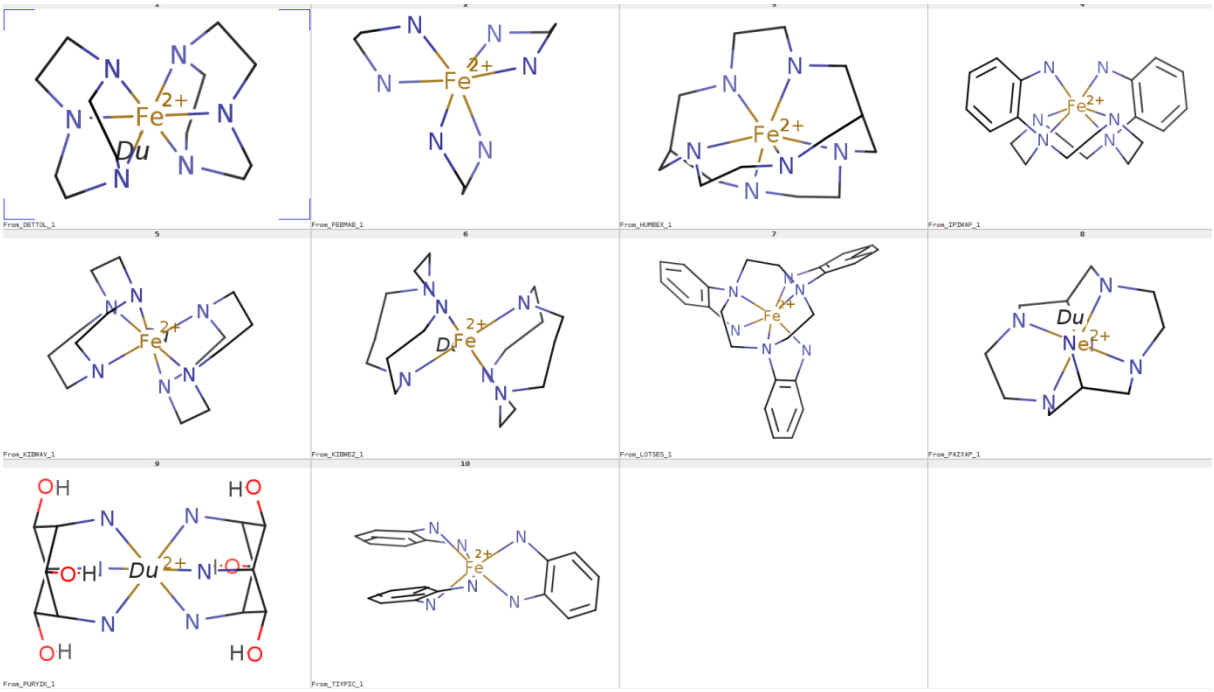

**Figure 3.1:** The scaffolds used in Denoptim run internally known as run_160. While not showing the four scaffolds in the upper line is numbered 1, 2, 3 and 4 from left to right.

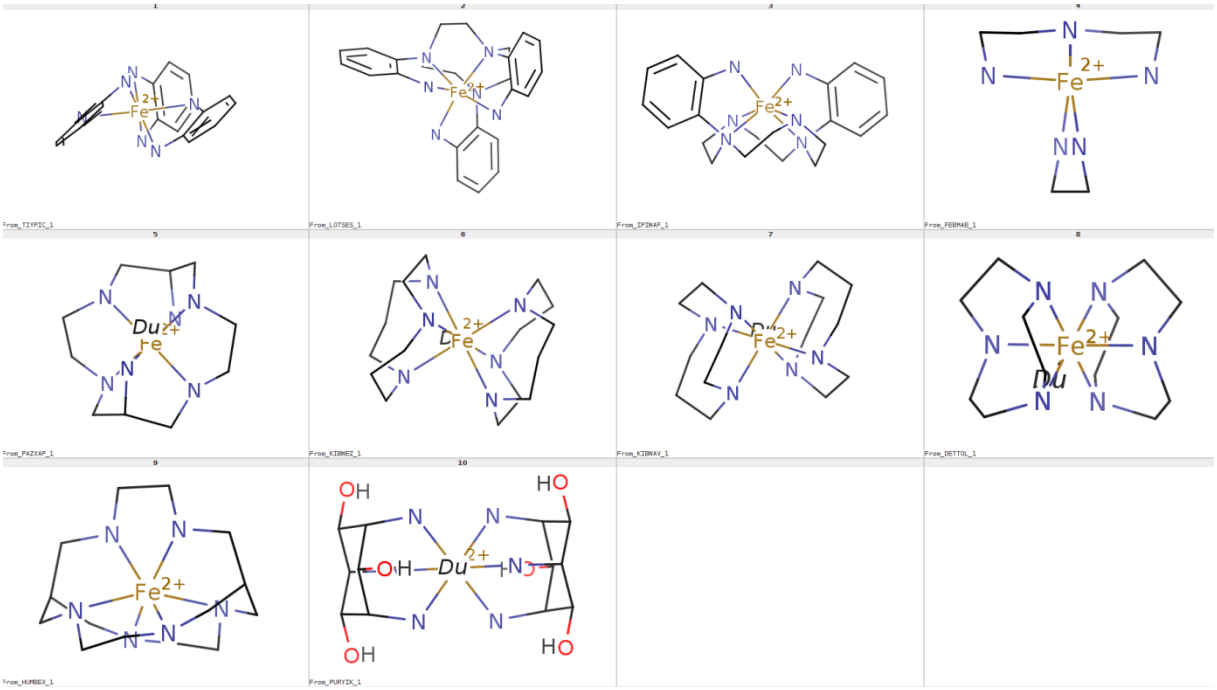

**Figure 3.2:** The scaffolds used for the Denoptim runs internally known as run_201, run_310, run_323 and run_325. The four scaffolds in the upper line is numberd 1, 2, 3 and 4 from left to right.

## 4. Results:

This chapter is split into 4 subchapters, with the first subchapter being the programming part containing mostly flowcharts. The second subchapter includes filtering of molecules, selection of the scaffolds, building of the other fragments and the compability matrices. Included in the third subchapter are the standalone stochastic conformational searches on the molecules choosen as basis for scaffolds. Also included here is a short test on molecular dynamics. The last subchapter includes results based on Denoptim runs.

### 4.1 Flowcharts, program code and the fitness function.

The code written in SVL includes over 100 functions and over 10000 lines of code and comments. Additionally the bash-script used as fitness-script for Denoptim is over 1000 lines. Due to the amount of functions and their volume only a few of the most important ones is included in the form of flowcharts or code-snippets.

Before starting on the flowcharts and code, one important feature necessary for *de novo* design is the fitness function. The selected fitness function is shown in chapter 4.1.1.

#### 4.1.1 The fitness function.

$$fitness = \sqrt{a(U_{HS} - U_{LS} + b)^2 + c\left\{\left(U_{LS\_in\_HS\_geometry} - U_{HS}\right)^2 + \left(U_{HS\_in\_LS\_geometry} - U_{LS}\right)^2\right\}} \quad \textbf{(1)}$$

The choosen fitness function is shown in equation 1. In the equation $U_{HS}$ stands for potential energy calculated in the highspin configuration while $U_{LS}$ stands for potential energy calculated in the lowspin configuration. $U_{LS\_in\_HS\_geometry}$ stands for the lowspin potential energy calculated in the higspin configuration while $U_{HS\_in\_LS\_geometry}$ stands for lowspin The different U is the potential energies in HS for highspin, LS for lowspin and the second to last one is lowspin potential energy calculated in the highspin geometry and the last one is highspin potential energy calculated in the lowspin geometry. Additionally a and c is two constants giving the weight on the two terms. Lastly b is a constant that adjust the zero-point. If larger weight to the lowspin-energy is desired b should be negative.

### 4.1.2 The main LFMM script.

The main script is the normal entrypoint to all LFMM-calculations in DommiMOE. This is also the only SVL-script used in the software with everything else being functions. The main points of the script is shown in figure 4.1 with figure 4.2 being the processing for each iteration i of the loop shown in the first flowchart.

One of the functions depicted in figure 4.1 is the "validate parameters" with the following 3 lines describing this function:

** Parse and verify the command line arguments.
** Some of the arguments are optional. For any arguments not included
** the defaults in this function is used instead.

While the description is short the function contains close to 1000 lines of code and this makes it the largest function.

The function "load database(s)" in figure 4.1 will either read in a single database or if selected to only calculate the descriptors necessary for fitness it will read in both a lowspin and a highspin database and afterwards combine them into two. The code for this is not shown.

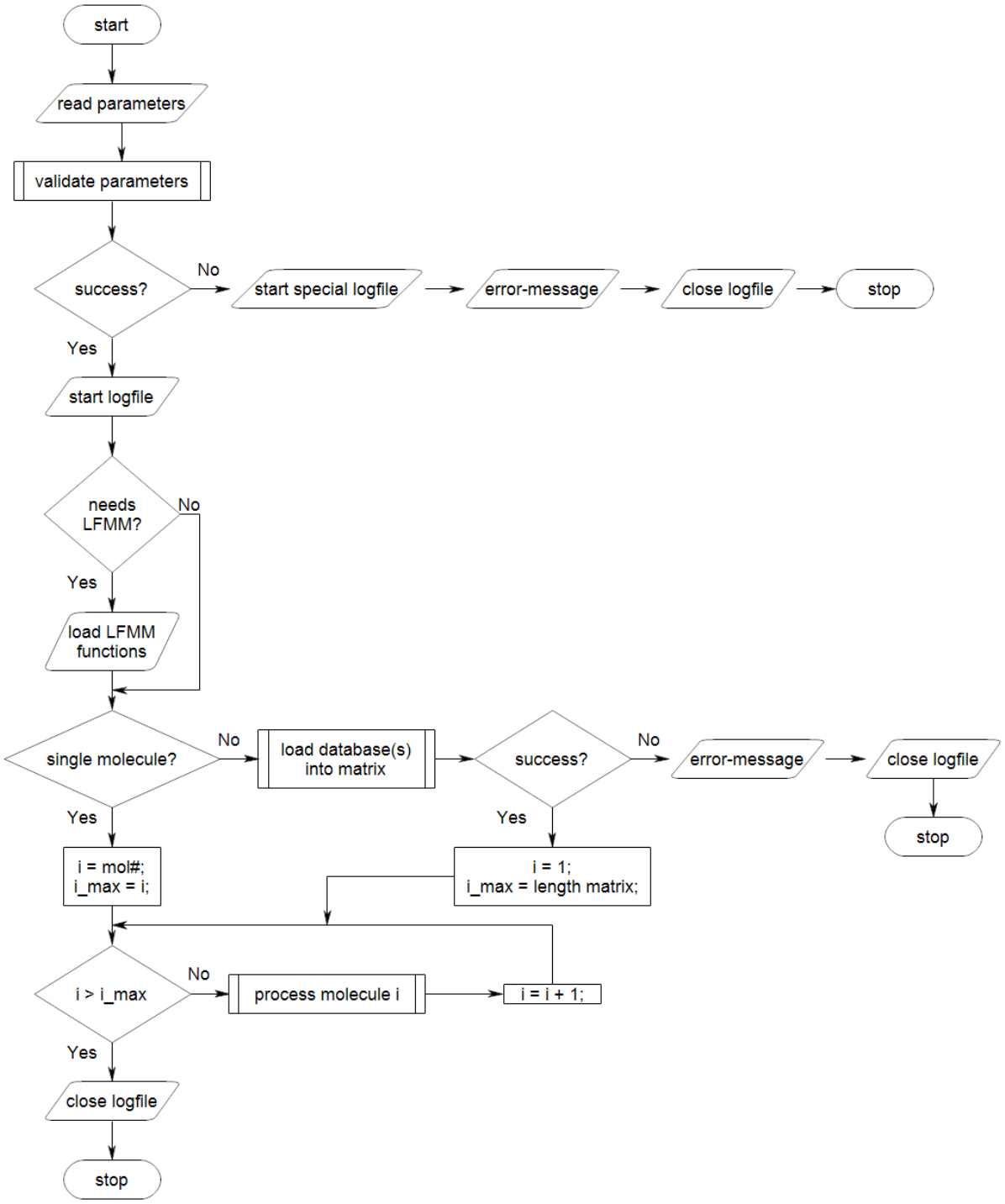


**Figure 4.1:** Flowchart showing the main parts of the MOE script normally controlling all processing of molecules.

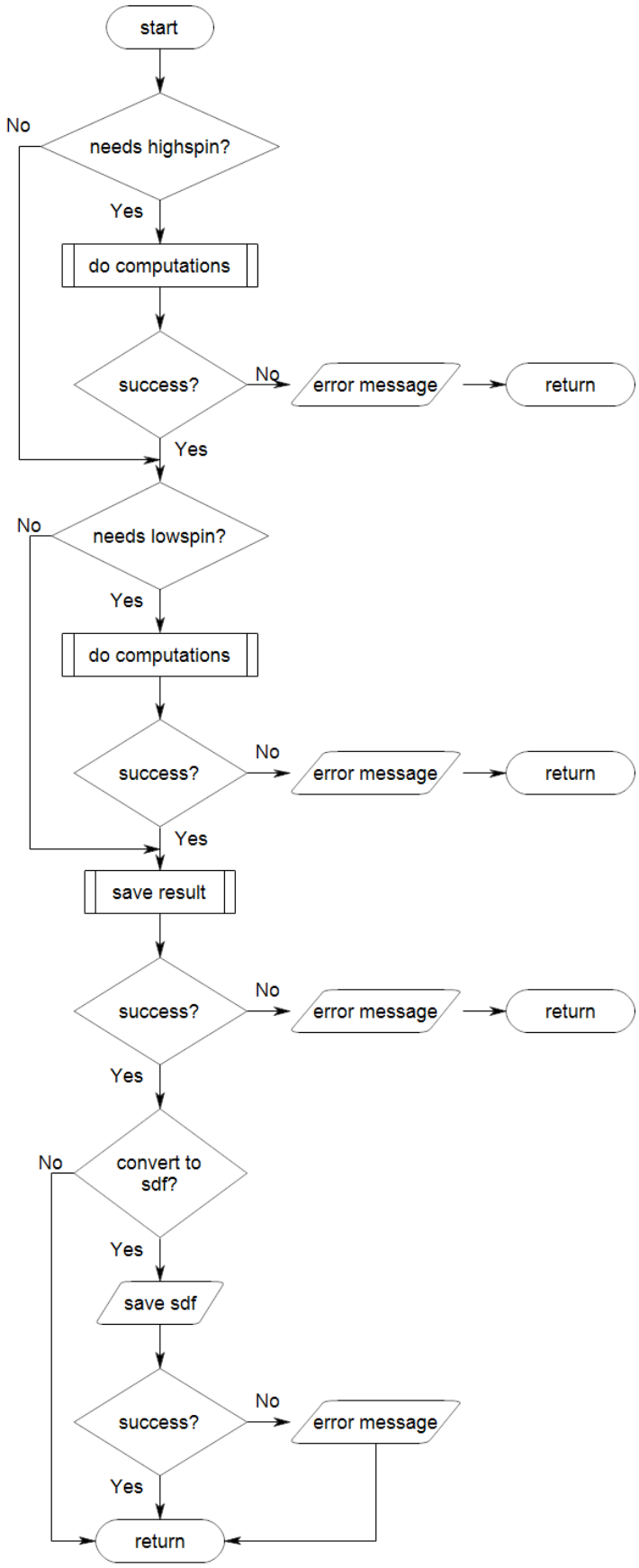


**Figure 4.2:** Flowchart showing the function "process molecule i" in the main MOE script.

### 4.1.3 The function marked "do computations" in figure 4.2:

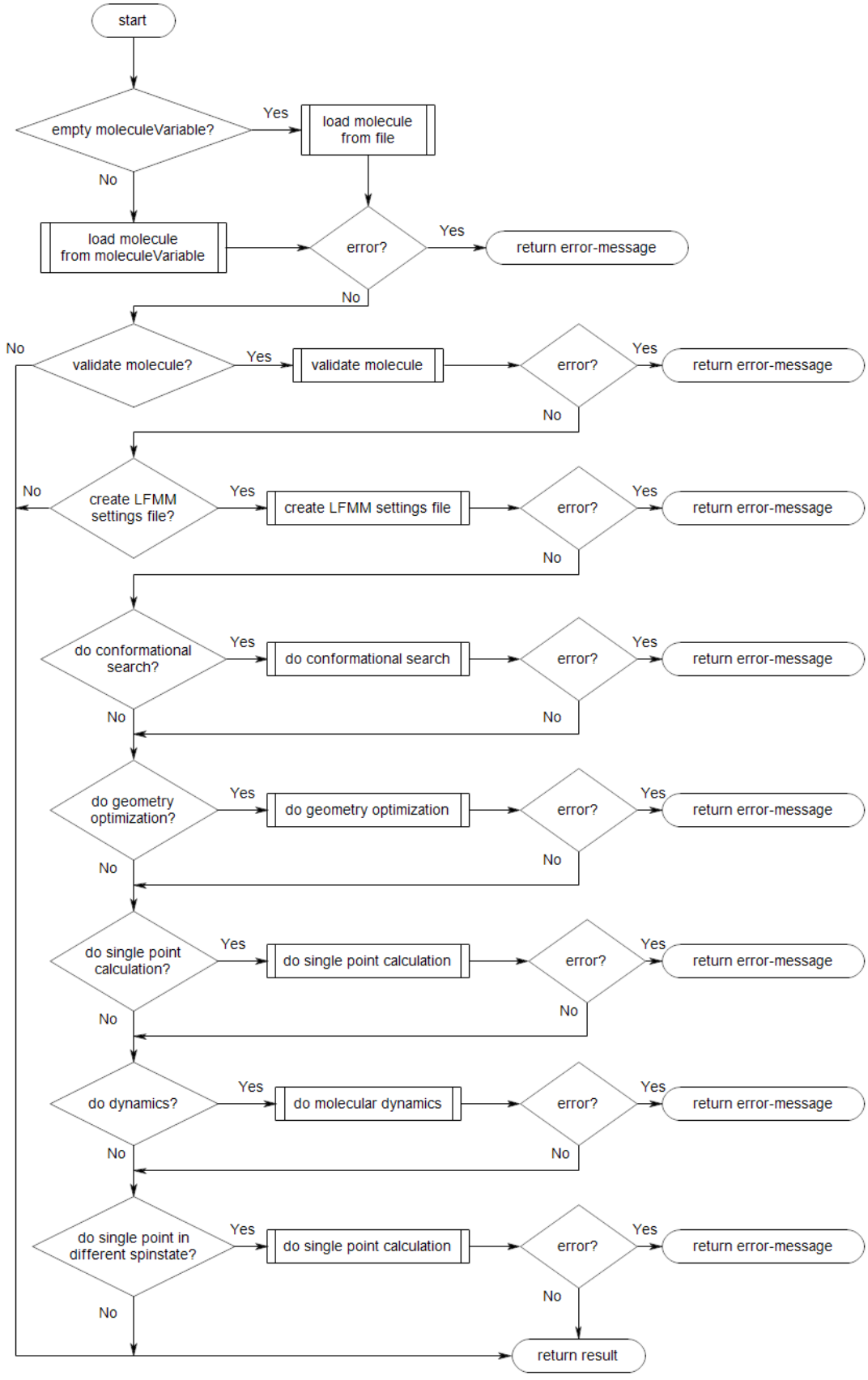


**Figure 4.3:** Flowchart showing the function included in the "do computations"-step of DommiMOE.

The "do computations" function is the main as far as doing all the various processes. This function is called once for highspin and once for lowspin unless no calculation for a spinstate is necessary. The main features of "do computations" is shown in figure 4.3. All the functions in figure 4.3 will be shown in next chapter.

### 4.1.4 The individual calculation steps as shown in figure 4.3

The first step on handling any molecule is to load it into MOE. The flowchart for this is shown in figure 4.4. Included in figure 4.4 is the function "remove previously loaded molecule(s)" and this function is included in full in appendix 9.1.

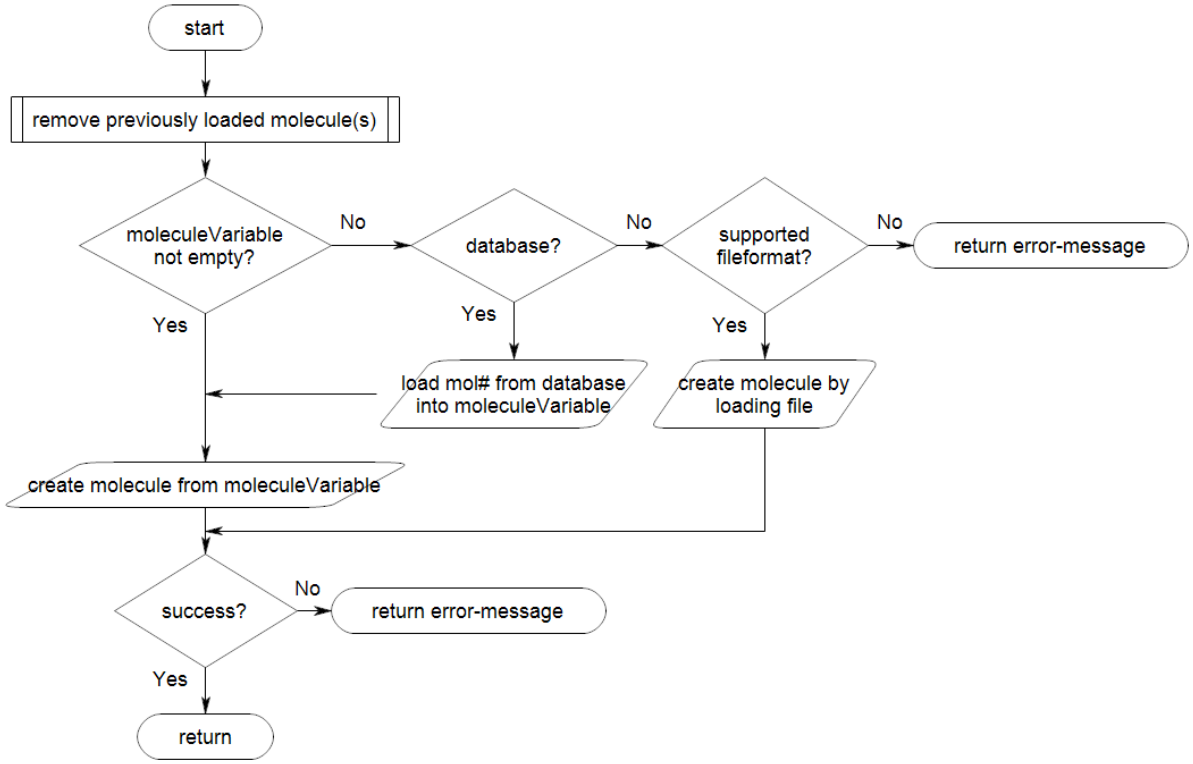


**Figure 4.4:** Flowchart showing the loading of a molecule into MOE. This can be done in many of the functions but normally the first occurrence is during the "do computations"-step.

After loading comes filtering of any invalid molecules and these steps is shown in figure 4.5 and figure 4.6. None of the substeps in either figure is included in a flowchart. A choise is included in case no processing is desired and in this case the filtering-step is skipped and instead returns back to the flowchart in figure 4.2.

An additional step needed before any LFMM-methods can be used is to create and save the LFMM settings-file, default called "settings.txt" if runs the graphical version of DommiMOE. This step is shown in figure 4.7. If all LFMM-methods is disabled this step is skipped and instead is returned back to the flowchart in figure 4.2.

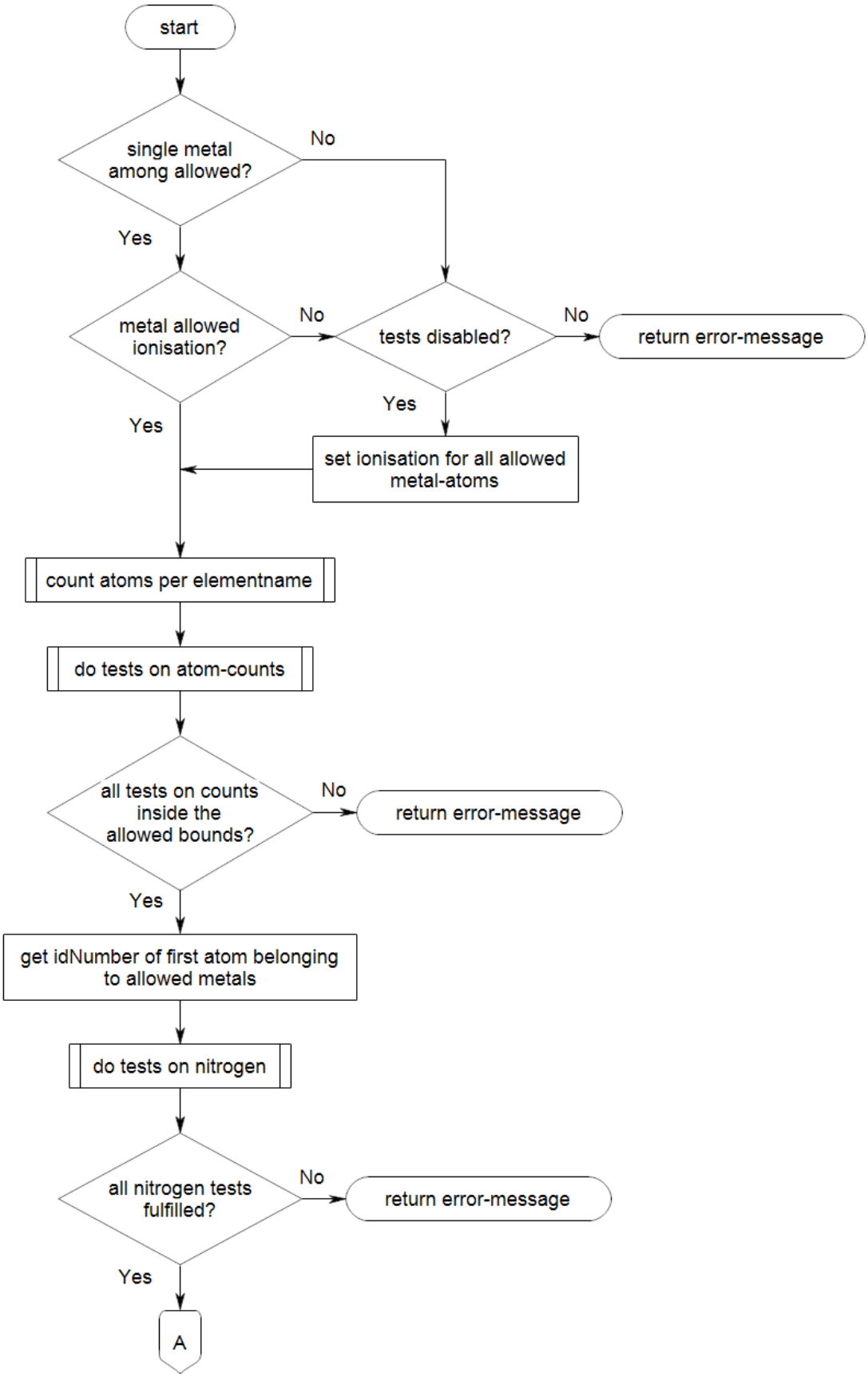


**Figure 4.5:** Flowchart showing the first half of the validation of molecules during the "do computation"-step.

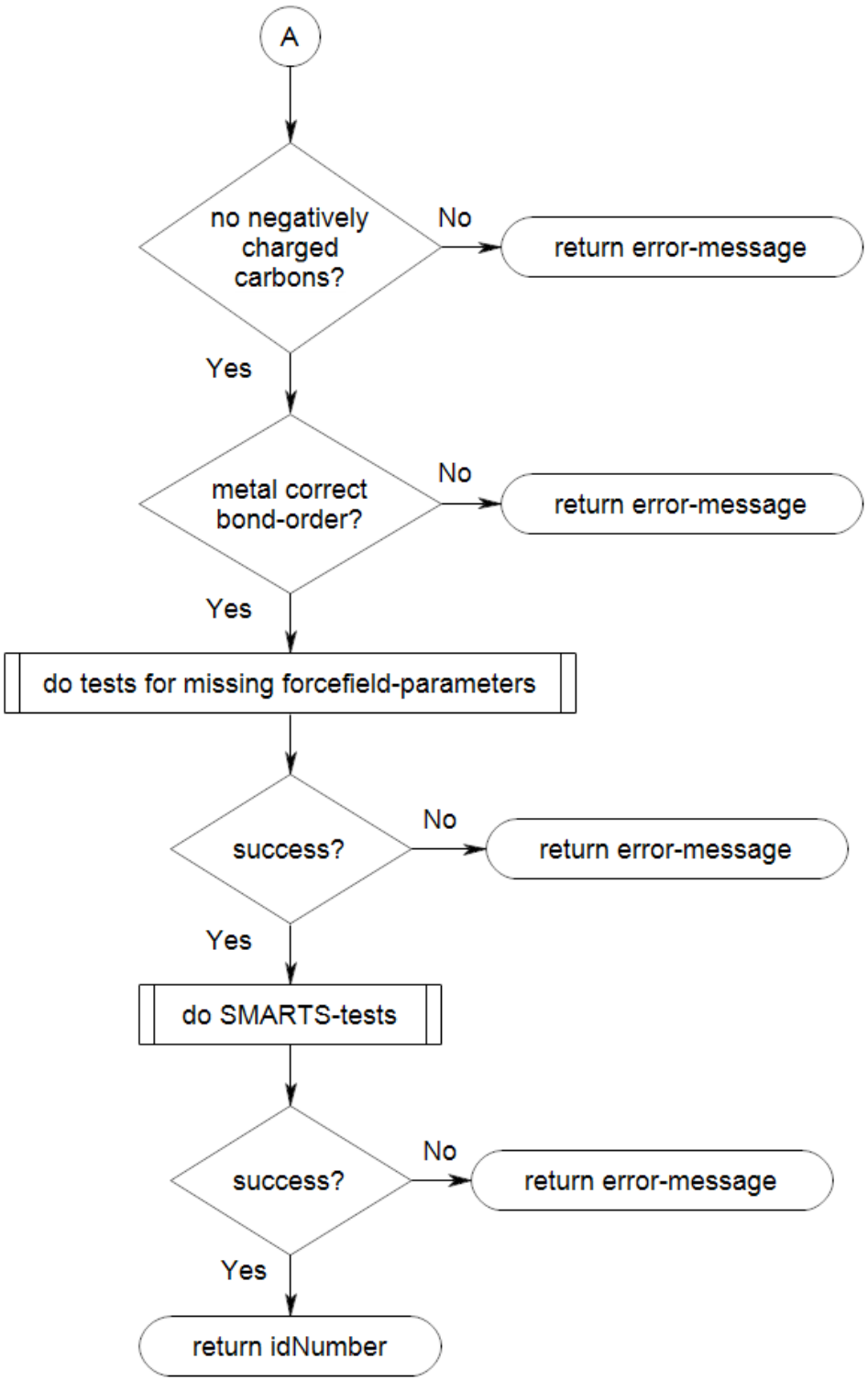


**Figure 4.6:** Flowchart showing the second half of the validation-function during the "do calculations"-step.

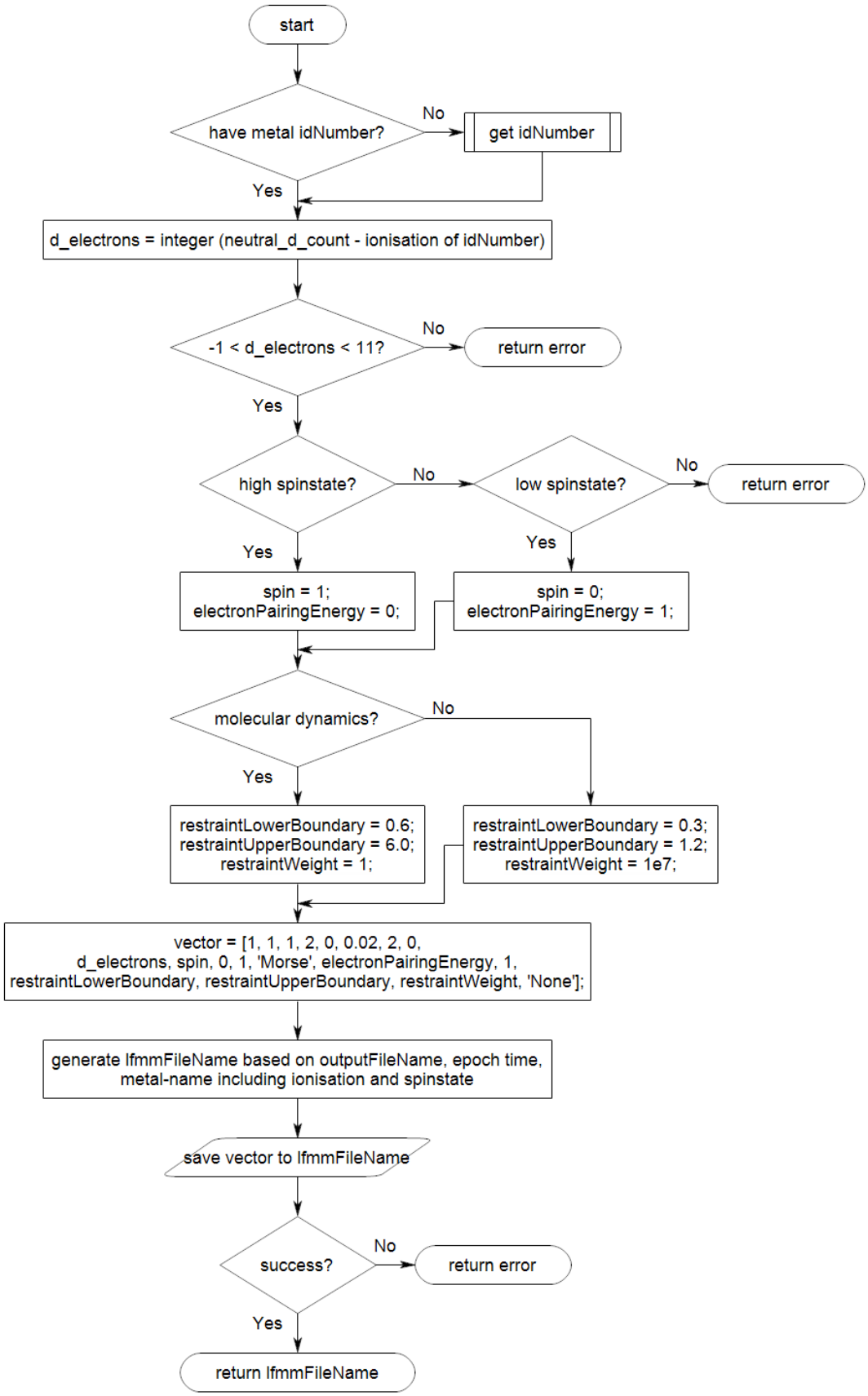


**Figure 4.7:** Flowchart showing the function responsible for the creation of the LFMM settings-file. The first place this will happen is during the "do calculations"-step.

After saving the LFMM settings-file the next step is often conformational search unless this step is skipped. The conformational search is shown in figure 4.8 and none of the functions shown here is commented any further.

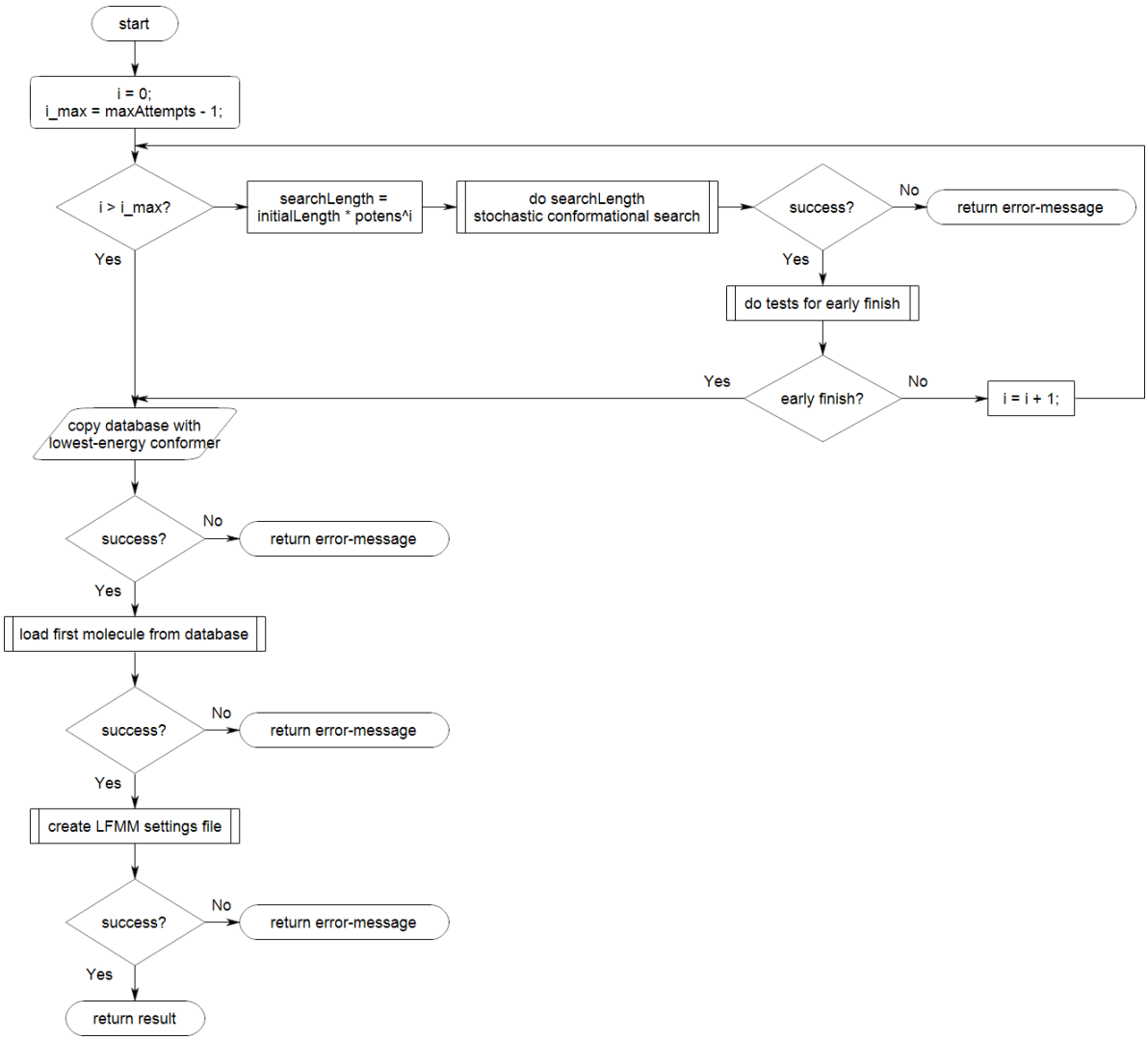


**Figure 4.8:** Flowchart showing the stochastic conformational step. This is part of the "do calculations"-step.

If MD is enabled the step with geometry optimization is always done but in other cases it is often skipped. The step with geometry optimization is shown in figure 4.9.

After a geometry optimization a single point calculation to extract the energy is done, unless MD is choosen. The single point calculation is shown in figure 4.10.

The last of the possible new steps is MD as shown in figure 4.11 and figure 4.12, since the single point calculation has already been shown before.

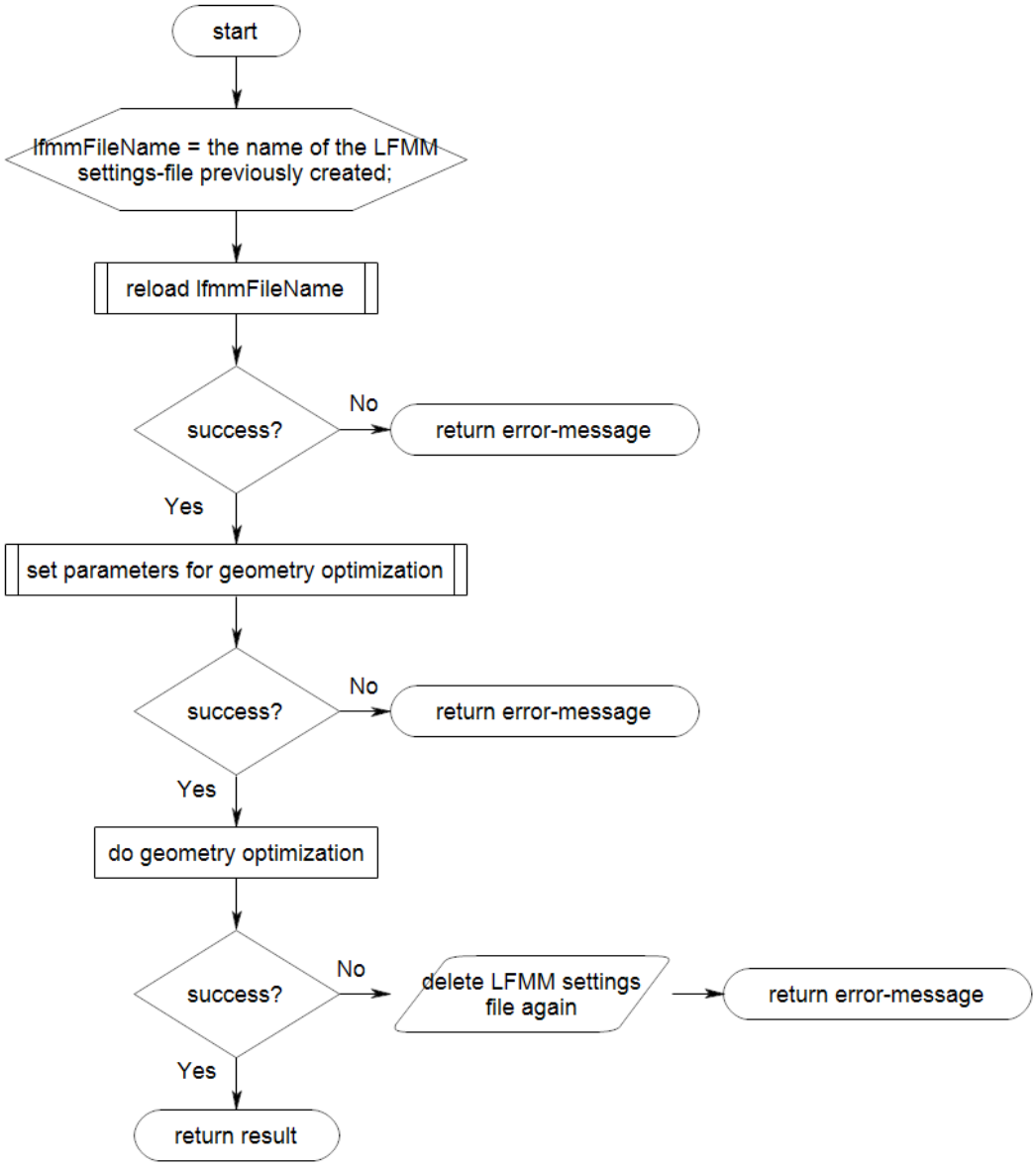


**Figure 4.9:** Flowchart showing the geometry optimization during the "do computations"-step.

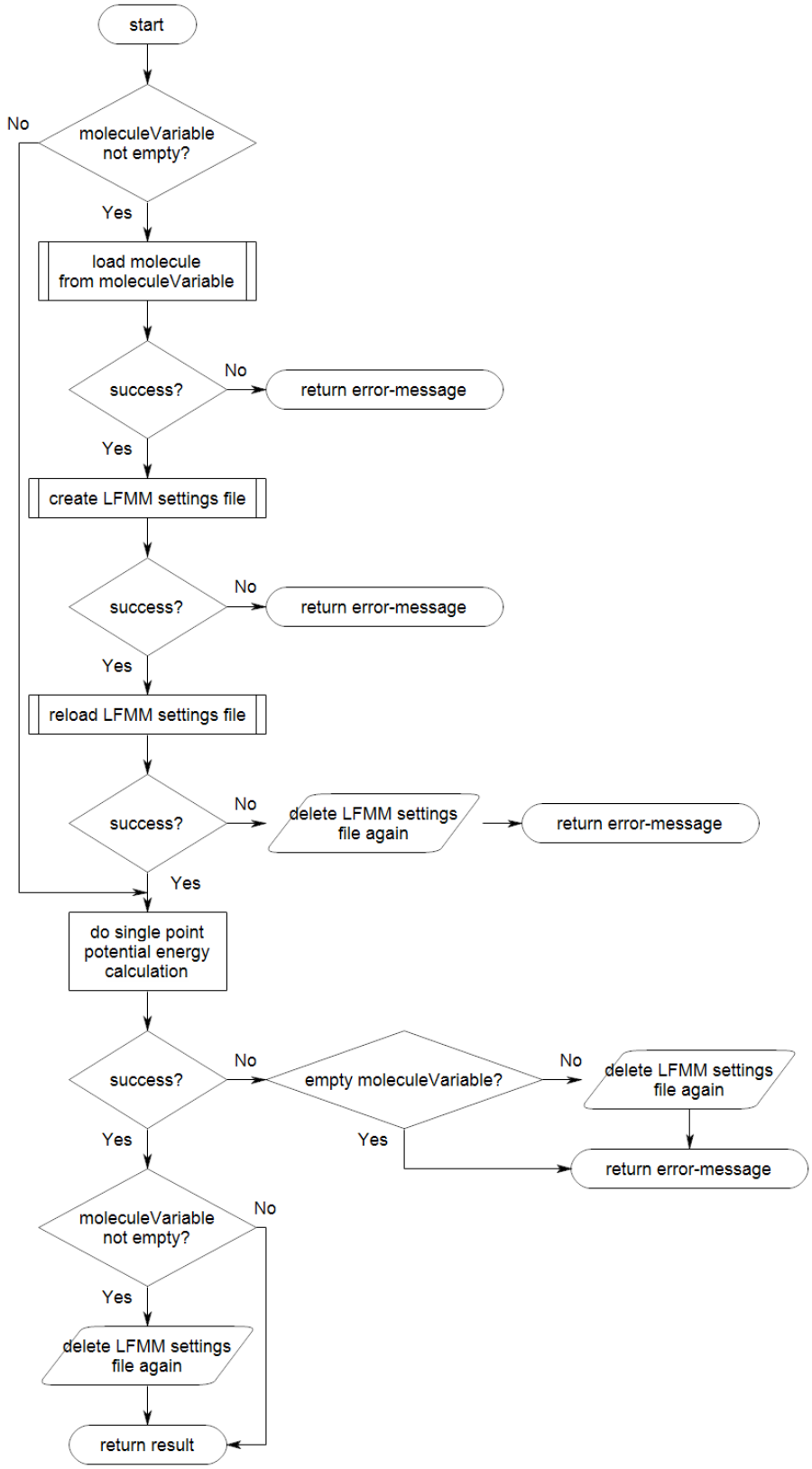


**Figure 4.10:** Flowchart showing the function for doing single point calculations. The function can load in a molecule before calculation if necessary. Calculation of single point is often done as part of conformational search but can also be done as part of the "do calculations"-step.

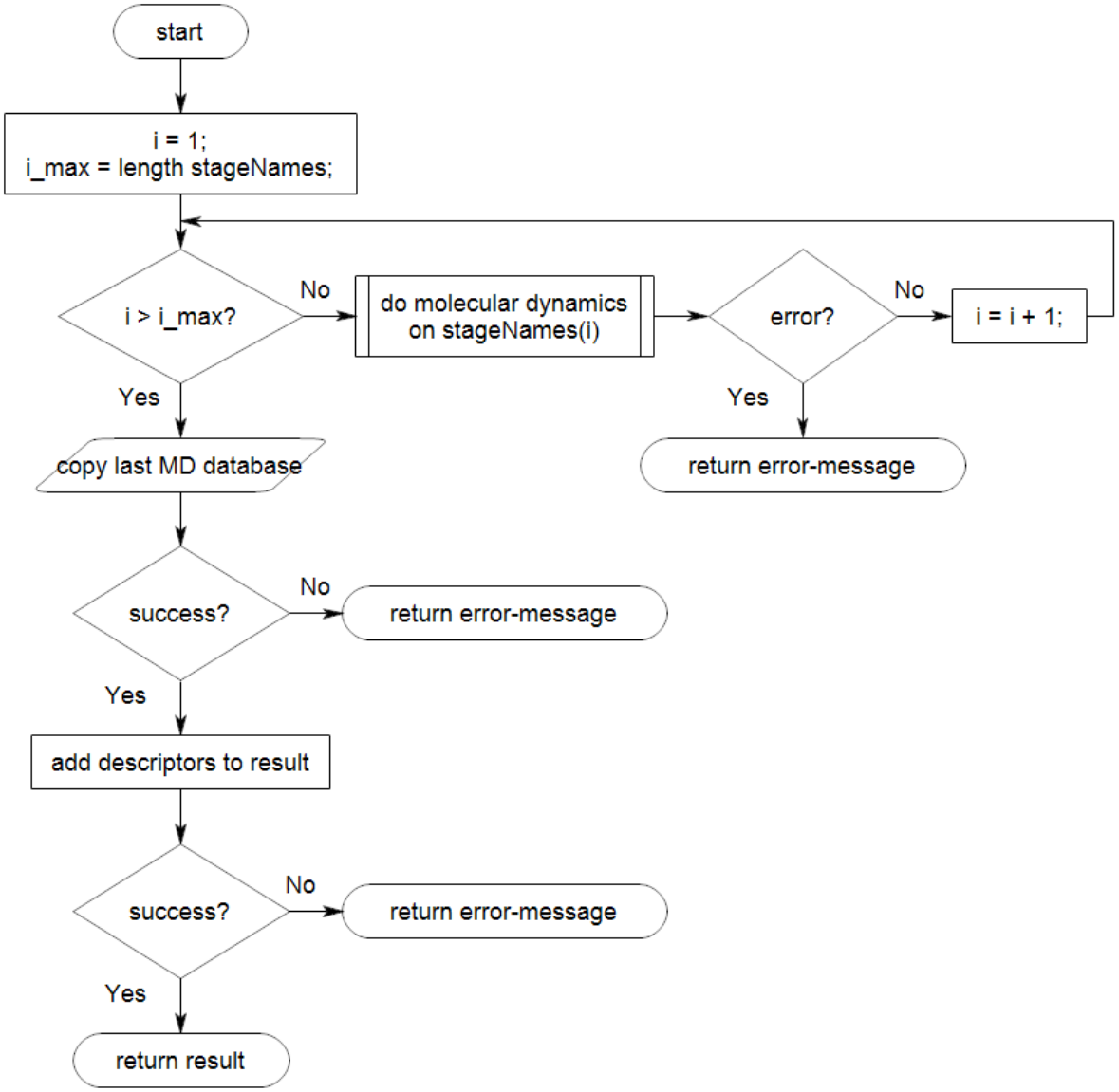


**Figure 4.11:** Flowchart showing the main loop for molecular dynamics. This is part of the "do computations"-step.

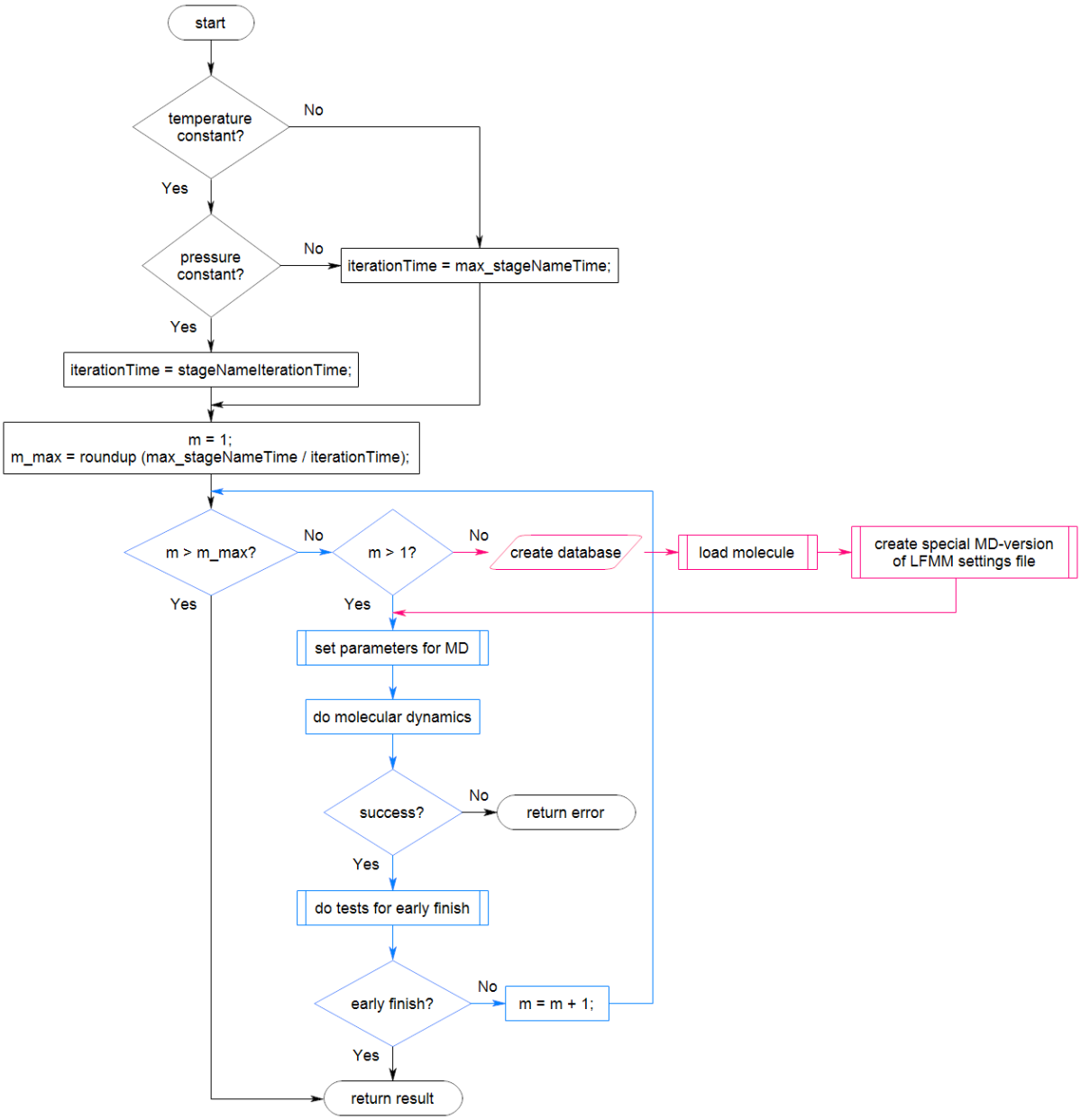


**Figure 4.12:** Flowchart showing the molecular dynamics steps as part of the "do molecular dynamics on stageNames(i)" part.

### 4.1.5 The main parts of the fitness-script for Denoptim.

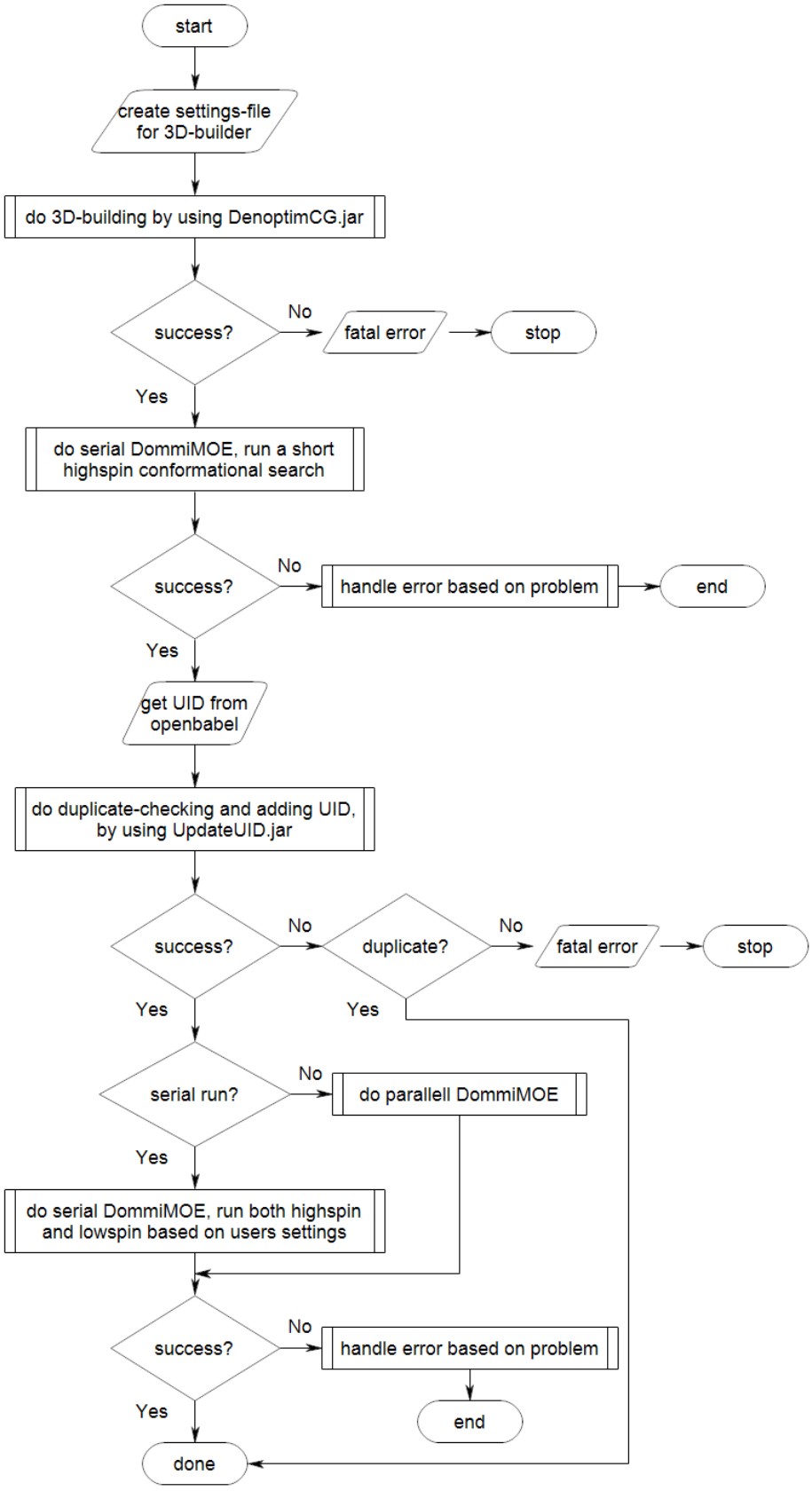


**Figure 4.13:** Flowchart of the main part of the DommiMOE fitness-script for Denoptim.

The main part of the fitness-script for Denoptim enabling DommiMOE to be a fitness-provider is shown in figure 4.13. For the main calculation-step there is a choise between running DommiMOE in parallel or serially. If the choise is to run in parallel the highspin calculation is run in one instance of DommiMOE while lowspin is run in in another and this is shown in figure 4.15. In case of serial this is shown in figure 4.14.

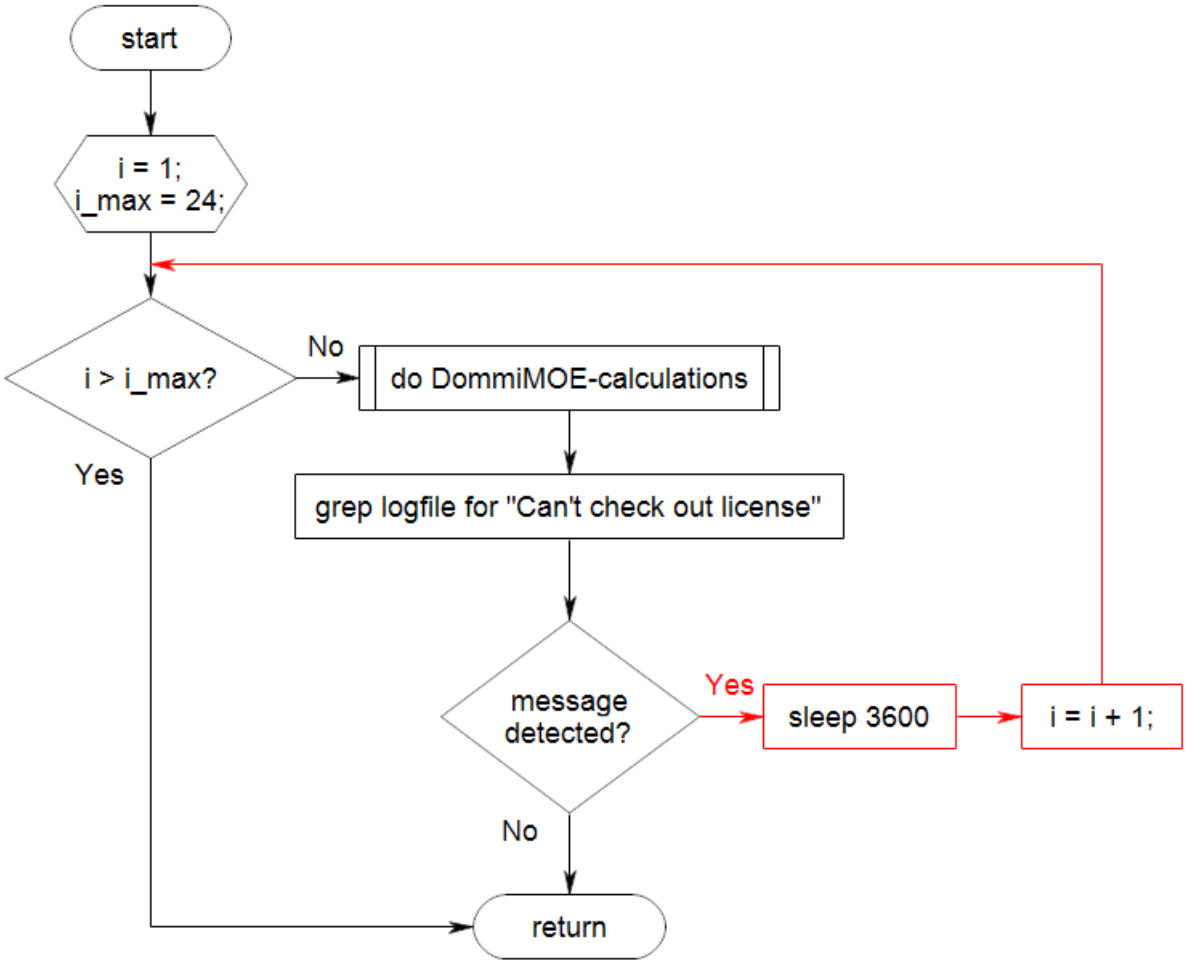


**Figure 4.14:** Flowchart showing the parallell choise in the DommiMOE in the fitness-script for Denoptim.

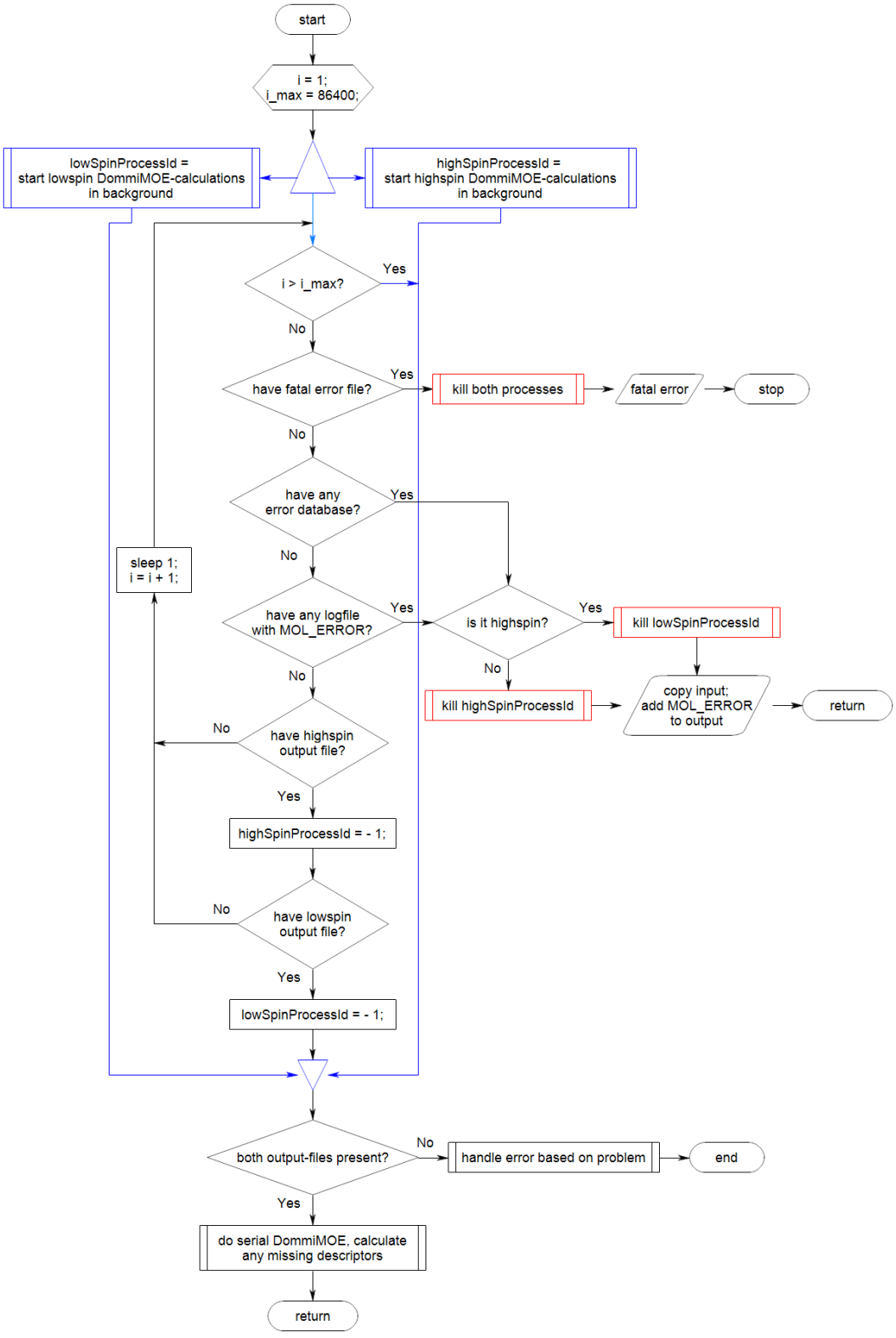


**Figure 4.15:** Flowchart showing the parallell choise in the DommiMOE fitness script for Denoptim.

### 4.2: Filtering, scaffolds, fragments and compability matrices.

The total count of 3D-structures in CCSD and some different filtering-criteria is shown in table 4.1. The options for CCSD and filtering is shown in computational details.

**Table 4.1:** Some results based on different methods to filtering CCSD. In all instances the option 3D coordinates determined was used.

| Method | count |
|---|---|
| CCSD no filtering except 3D-structure | 713715 |
| CCSD only iron | 39258 |
| CCSD $Fe^{2+}N_6$-single bond | 1779 |
| DommiMOE | 627 |
| Filtered by new code | 26 |

After the filtering 26 molecules was present and based on visual inspection all was valid. To help remove any duplicates visual inspection, molecular weight and the INCHI-key was used. For CCSD refcodes and INCHI-keys of the 26 molecules see table 4.2.

**Table 4.2:** The refcodes and the INCHI-keys of the molecules found after filtering the CCSD. For the filtering-options used see computational details. The INCHI-keys was generated after filtering with the help of openbabel.

| CCSD refcode | INCHI-key |
|---|---|
| PURYIK | LBHSFOSWIGOFGA-WTKWRTAFSA-N |
| HUMBEX | LKROZQMKPBJPHY-KAECKJJSSA-P |
| KIBWEZ | LSZLWUGRFLLXKT-UHFFFAOYSA-N |
| TIYPIC | LZFIZKVGVNHHIQ-UHFFFAOYSA-N |
| LIWKOR01 | NVPOVTDLXMOECW-UHFFFAOYSA-N |
| QAXDID | NVPOVTDLXMOECW-UHFFFAOYSA-N |
| QAXDOJ | NVPOVTDLXMOECW-UHFFFAOYSA-N |
| RITKUA | NVPOVTDLXMOECW-UHFFFAOYSA-N |
| TILQUB | NVPOVTDLXMOECW-UHFFFAOYSA-N |
| WOCPAF | NVPOVTDLXMOECW-UHFFFAOYSA-N |
| ZIWDUG | NVPOVTDLXMOECW-UHFFFAOYSA-N |
| XOZBEV | NVPOVTDLXMOECW-UHFFFAOYSA-N |
| FEBMAB | NVPOVTDLXMOECW-UHFFFAOYSA-N |
| IPIWAF | QEKUEESIVBHRSO-UHFFFAOYSA-N |
| PAZXAP | SHPCSTQSCKOTLV-GJTSMBTKSA-N |
| DETTOL | SQTLXNYBGRTGCC-UHFFFAOYSA-N |
| KIBVUO | SQTLXNYBGRTGCC-UHFFFAOYSA-N |
| LOTSES | VMTZSIDLBMKNEB-UHFFFAOYSA-N |
| CODQAO | WCZDZKLJPNLCER-UHFFFAOYSA-N |
| LEVQAF | WCZDZKLJPNLCER-UHFFFAOYSA-N |
| QIBZUX | WCZDZKLJPNLCER-UHFFFAOYSA-N |
| RENHUO | WCZDZKLJPNLCER-UHFFFAOYSA-N |
| XUZVAP | WCZDZKLJPNLCER-UHFFFAOYSA-N |
| GOXSET | WCZDZKLJPNLCER-UHFFFAOYSA-N |
| EYEWOV | WCZDZKLJPNLCER-UHFFFAOYSA-N |
| KIBWAV | WUJINVAGFPJETP-UHFFFAOYSA-N |

Among the 26 possible molecules in table 4.2 the 10 molecules shown in figure 4.16, 4.17 and 4.18 was picked as the basis for making scaffolds.

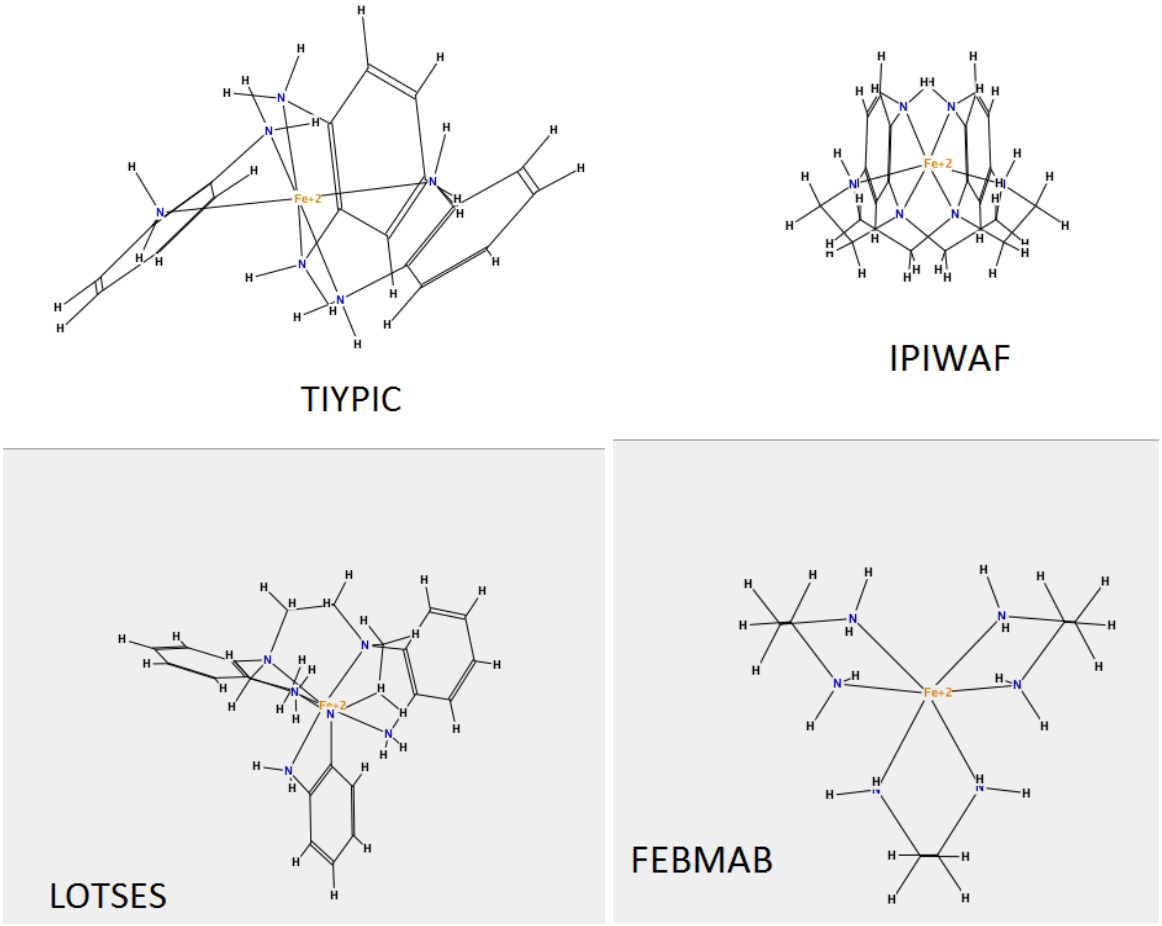


**Figure 4.16:** This is four of the molecules used as scaffold. They have CCSD refcodes FEBMAB, IPIWAF, LOTSES and TIYPIC.

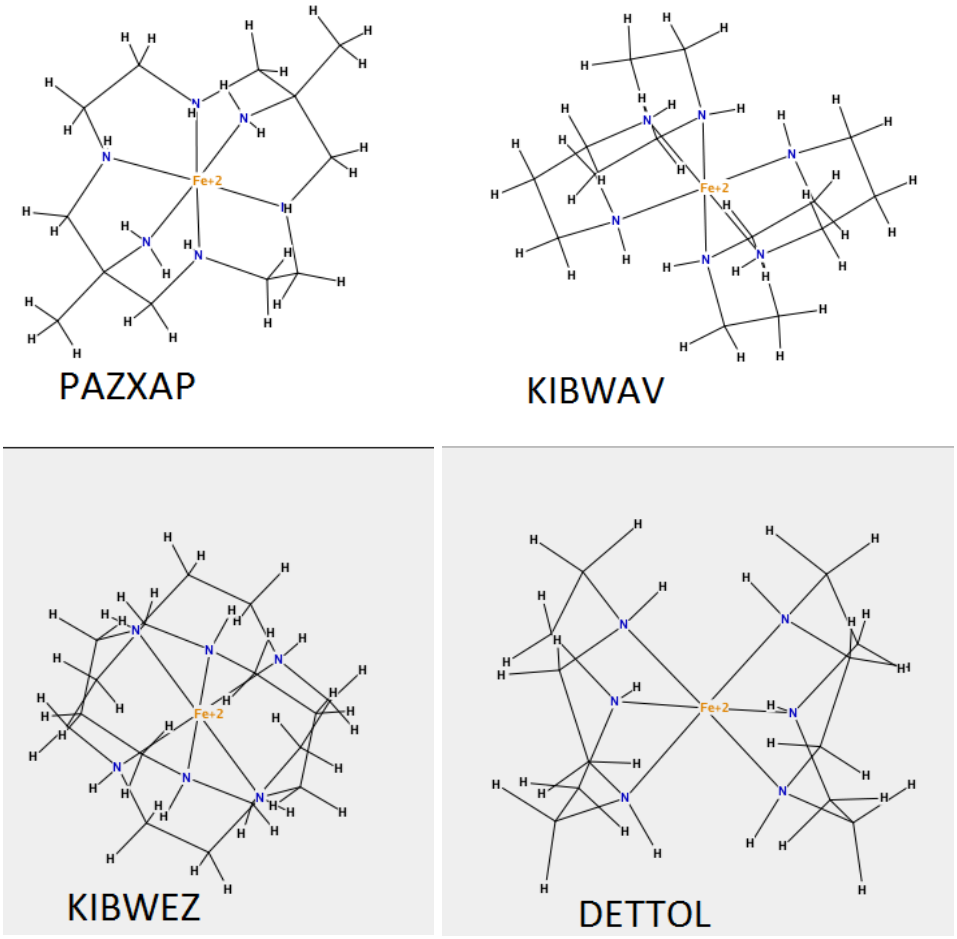


**Figure 4.17:** This is four of the molecules used as scaffold. They have CCSD refcodes DETTOL, KIBWAV, KIBWEZ and PAZXAP.

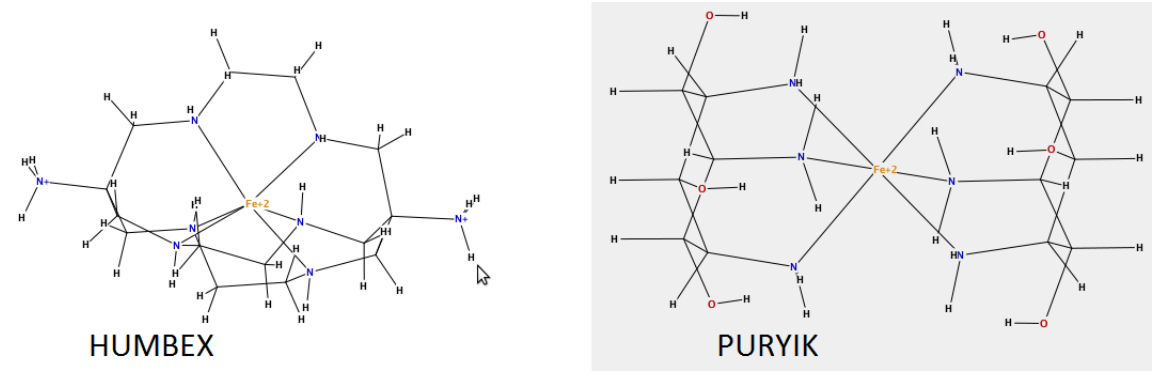


**Figure 4.18:** This is four of the molecules used as scaffold. They have CCSD refcodes DETTOL, KIBWAV, KIBWEZ and PAZXAP.

In addition to molecules usable as scaffolds some molecules usable as fragments was made. These are shown in figure 4.19.

For the compability matrices, see appendix 9.2.

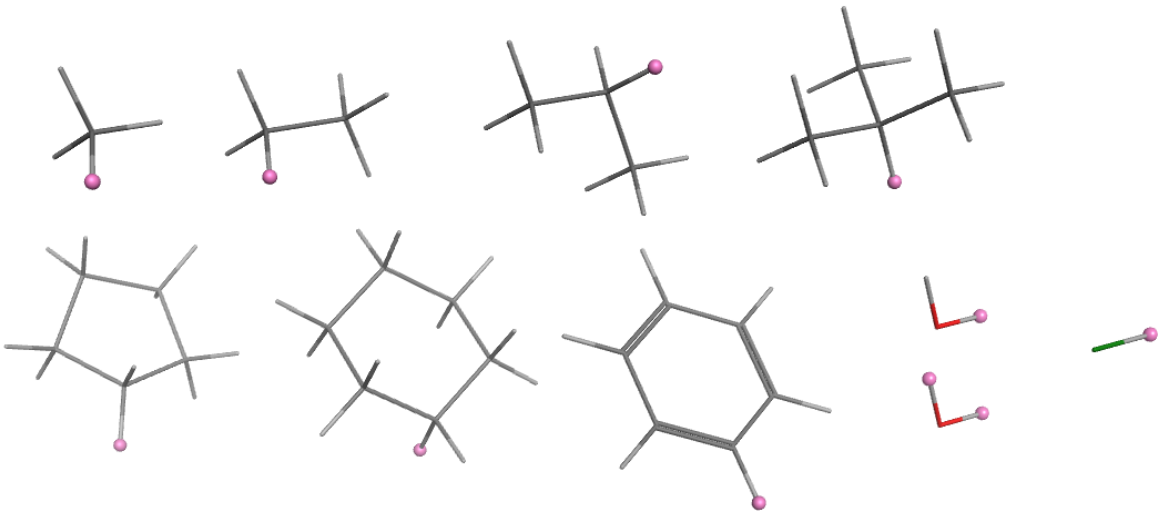

**Figure 4.19:** This is the collection of fragments manually built in MOE. On each molecule the hydrogen(s) highlighted pink shows the connection-point(s) to other fragments. The green atom is chlorine while the red is oxygen. For the oxygen molecules the angle is just an artefact. Additionally the last oxygen-molecule is ether and is monodentate in the binding to another fragment.

### 4.3 Conformational searches and molecular dynamics on ten molecules:

On the ten selected molecules also used for scaffolds as depicted in figure 4.16, 4.17 and 4.18 two different sets of conformational searches was done. The first was independent searches with a given length while the second was multi-stages as shown in flowchart in figure 4.8 by setting max_steps > 1. In this instance the exponent was kept at 1. Additionally the RMSD was calculated on all combinations for each individual molecule. An overview of all results is shown in table 4.3. A more detailed look on some selected results is shown in the following tables, graphs and pictures while more results is found in appendix 9.3.

**Table 4.3:** An overview of all conformational searches on the different scaffolds with CCSD refcodes as given. Single stands for single-stage conformational search while multi-stage is multi-stage conformational search. Where known from reference the difference in potential energy between highspin and lowspin is shown and additionally the known spinstate if known. Also included is link to the reference with energy and/or known spinstate.

| | | Highspin [kJ/mol] | | Lowspin [kJ/mol] | | HS - LS [kJ/mol] | | Energy from linked reference HS - LS [kJ/mol] | Spinstate (if known) |
|---|---|---|---|---|---|---|---|---|---|
| | | Average | std.dev | average | std.dev | average | std.dev | | |
| **dettol** | single | -188.76 | 5.74 | -193.74 | 6.49 | 4.97 | 5.71 | 6.276[39] | LS |
| | multi | -188.26 | 6.10 | -194.86 | 5.67 | 6.60 | 6.37 | | |
| **febmab** | single | -739.78 | 0.002 | -710.85 | 0.002 | -28.93 | 0.004 | -19.6648[39] | HS |
| | multi | -739.78 | 0.002 | -710.86 | 0.001 | -28.93 | 0.002 | | |
| **humbex** | single | -179.36 | 0.002 | -179.67 | 19.66 | 0.31 | 19.66 | 6.6944[39]* | LS |
| | multi | -177.86 | 7.52 | -185.85 | 0.002 | 8.00 | 7.52 | | |
| **ipiwaf** | single | -325.33 | 0.003 | -263.18 | 0.003 | -62.15 | 0.003 | | |
| | multi | -325.33 | 0.002 | -260.55 | 9.09 | -64.78 | 9.09 | | |
| **kibwav** | single | -202.20 | 10.49 | -177.89 | 14.70 | -24.31 | 11.72 | | LS[40] |
| | multi | -201.91 | 9.56 | -183.39 | 11.76 | -18.51 | 13.92 | | |
| **kibwez** | single | -149.16 | 12.02 | -66.11 | 6.37 | -83.05 | 9.24 | | HS[40] |
| | multi | -154.62 | 6.72 | -66.32 | 6.17 | -88.30 | 8.75 | | |
| **lotses** | single | -319.57 | 0.004 | -307.84 | 0.003 | -11.73 | 0.005 | -13.3888[39] | |
| | multi | -319.57 | 0.002 | -307.84 | 0.002 | -11.73 | 0.003 | | |
| **pazxap** | single | -164.31 | 0.002 | -124.18 | 1.78 | -40.12 | 1.78 | 60.2496[39] | LS |
| | multi | -164.31 | 0.001 | -122.51 | 1.41 | -41.80 | 1.41 | | |
| **puryik** | single | -383.27 | 2.20 | -368.54 | 3.08 | -14.73 | 3.57 | -6.6944[39] | HS |
| | multi | -383.21 | 2.25 | -368.62 | 2.21 | -14.59 | 2.27 | | |
| **tiypic** | single | -869.18 | 0.002 | -855.86 | 0.003 | -13.33 | 0.003 | | HS[41] |
| | multi | -869.19 | 0.002 | -855.86 | 0.002 | -13.32 | 0.002 | | |

* Note the reference is without $(NH_3)_2$.

The first molecule has CCSD refcode KIBWAV and both types of conformational searches is shown in table 4.4 and 4.5. A plot of the highspin and lowspin potential energy from table 4.4 is shown in figure 4.20. A superposition of two different conformational searches in lowspin state is shown in figure 4.21.

**Table 4.4:** The potential energies and the descriptor called FITNESS based on running multiple independent stochastic conformational searches on the molecule with CCSD refcode KIBWAV. Also includes is RMSD calculated with the help of openbabel with SMARTS [Fe](N)(N)(N)(N)(N)(N)

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -185.10 | -99.75 | -200.51 | -97.16 | -15.40 | -11.37 | 0.271 |
| 200 | -158.08 | -78.04 | -178.81 | -66.62 | -20.73 | -12.24 | 0.242 |
| 300 | -166.92 | -74.72 | -191.15 | -76.81 | -24.23 | -13.27 | 0.293 |
| 400 | -144.07 | -62.88 | -191.16 | -76.79 | -47.09 | -16.56 | 0.254 |
| 500 | -166.93 | -74.09 | -191.16 | -77.00 | -24.23 | -13.28 | 0.292 |
| 600 | -168.90 | -92.81 | -211.38 | -103.94 | -42.48 | -15.30 | 0.291 |
| 700 | -181.26 | -86.94 | -190.38 | -74.66 | -9.12 | -11.86 | 0.279 |
| 800 | -170.70 | -88.74 | -212.86 | -109.90 | -42.16 | -15.24 | 0.258 |
| 900 | -170.70 | -88.86 | -199.19 | -91.82 | -28.49 | -13.14 | 0.271 |
| 1000 | -166.93 | -74.63 | -199.28 | -90.42 | -32.35 | -14.18 | 0.284 |
| 1500 | -194.95 | -98.59 | -200.51 | -97.12 | -5.57 | -11.04 | 0.243 |
| 2000 | -181.26 | -86.97 | -212.85 | -110.03 | -31.59 | -13.88 | 0.258 |
| 2500 | -181.26 | -87.09 | -212.86 | -109.95 | -31.59 | -13.88 | 0.258 |
| 3000 | -194.95 | -98.38 | -211.36 | -104.52 | -16.41 | -12.15 | 0.270 |
| 3500 | -194.95 | -98.60 | -212.85 | -109.82 | -17.91 | -12.11 | 0.238 |
| 4000 | -194.95 | -99.01 | -211.38 | -103.73 | -16.43 | -12.16 | 0.274 |
| 4500 | -185.10 | -99.53 | -200.51 | -96.89 | -15.41 | -11.39 | 0.271 |
| 5000 | -194.95 | -98.55 | -211.38 | -103.73 | -16.44 | -12.18 | 0.275 |

**Table 4.5:** The potential energies and the descriptor called FITNESS based on running multistep method of stochastic conformational searches on the molecule with CCSD refcode KIBWAV. The column with "needs # good" shows how many results close to the lowest potential energy must be present before not doing an additional conformational search iteration. If method is old it is also possible to stop doing an additional conformational search iteration in cases very few conformers was found while this option is removed if method is new. "tight" stands for only allowing cutoff if energy within 0.04 kJ/mol from minimum while the rest allows 0.41 kJ/mol deviation from the minimum.

| Name | # steps | needs # good | method | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS |
|---|---|---|---|---|---|---|---|---|---|
| 100_2_old | 100 | 2 | old | -185.11 | -99.71 | -185.50 | -73.55 | -0.40 | -10.76 |
| 100_2_new | 100 | 2 | new | -194.95 | -98.57 | -199.28 | -90.56 | -4.34 | -11.26 |
| 100_3_old | 100 | 3 | old | -181.26 | -87.09 | -191.16 | -76.92 | -9.90 | -11.83 |
| 100_3_new | 100 | 3 | new | -185.11 | -99.51 | -191.16 | -76.69 | -6.05 | -11.19 |
| 100_3_new_tight | 100 | 3 | new | -194.95 | -98.37 | -211.39 | -103.69 | -16.44 | -12.19 |
| 200_2_old | 200 | 2 | old | -170.70 | -88.74 | -181.12 | -29.75 | -10.42 | -13.60 |
| 200_2_new | 200 | 2 | new | -159.77 | -81.98 | -199.19 | -91.56 | -39.42 | -14.81 |
| 200_3_old | 200 | 3 | old | -185.10 | -99.88 | -191.15 | -76.90 | -6.05 | -11.16 |
| 200_3_new | 200 | 3 | new | -185.11 | -99.36 | -199.28 | -90.55 | -14.18 | -11.54 |
| 200_3_new_tight | 200 | 3 | new | -181.26 | -87.17 | -212.86 | -110.18 | -31.59 | -13.86 |
| 300_2_old | 300 | 2 | old | -185.10 | -99.51 | -200.51 | -96.68 | -15.40 | -11.40 |
| 300_2_new | 300 | 2 | new | -194.95 | -98.63 | -211.38 | -103.96 | -16.43 | -12.17 |
| 300_3_old | 300 | 3 | old | -185.10 | -99.59 | -211.38 | -103.63 | -26.28 | -12.97 |
| 300_3_new | 300 | 3 | new | -194.95 | -98.44 | -211.38 | -103.82 | -16.44 | -12.18 |
| 300_3_new_tight | 300 | 3 | new | -181.26 | -87.08 | -212.85 | -109.93 | -31.59 | -13.88 |
| 400_2_old | 400 | 2 | old | -168.90 | -92.27 | -199.28 | -90.54 | -30.39 | -13.32 |
| 400_2_new | 400 | 2 | new | -158.08 | -78.41 | -212.86 | -110.10 | -54.78 | -17.57 |
| 400_3_old | 400 | 3 | old | -194.95 | -98.68 | -191.16 | -76.88 | 3.80 | -11.31 |
| 400_3_new | 400 | 3 | new | -194.95 | -98.76 | -211.39 | -103.54 | -16.44 | -12.18 |
| 400_3_new_tight | 400 | 3 | new | -185.11 | -99.79 | -211.39 | -103.59 | -26.28 | -12.96 |
| 500_2_old | 500 | 2 | old | -185.11 | -99.81 | -200.39 | -4.56 | -15.28 | -16.94 |
| 500_2_new | 500 | 2 | new | -185.11 | -99.47 | -199.28 | -90.45 | -14.17 | -11.54 |
| 500_3_old | 500 | 3 | old | -194.95 | -98.39 | -200.51 | -97.16 | -5.56 | -11.05 |
| 500_3_new | 500 | 3 | new | -158.07 | -78.10 | -200.49 | -97.05 | -42.42 | -15.25 |
| 500_3_new_tight | 500 | 3 | new | -194.95 | -98.32 | -211.38 | -103.64 | -16.43 | -12.20 |

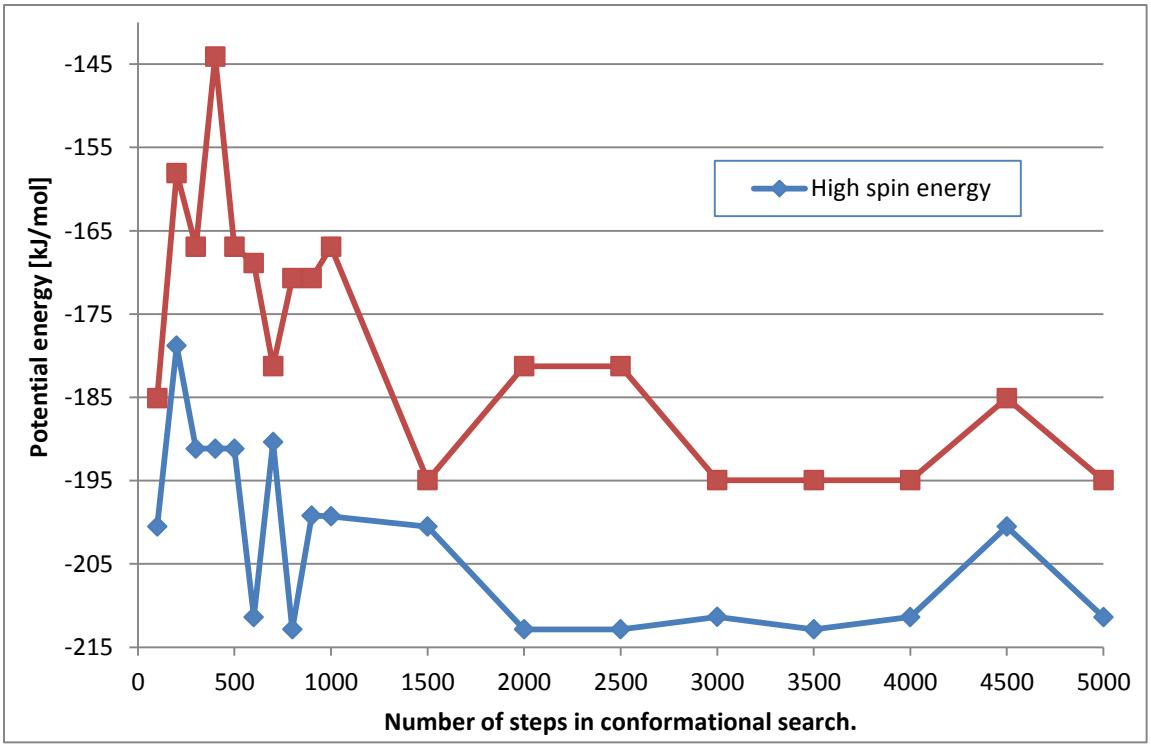


**Figure 4.20:** A plot between conformational search limit and potential energy for both highspin and lowspin states. This is for the molecule with CCSD refcode KIBWAV. The lines between points is to easier show the variation between different conformational searches.

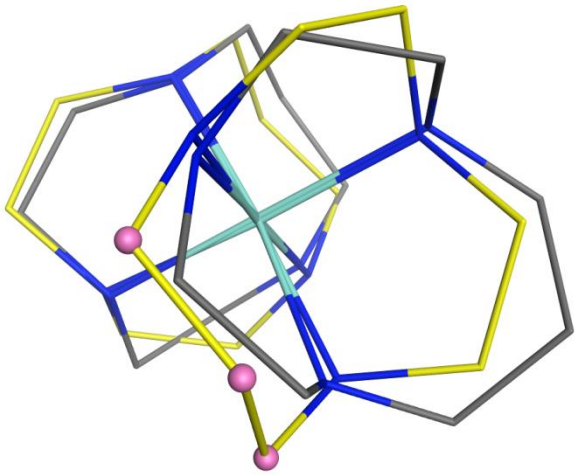

**Figure 4.21:** Two superposed molecules based on different conformational searches in the lowspin state. Molecule has CCSD refcode KIBWAV. The colour-coding is light blue for iron, dark blue for nitrogen, yellow for carbon in the 1000 step conformational search and grey for carbon in the 4500 step conformational search. Hydrogens is intentially hidden to easier reveal the structure. The three highlighted pink atoms shows a six-ring is superposed against a five-ring in the other molecule. The RMSD between molecules by using SMARTS [Fe](N)(N)(N)(N)(N)(N) are 0.1829 Å.

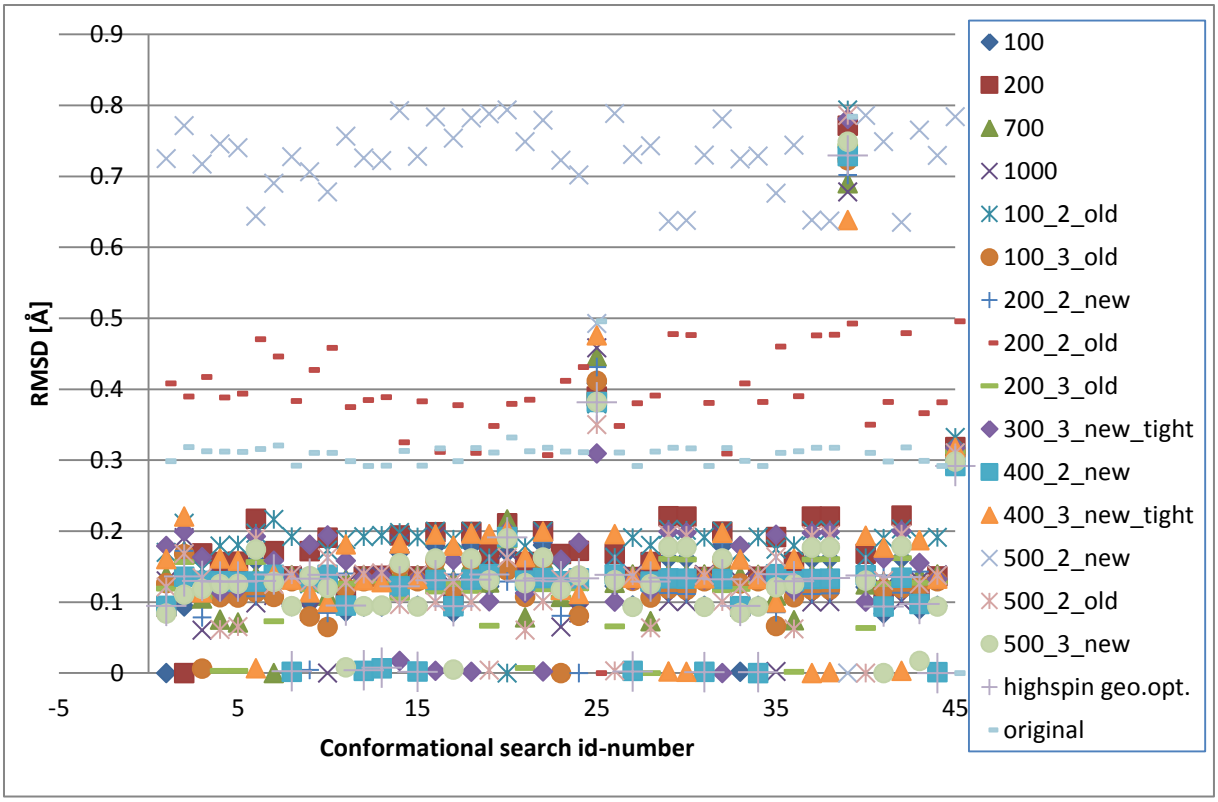


**Figure 4.22:** The RMSD with the highspin conformers of molecule with CCSD refcode KIBWAV is paired against eachother. The Conformational search id-number is 1-18 for single-conformers 100 – 5000 while 19-43 is multistep same order as table 4.5.

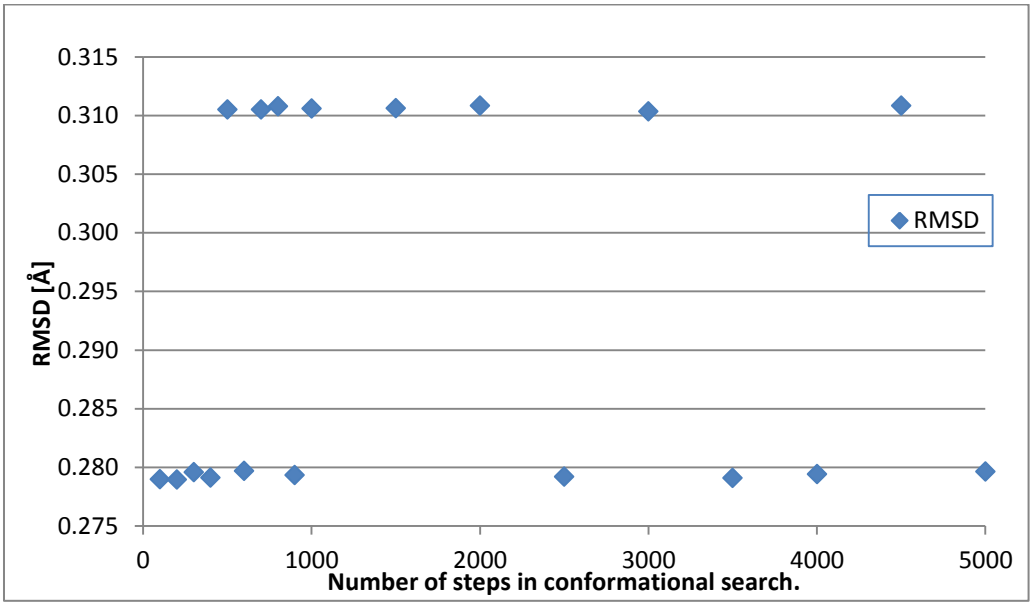


**Figure 4.23:** A plot between conformational search limit and RMSD between the lowpin and highspin molecules geometry. This is for the molecule with CCSD refcode TIYPIC. The RMSD is based on using the SMARTS [Fe](N)(N)(N)(N)(N)(N) in openbabel.

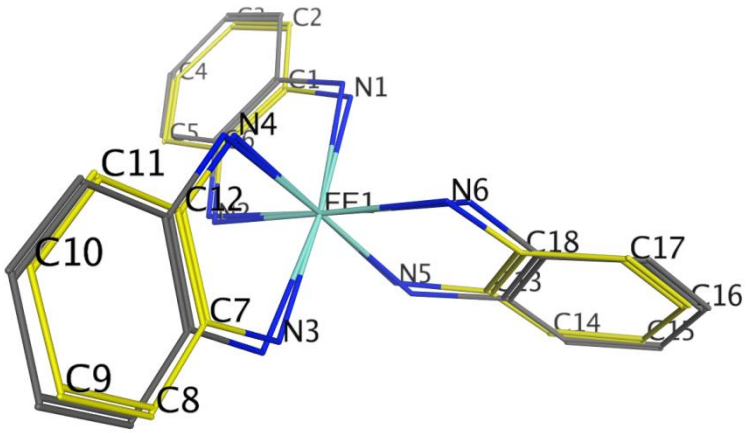


**Figure 4.24:** Two superposed molecules based on different conformational searches comparing lowspin against highspin. Molecule has CCSD refcode TIYPIC and is based on conformational search 400_3_new. The colour-coding is light blue for iron, dark blue for nitrogen, yellow for carbon in lowspin state and grey in highspin state. The RMSD between molecules are 0.279318 Å. The distance between Fe-C7 in lowspin is 2.855Å while in highspin it is 3.125 Å. In lowspin Fe-C9 is 5.222 Å and for higspin it is 5.508 Å.

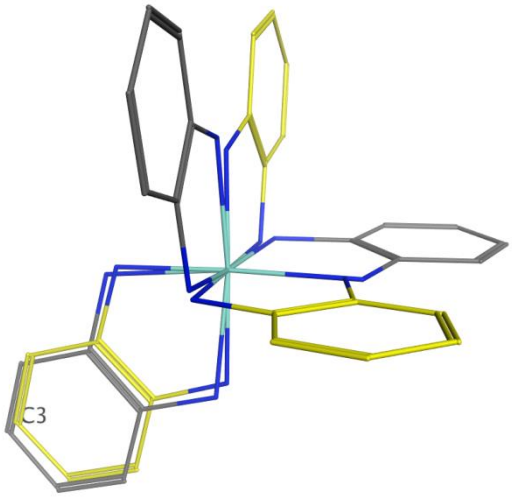


**Figure 4.25:** Two superposed molecules based on comparing lowspin against highspin. Molecule has CCSD refcode TIYPIC and is based on conformational search 1000. The RMSD is 0.310567 Å. C3 is only place there are ring-overlap between the superposed molecules. Colour-code is yellow for lowspin carbon and grey for highspin, light blue for iron and dark blue for nitrogen.

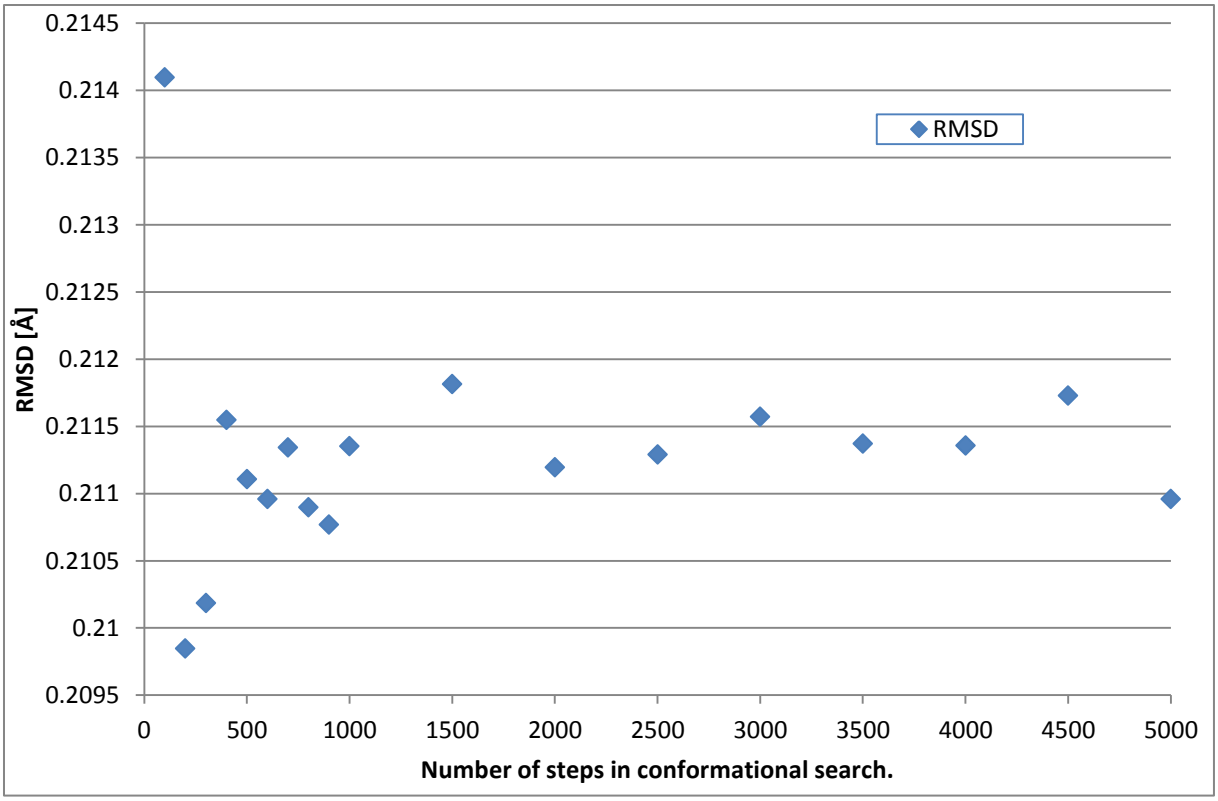


**Figure 4.26:** A plot between conformational search limit and RMSD between the lowpin and highspin molecules geometry. This is for the molecule with CCSD refcode PURYIK. The RMSD is based on using the SMARTS [Fe](N)(N)(N)(N)(N)(N) in openbabel.

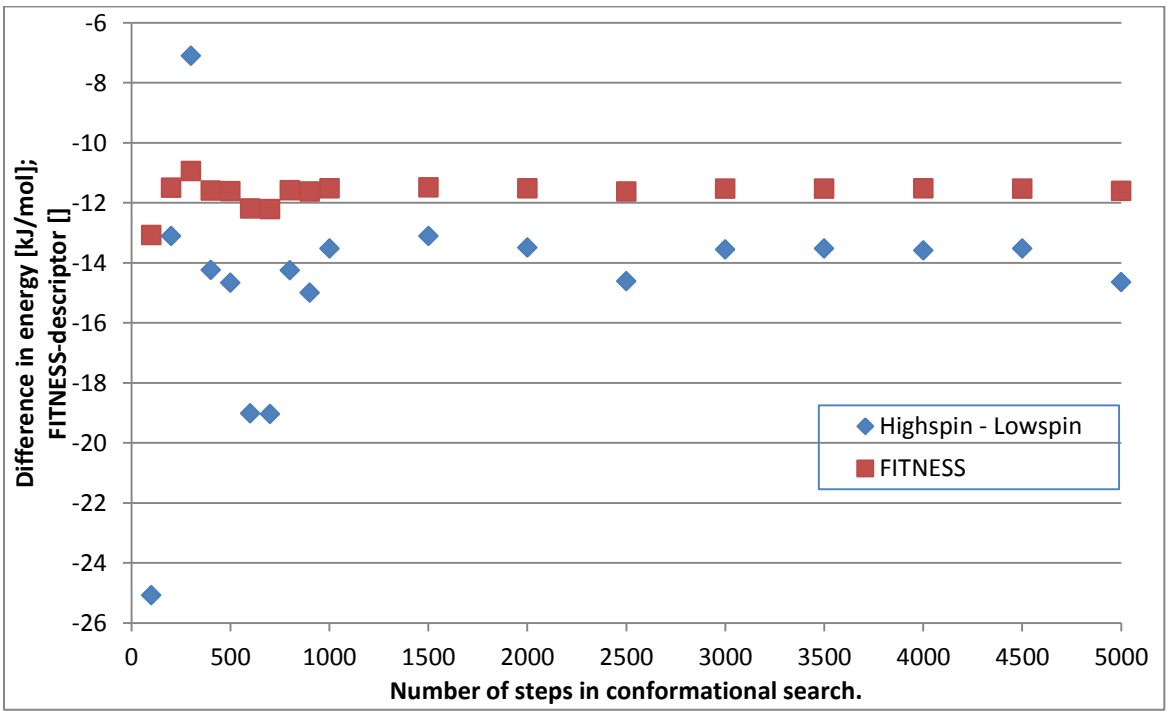


**Figure 4.27:** A plot including both the difference in potential energy $U_{HS} - U_{LS}$ against conformational search limit and the fitness against conformational search limit. This is for the molecule with CCSD refcode PURYIK.

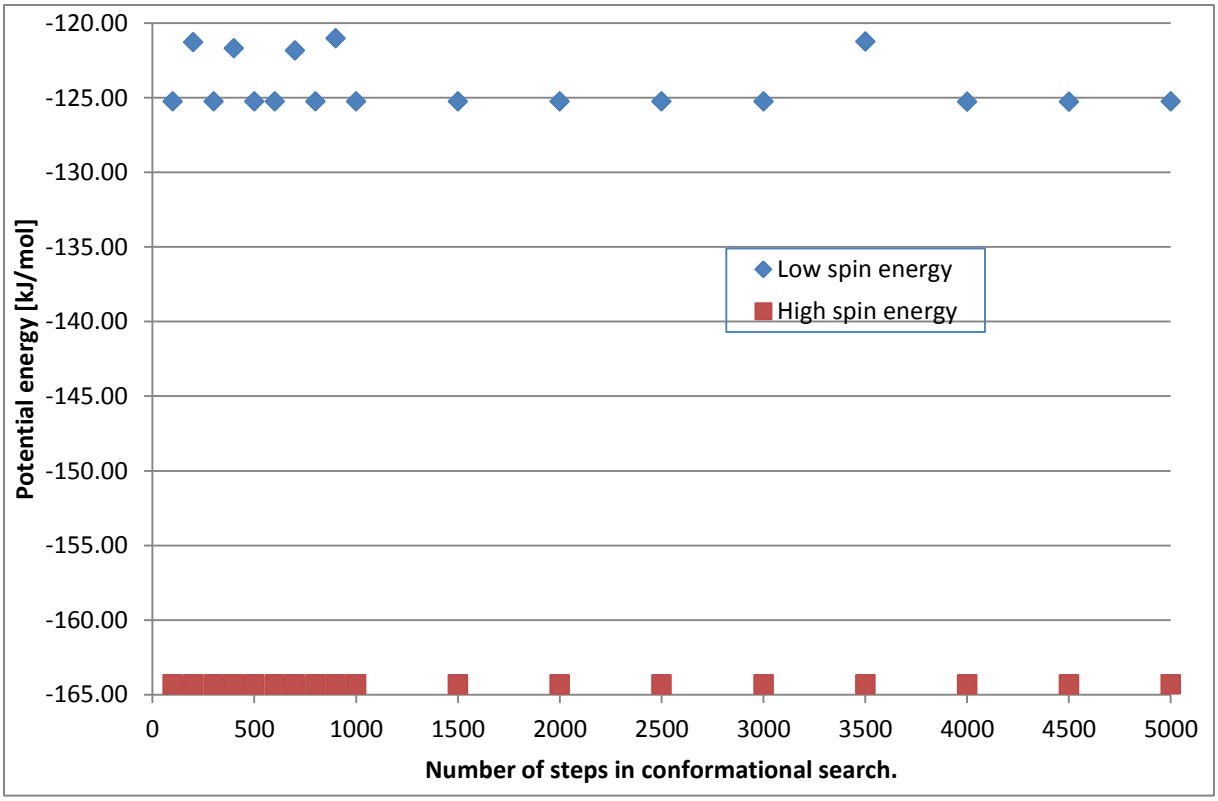


**Figure 4.28:** A plot between conformational search limit and potential energy for both highspin and lowspin states. This is for the molecule with CCSD refcode PAZXAP.

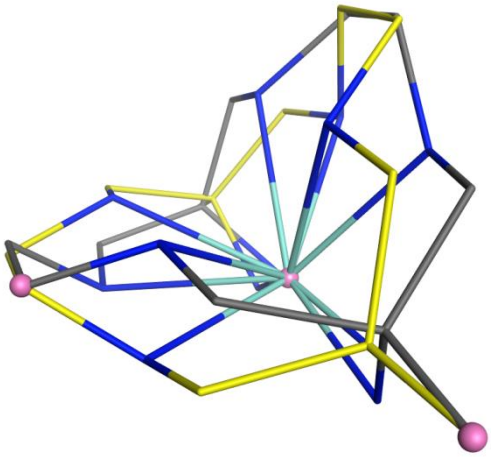

**Figure 4.29:** The superposition of two highspin molecules with CCSD refcode PAZXAP. Conformational search 500_2_old shown with yellow carbons against 500_2_new shown with grey carbons. RMSD 0.03029 Å by superposition the 3 atoms highlighted in pink.

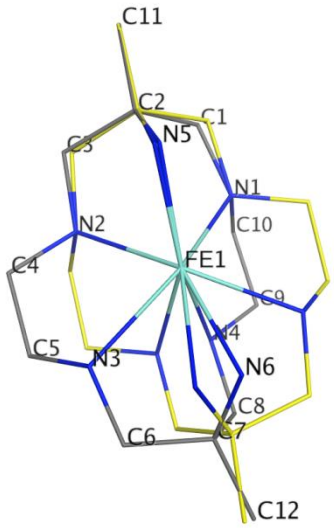


**Figure 4.30:** Two superposed molecules based on different conformational searches in the highspin state. Molecule has CCSD refcode PAZXAP. The colour-coding is light blue for iron, dark blue for nitrogen, yellow for carbon in the 100 step conformational search and grey for carbon in the 1000 step conformational search. Hydrogens is intentially hidden to easier reveal the structure. The superposition was created by manually combining the atoms marked as FE1, N1, N2 and C11 with a resulting RMSD of 0.00254 Å.

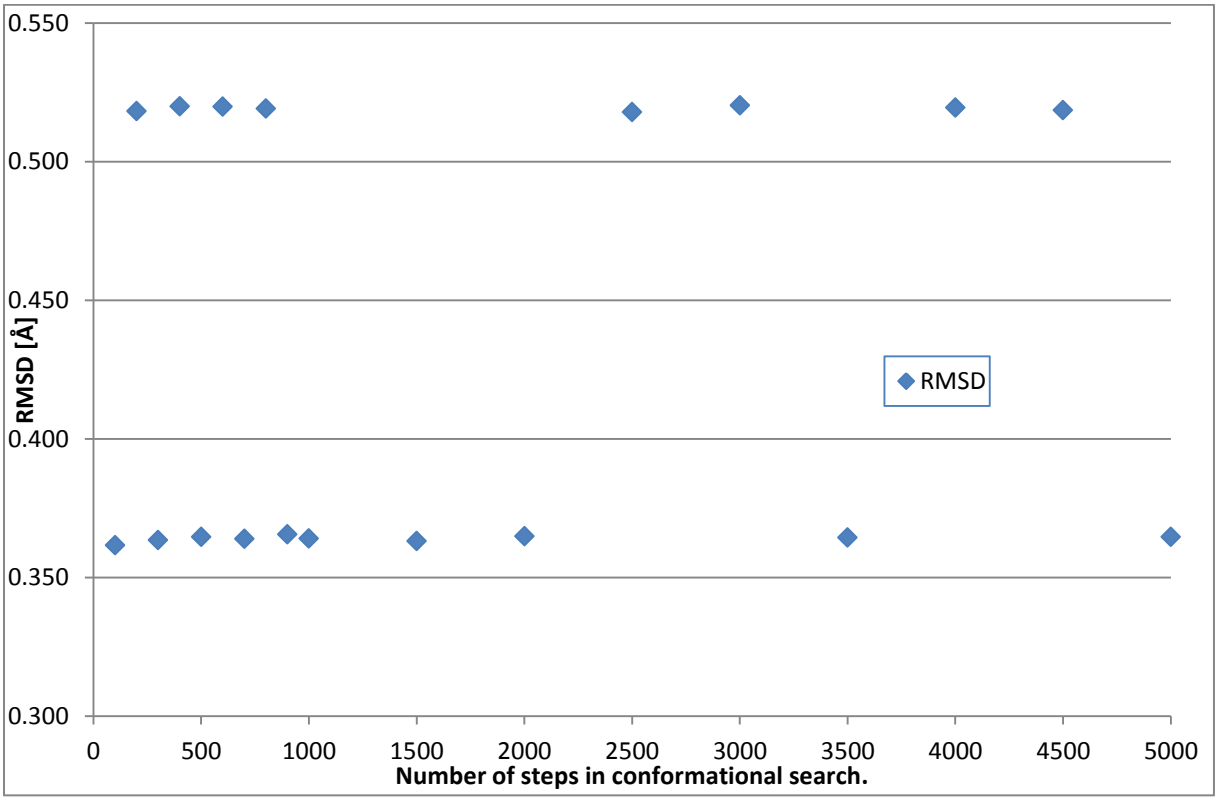


**Figure 4.31:** A plot between conformational search limit and RMSD between the lowpin and highspin molecules geometry. This is for the molecule with CCSD refcode LOTSES. The RMSD is based on using the SMARTS [Fe](N)(N)(N)(N)(N)(N) in openbabel.

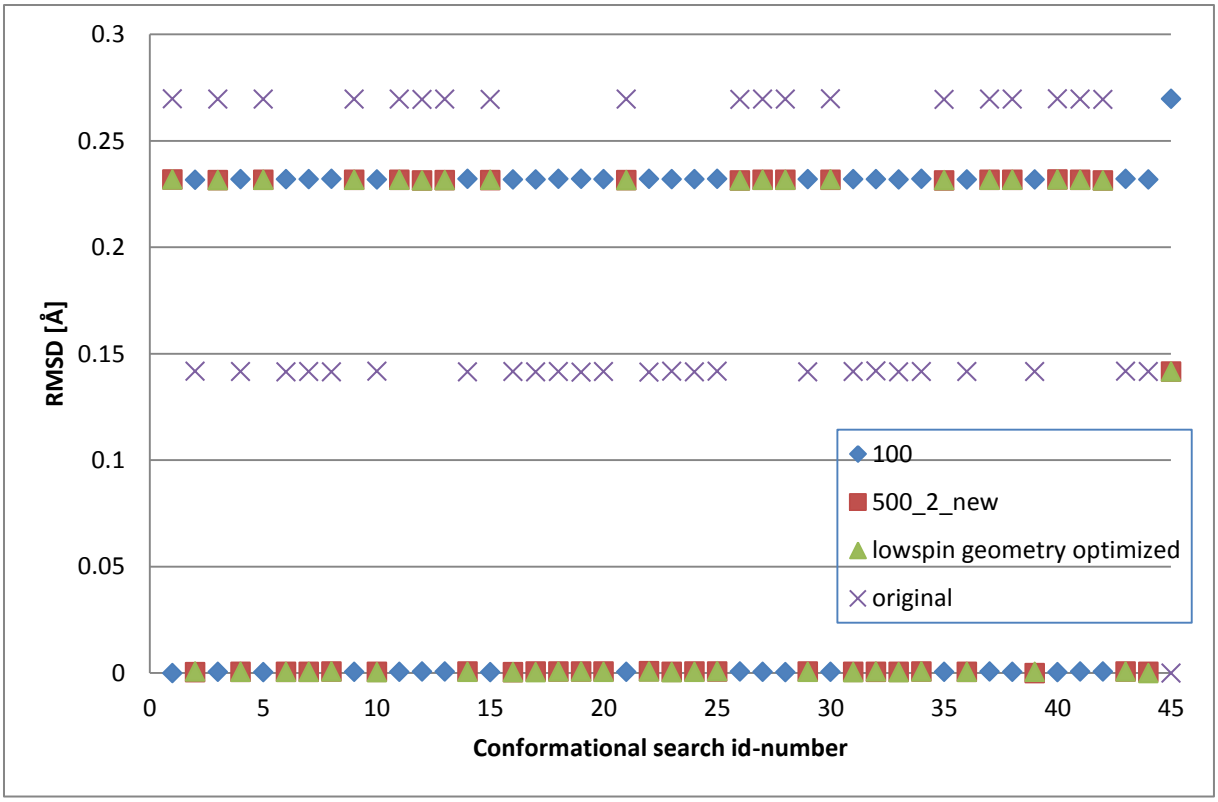


**Figure 4.32:** A plot between conformational search id-number and RMSD. This is for the molecule with CCSD refcode LOTSES and is for the lowspin case. The order of conformational search id is 1-18 is single conformational 100-500 while 19-43 is same order as in table 4.5. Id 44 is the geometry-optimized while 45 is original.

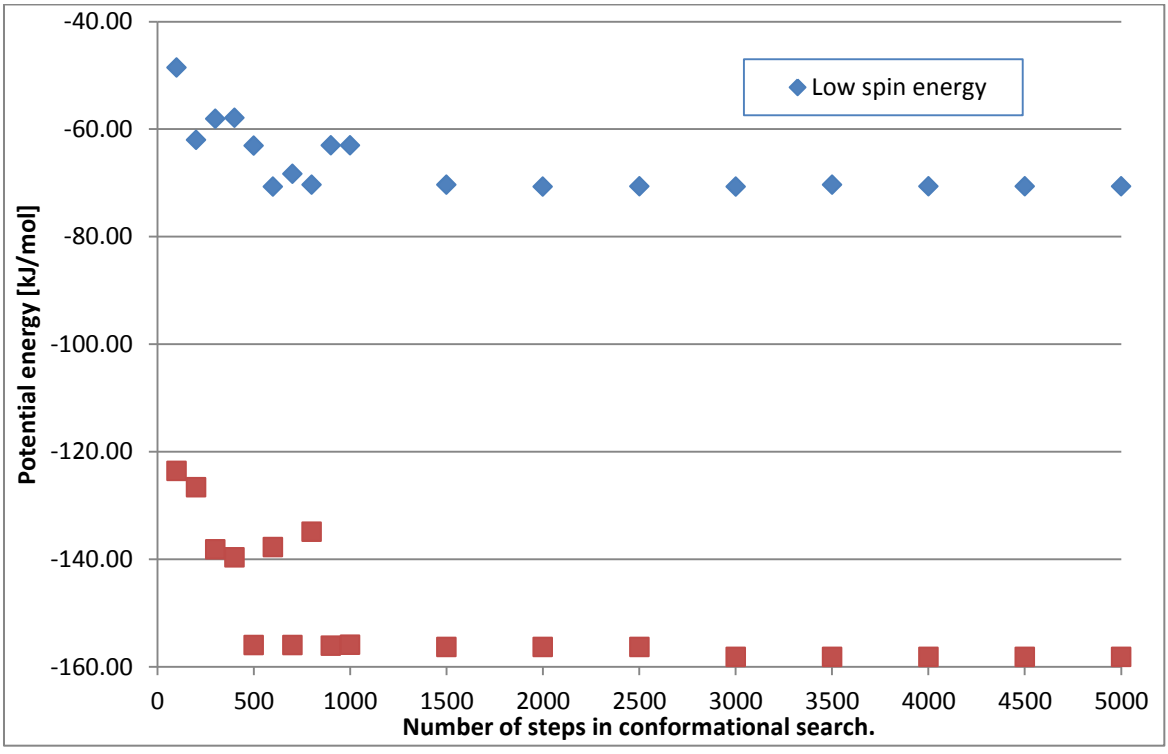


**Figure 4.33:** A plot between conformational search limit and potential energy for both highspin and lowspin states. This is for the molecule with CCSD refcode KIBWEZ.

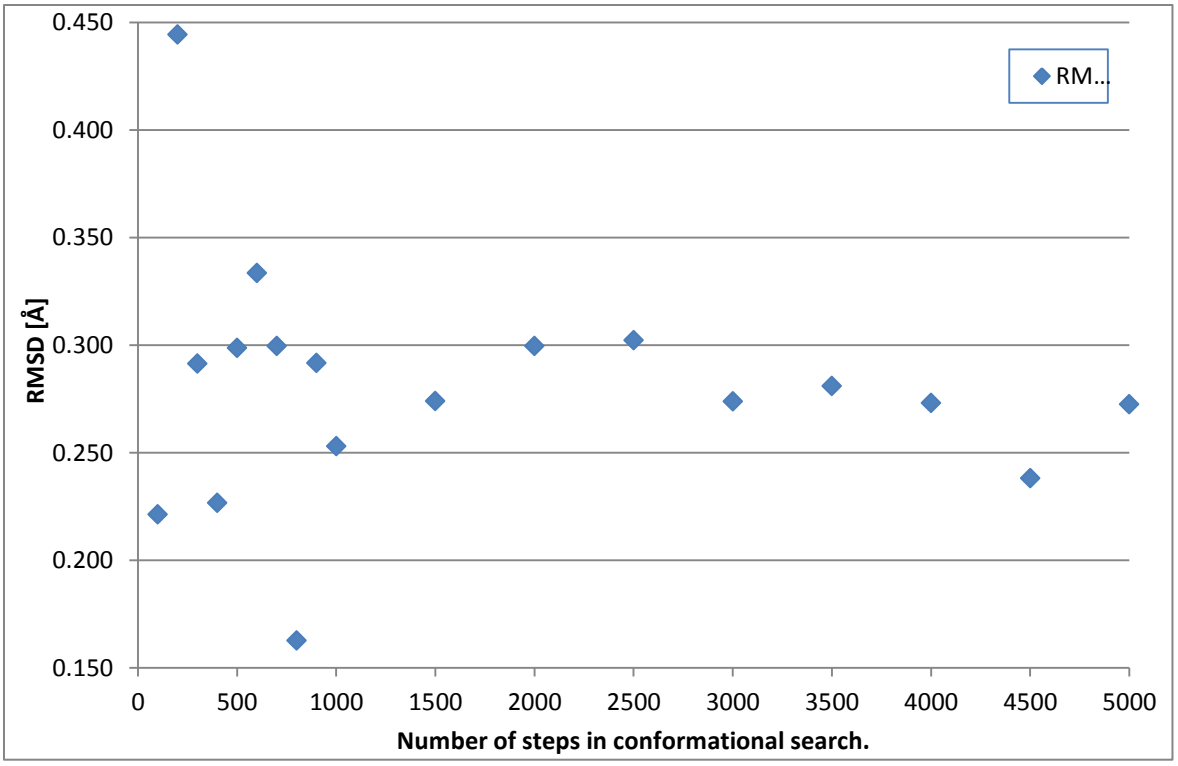


**Figure 4.34:** A plot between conformational search limit and RMSD between the lowpin and highspin molecules geometry. This is for the molecule with CCSD refcode KIBWEZ. The RMSD is based on using the SMARTS [Fe](N)(N)(N)(N)(N)(N) in openbabel.

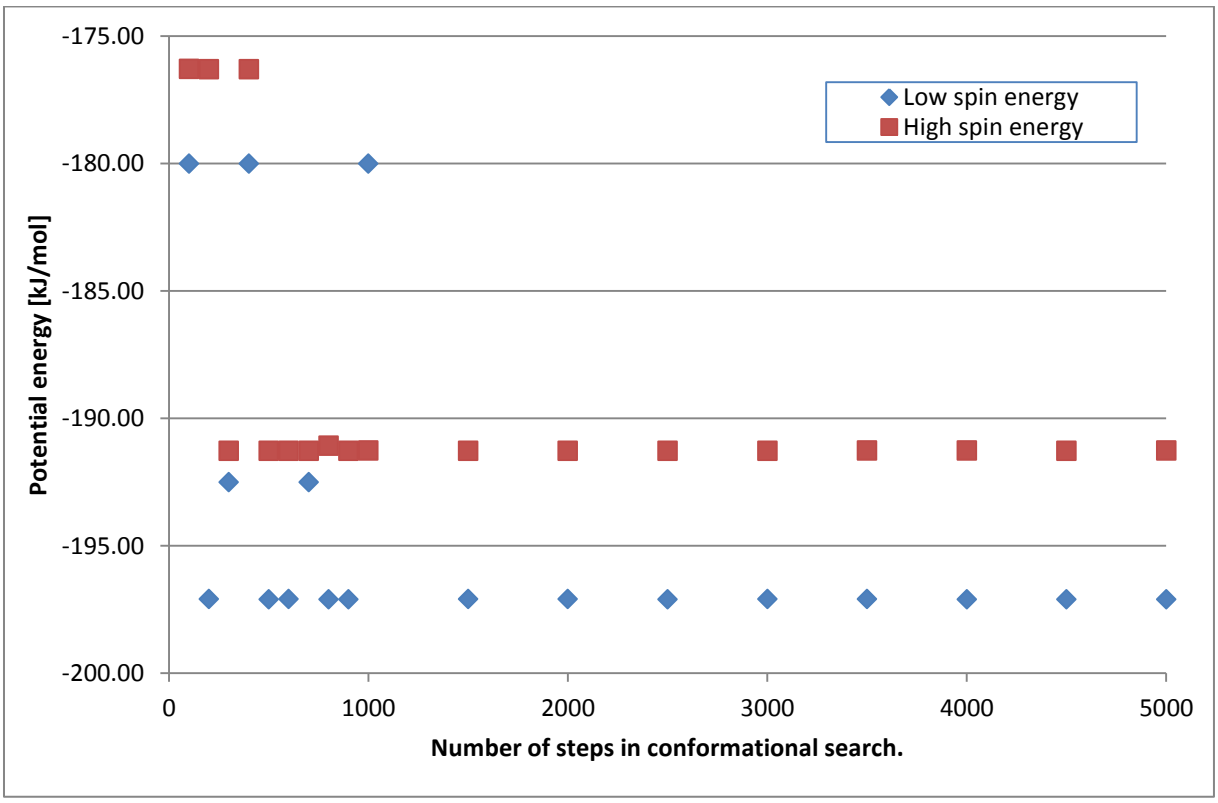


**Figure 4.35:** A plot between conformational search limit and potential energy for both highspin and lowspin states. This is for the molecule with CCSD refcode DETTOL.

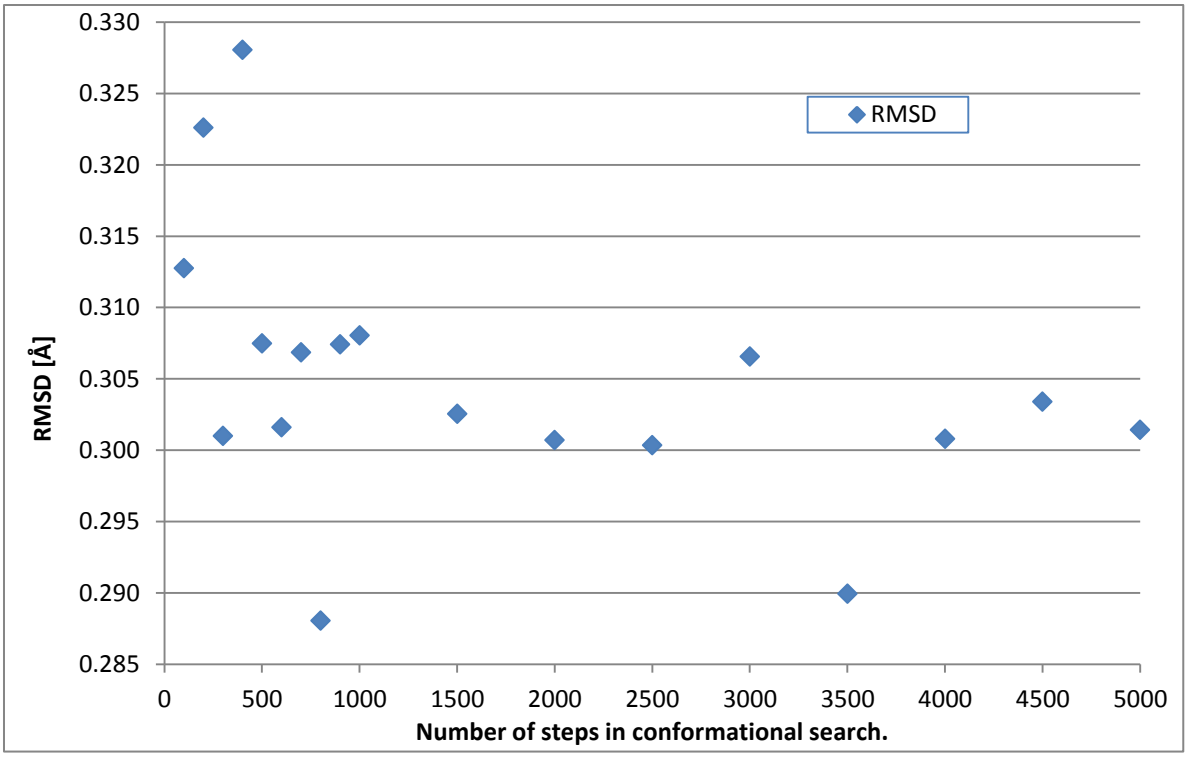


**Figure 4.36:** A plot between conformational search limit and RMSD between the lowpin and highspin molecules geometry. This is for the molecule with CCSD refcode DETTOL. The RMSD is based on using the SMARTS [Fe](N)(N)(N)(N)(N)(N) in openbabel.

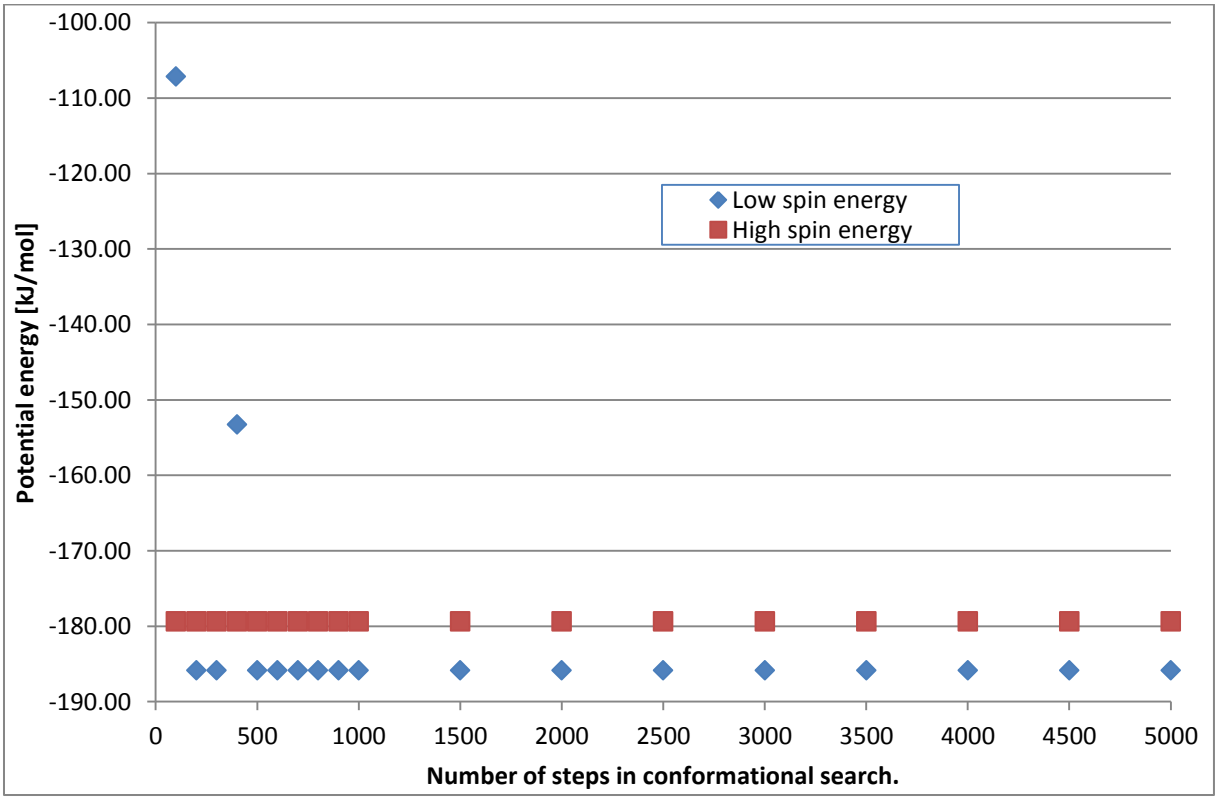


**Figure 4.37:** A plot between conformational search limit and potential energy for both highspin and lowspin states. This is for the molecule with CCSD refcode HUMBEX.

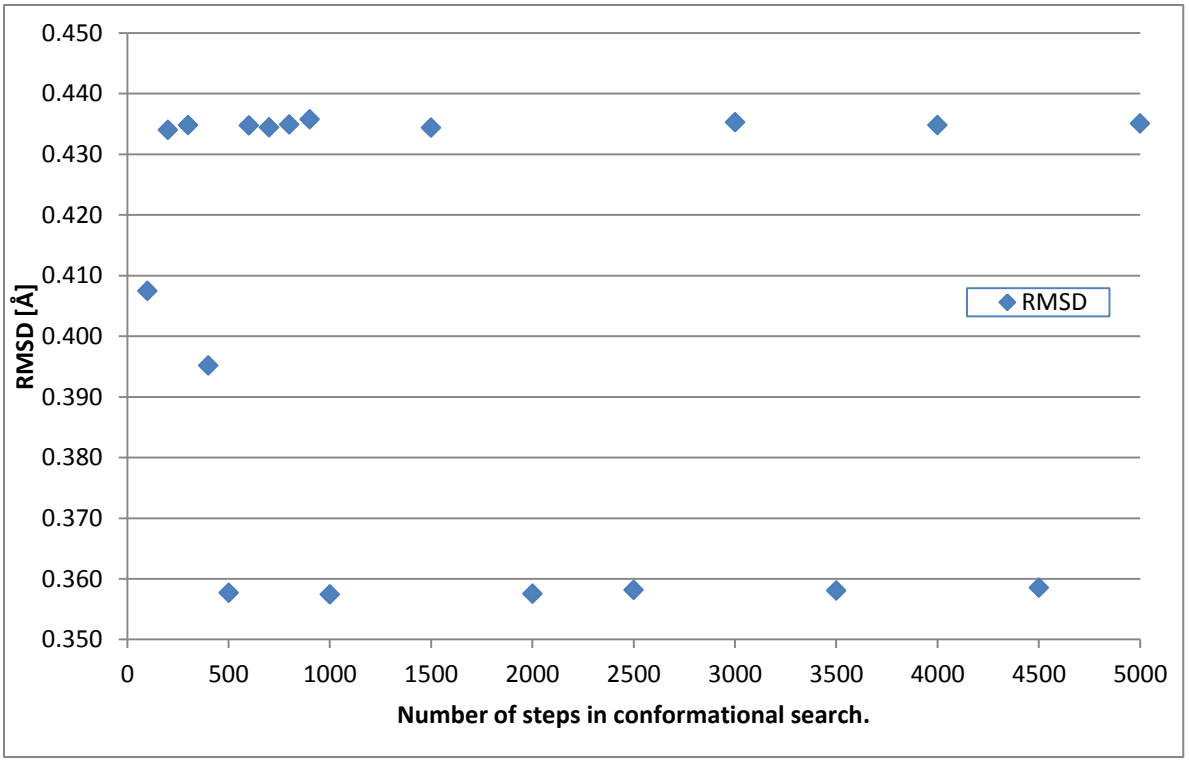


**Figure 4.38:** A plot between conformational search limit and RMSD between the lowpin and highspin molecules geometry. This is for the molecule with CCSD refcode HUMBEX. The RMSD is based on using the SMARTS [Fe](N)(N)(N)(N)(N)(N) in openbabel.

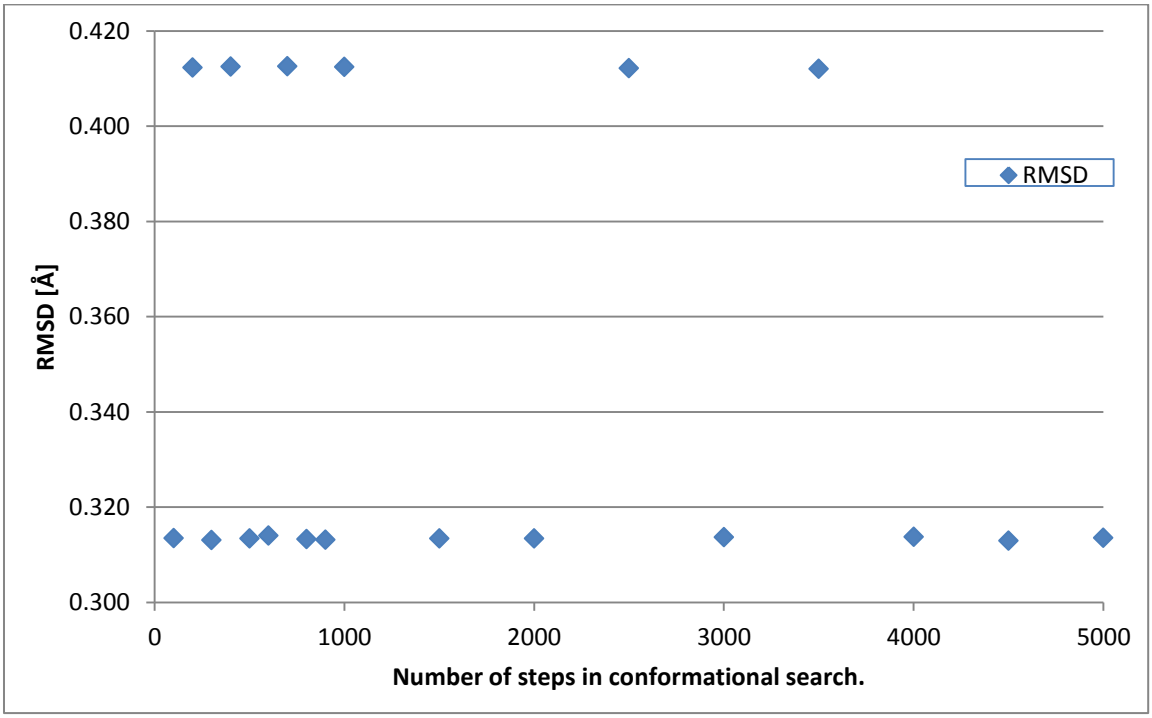


**Figure 4.39:** A plot between conformational search limit and RMSD between the lowpin and highspin molecules geometry. This is for the molecule with CCSD refcode FEBMAB. The RMSD is based on using the SMARTS [Fe](N)(N)(N)(N)(N)(N) in openbabel.

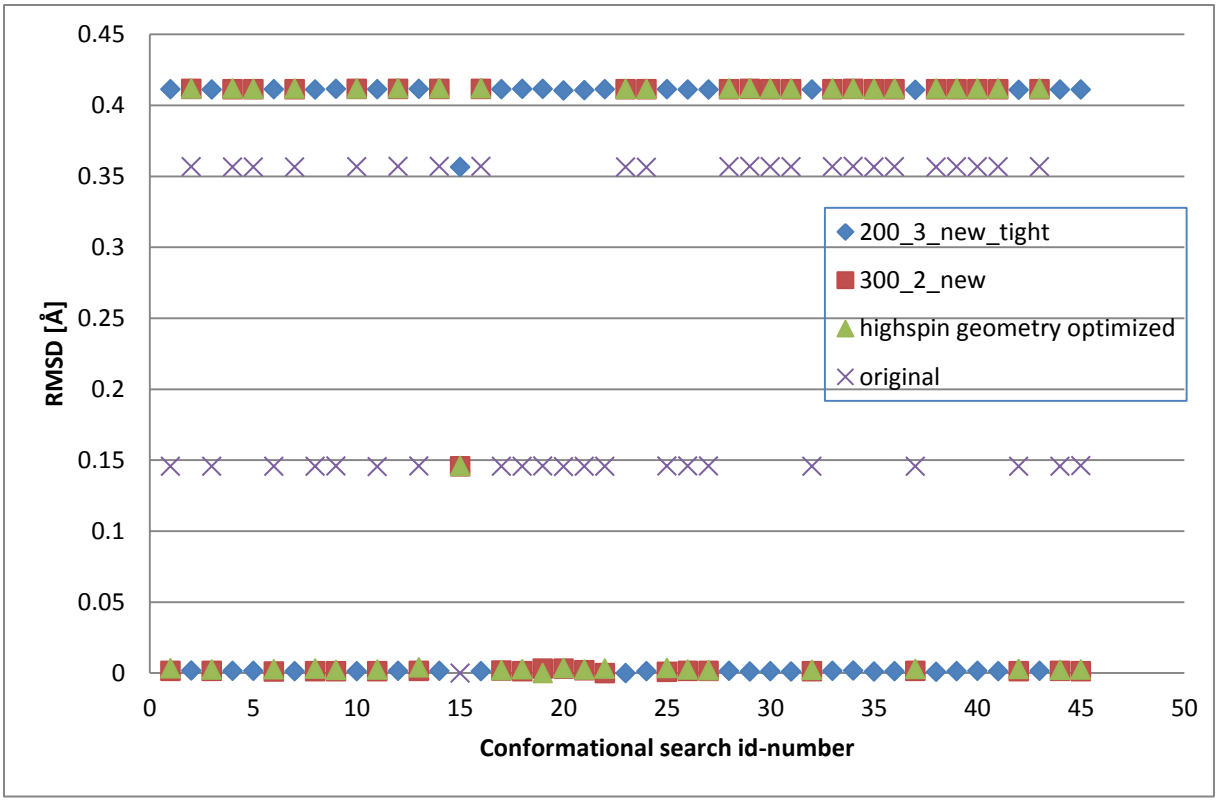


**Figure 4.40:** A plot between conformational search id number and RMSD for highspin geometry of molecule with CCSD refcode FEBMAB. The search id is 1-18 for single conformational search 100 – 5000, 19-43 for multi in same order as in table 4.5, 44 is geometry optimized and 45 is original.

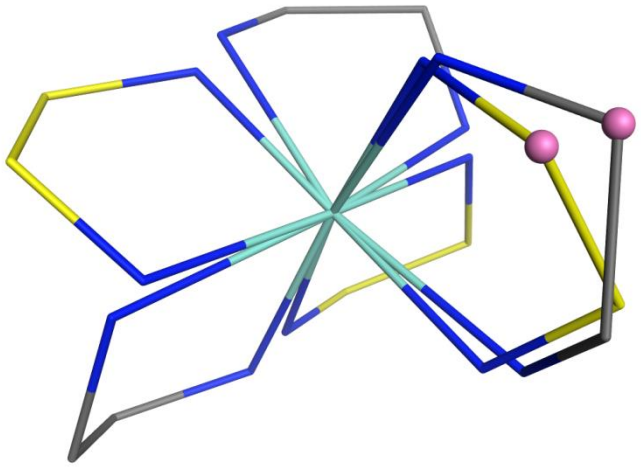

**Figure 4.41:** A superposition between lowspin with yellow carbons and highspin with grey carbons for a 1000-step conformational search on molecule with CCSD refcode FEBMAB. The RMSD is 0.4124 Å while the marked distance between the two pink atoms is 0.621 Å.

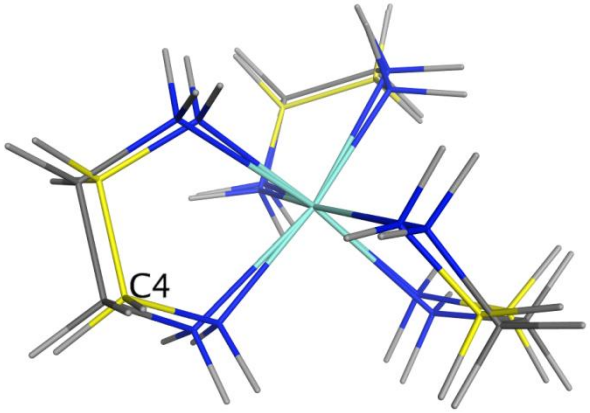


**Figure 4.42:** A superposition between lowspin marked with yellow carbons and highspin marked with grey carbons for a 100-step conformational search on molecule with CCSD refcode FEBMAB. The RMSD is 0.3135 Å. The distance between Fe and C4 is 2.923 Å for lowspin and 3.178 Å for highspin.

### 4.3.1 Results from molecular dynamics.

There are not many results from MD, just multiple lowspinstate runs from the same long conformational search (100 upto 6400). Bond-constaint was light. See table 4.6, figure 4.43, 4.44 and 4.45 for the results.

**Table 4.6:** The table shows how long a lowspin molecular dynamics search lasted before crashing or reaching the end at 300 ps with timestep length as indicated. The CCSD refcodes is as indicated in the table.

| Timestep [fs] | KIBWAV | HUMBEX | KIBWEZ | PURYIK | DETTOL | PAZXAP | LOTSES | FEBMAB | TIYPIC | IPIWAF |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.1 | 36.976 | 16.263 | 300 | 1.176 | 1.842 | 0.152 | 0.616 | 1.584 | 0.234 | 0.372 |
| 0.2 | 74.215 | 11.962 | 16.946 | 5.434 | 0.451 | 0.125 | 1.056 | 0.331 | 0.246 | 0.361 |
| 0.3 | 2.3232 | 38.304 | 176.9412 | 4.2156 | 0.3516 | 0.2724 | 1.1148 | 1.6392 | 0.084 | 0.6876 |
| 0.4 | 19.6536 | 7.38 | 300 | 1.5732 | 0.234 | 0.0972 | 0.2244 | 0.2832 | 0.2664 | 0.3504 |
| 0.5 | 11.854 | 20.231 | 300 | 2.329 | 1.035 | 0.347 | 0.439 | 0.264 | 0.117 | 0.252 |
| 0.6 | 37.5708 | 10.6236 | 300 | 0.9168 | 0.6696 | 0.1248 | 2.2608 | 0.2064 | 1.26 | 1.9068 |
| 0.7 | 1.6688 | 13.272 | 300 | 24.6148 | 1.0458 | 0.1358 | 0.0952 | 0.2534 | 0.3486 | 0.448 |
| 0.8 | 2.3152 | 54.1616 | 138.1488 | 7.3296 | 0.3408 | 0.0848 | 0.8336 | 0.2576 | 0.2672 | 0.2208 |
| 0.9 | 22.1508 | 9.9846 | 300 | 2.259 | 0.0972 | 0.1116 | 0.4608 | 0.2754 | 0.3492 | 0.2916 |
| 1 | 13.811 | 10.682 | 300 | 2.218 | 0.386 | 0.213 | 1.526 | 0.364 | 0.081 | 0.275 |
| 1.1 | 6.1985 | 11.8976 | 21.5325 | 36.7136 | 0.2211 | 0.2178 | 0.4576 | 0.3751 | 0.2233 | 0.1958 |
| 1.2 | 3.5184 | 4.7148 | 300 | 5.4744 | 1.5768 | 0.1008 | 0.2928 | 0.2184 | 0.3468 | 0.2172 |
| 1.3 | 2.2295 | 10.5339 | 4.9231 | 1.6705 | 0.3211 | 1.0387 | 0.3601 | 0.3211 | 0.2782 | 0.1651 |
| 1.4 | 24.1178 | 18.977 | 32.438 | 61.2178 | 0.4186 | 0.0882 | 0.3122 | 0.3332 | 0.3416 | 1.5918 |
| 1.5 | 91.2345 | 4.902 | 6.858 | 1.074 | 5.4945 | 0.1215 | 0.3075 | 0.246 | 0.2685 | 0.264 |
| 1.6 | 6.1776 | 10.6528 | 300 | 28.3904 | 2.3184 | 0.152 | 0.328 | 2.8832 | 0.3968 | 0.3472 |
| 1.7 | 11.9425 | 1.6082 | 114.8979 | 19.3953 | 2.9138 | 0.1615 | 0.2941 | 0.3604 | 0.5389 | 0.0816 |
| 1.8 | 30.402 | 5.5062 | 9.5094 | 0.2592 | 0.3528 | 0.2916 | 0.4428 | 1.692 | 0.2664 | 0.2844 |
| 1.9 | 9.9275 | 23.2256 | 2.9222 | 9.9674 | 2.3142 | 0.1976 | 0.9329 | 0.5548 | 0.4161 | 0.1919 |
| 2 | 7.418 | 2.844 | 300 | 23.994 | 0.286 | 0.328 | 4.176 | 0.356 | 1.19 | 0.298 |

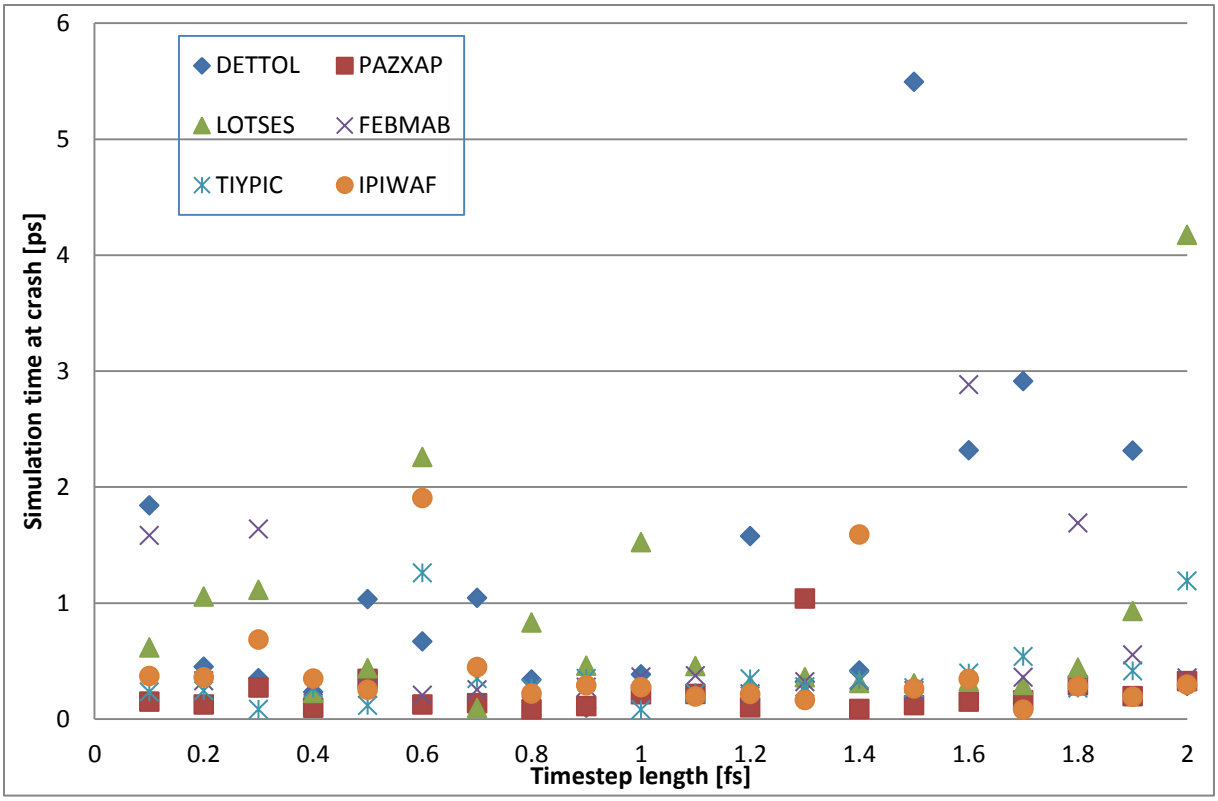


**Figure 4.43:** Molecular dynamics in lowspin state showing the crashes for the molecules with CCSD refcodes as indicated in the graph.

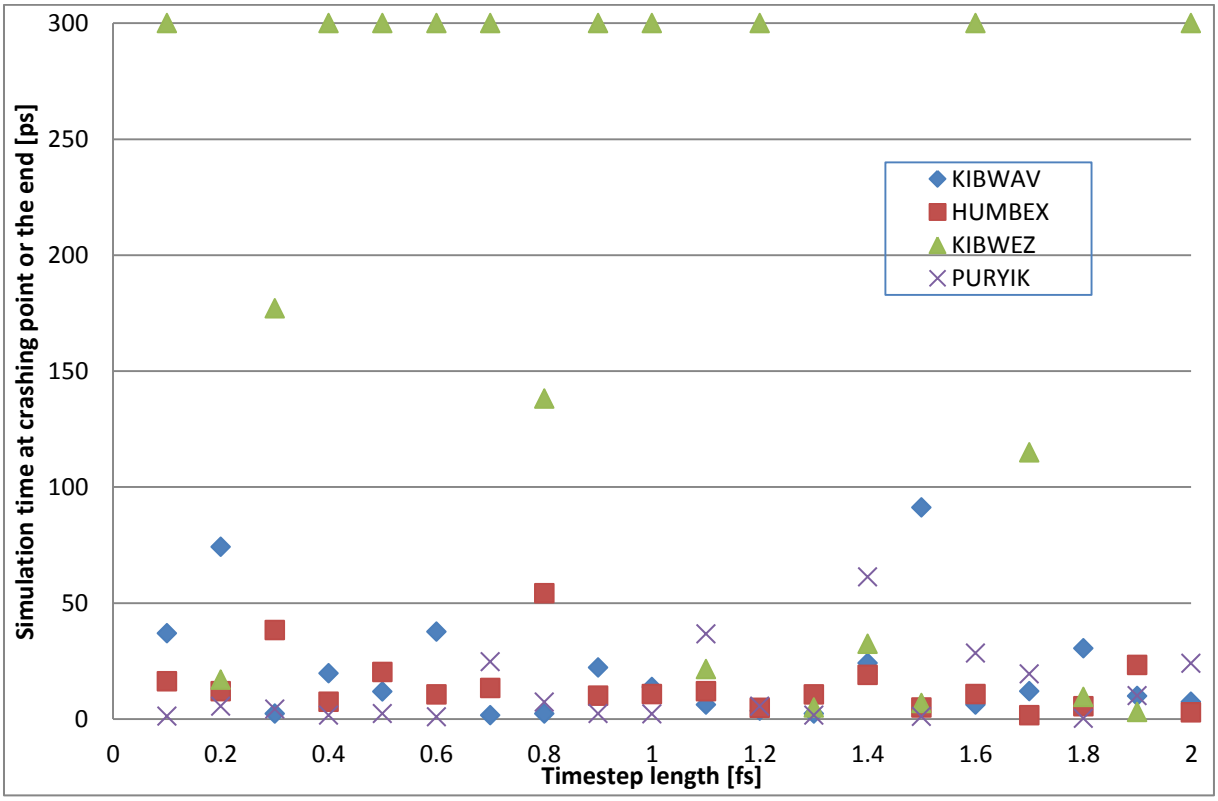


**Figure 4.44:** Molecular dynamics in lowspin state showing the crashes or the ocassional successes in a 300 ps MD. The CCSD refcodes are as indicated in the graph.

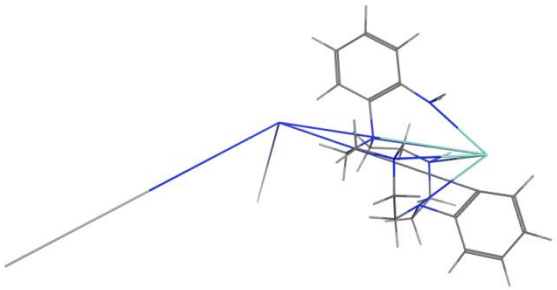

**Figure 4.45:** The figure shows a badly distorted molecule in MD simulation. The molecule has CCSD refcode IPIWAF.

### 4.4: Results from Denoptim runs.

With over 16000 created molecules in the different runs any detailed description is not an option. In the following tables and graphs only a few main features is shown, with a total overview of the different runs in table 4.7. Some more graphs is located in appendix 9.4.

**Table 4.7:** An overview of Denoptim runs overall number of generations, count of built good molecules and errors.

| # | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Run id | run_160 | run_201 | run_310 | run_323 | run_325 |
| Initial population | 100 | 100 | 100 | 100 | 100 |
| Children per generation | 50 | 25 | 25 | 25 | 25 |
| Number of generations | 100 | 70 | 130 | 130 | 130 |
| Number of good molecules | 5108 | 1871 | 3359 | 3359 | 3357 |
| Number of duplicates | 4837 | 2089 | 4790 | 2840 | 9423 |
| Number of invalid molecules | 81 | 35 | 61 | 87 | 253 |
| Too short distances between atoms | 474 | 426 | 1058 | 489 | 3062 |
| Errored due to programming bugs | | 17 | | | |
| Number of molecules rejected for other reasons | 14 | 30 | 34 | 19 | 10 |

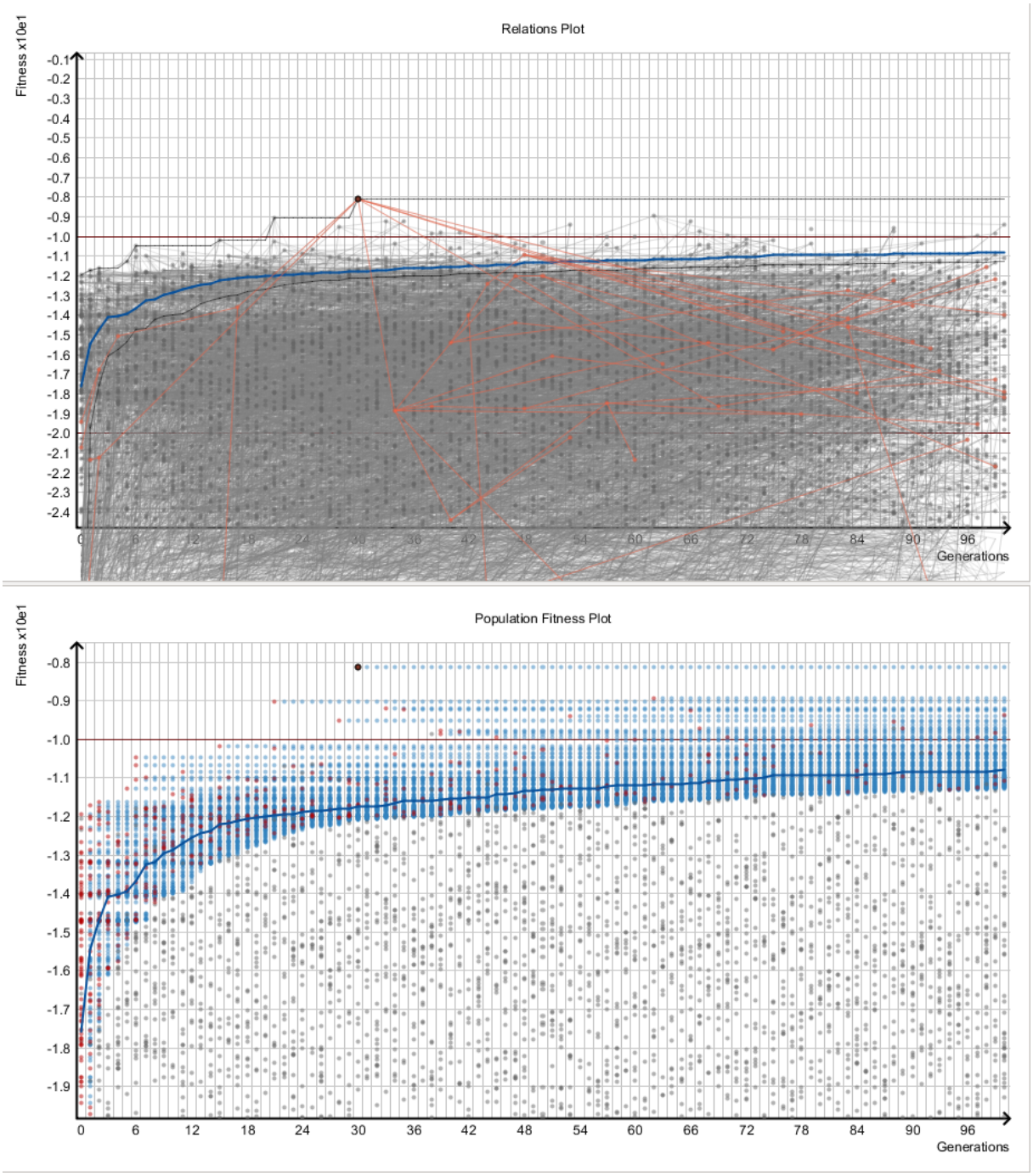


**Figure 4.46:** An overview of the population in Denoptim run internally known as run_160. The plot shows the fittest molecule was made by EA. Since run_160 is "keep growing" the full population is eligible for creating children. The lowest part of the population is cut off in this plot.

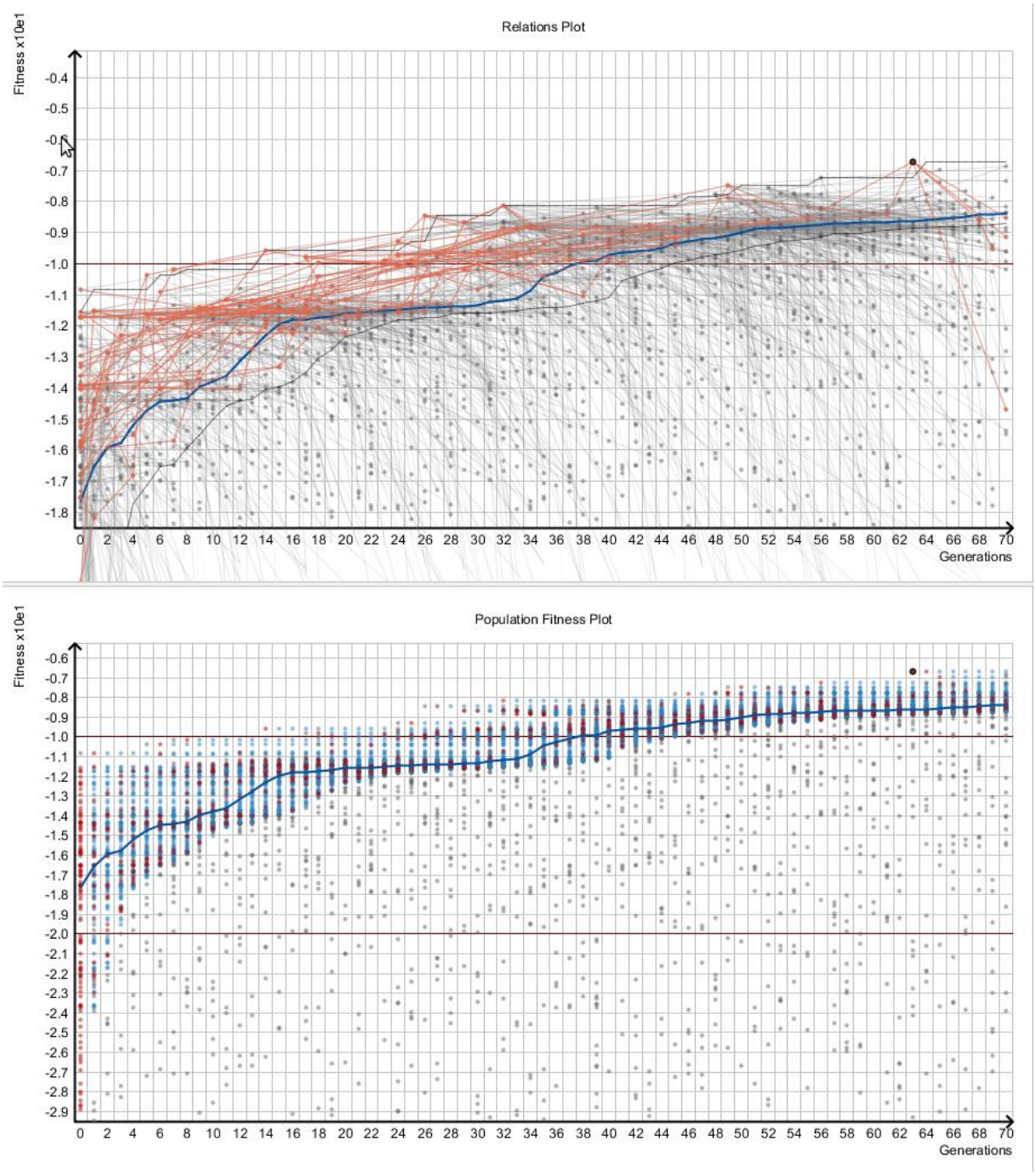


**Figure 4.47:** An overview of the population in Denoptim run internally known as run_201. The plot shows the fittest molecule was made by EA.Since run_201 is "survival of the fittest" only molecules among the blue and orange is eligible for creating children. The lowest part is cut off in this plot.

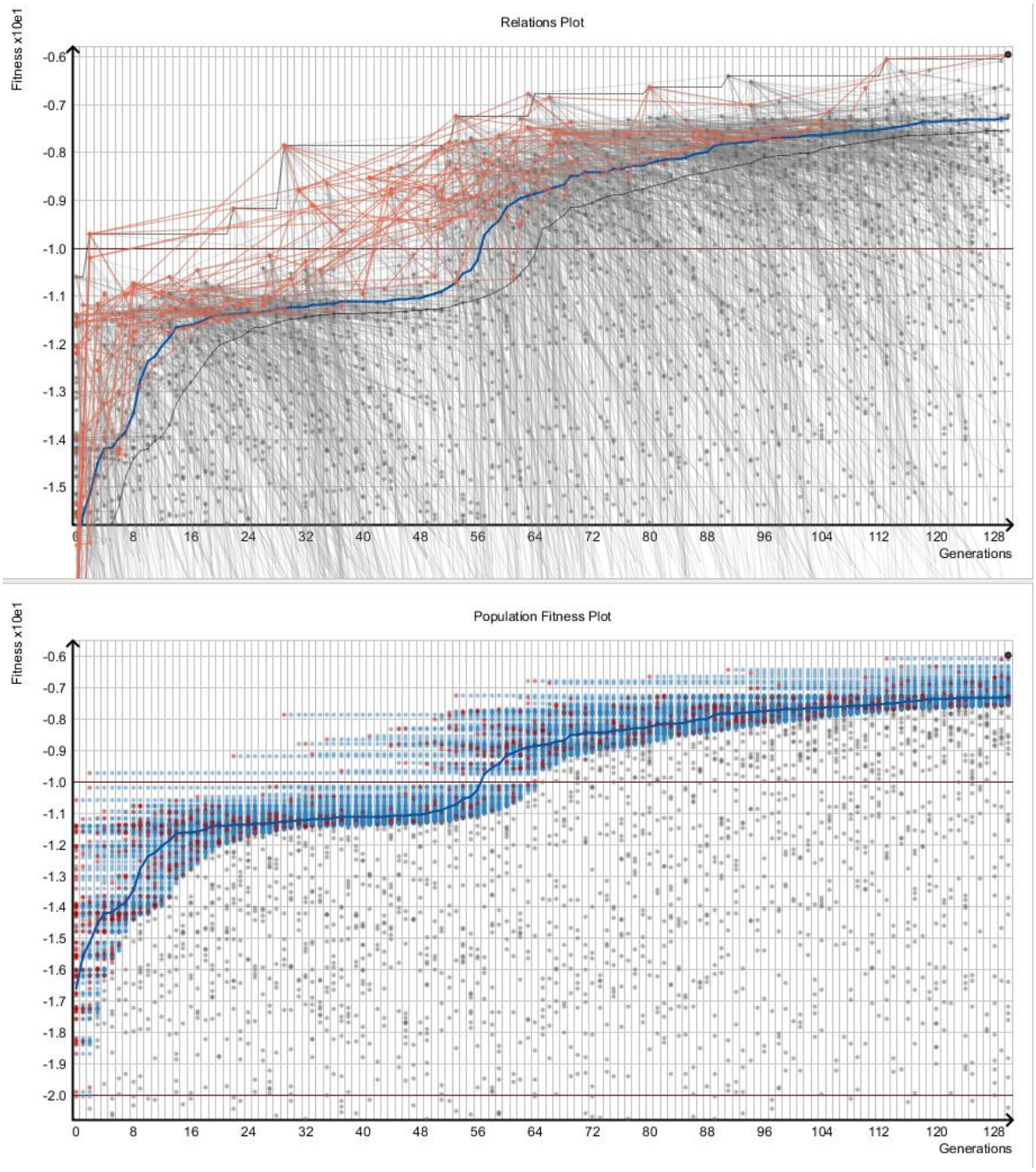


**Figure 4.48:** An overview of the population in Denoptim run internally known as run_310.The plot shows the fittest molecule was made by EA. Since run_310 is "survival of the fittest" only molecules among the blue and orange is eligible for creating children. The lowest part is cut off in this plot.

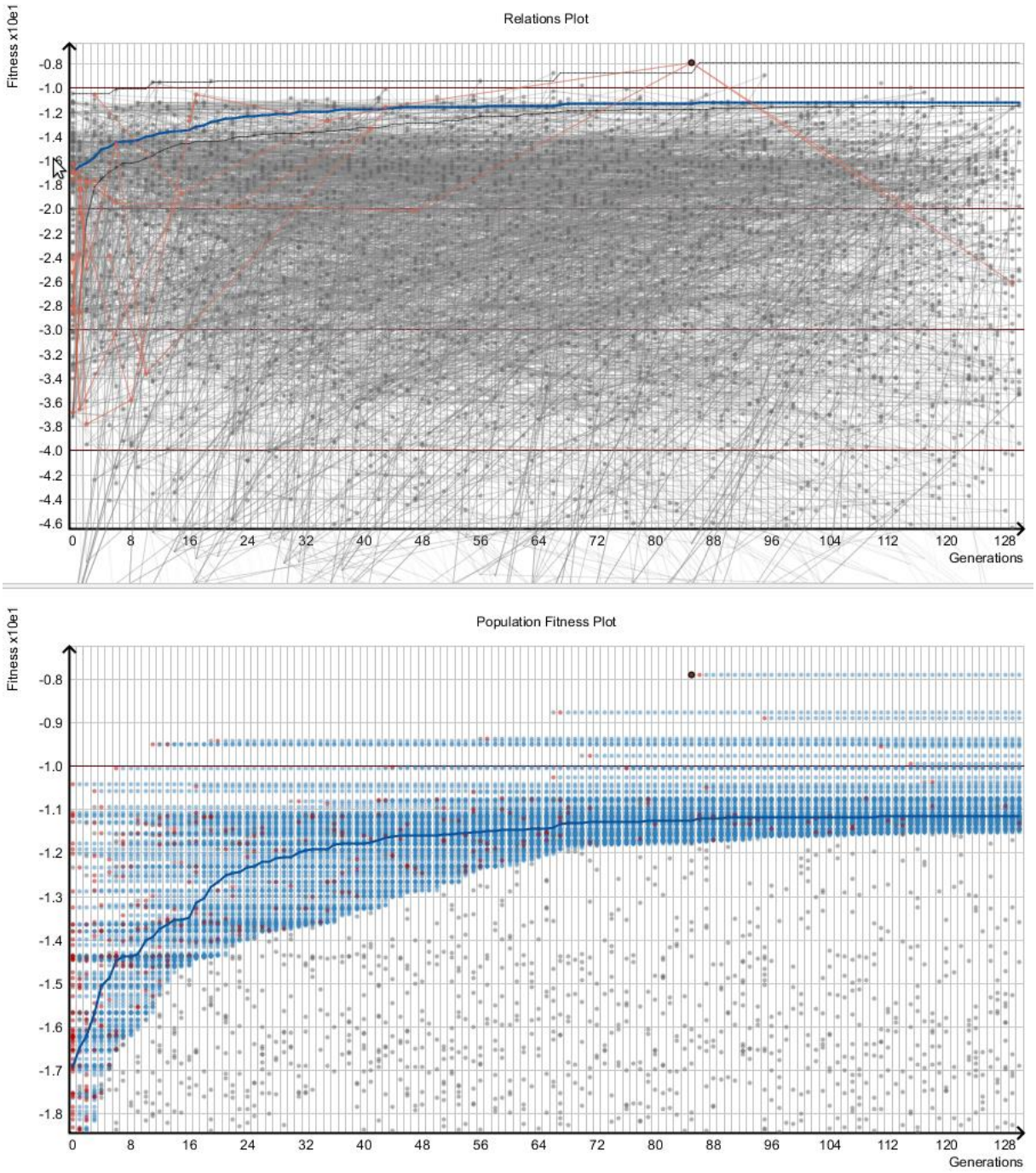


**Figure 4.49:** An overview of the population in Denoptim run internally known as run_323. The plot shows the fittest was made by EA. Since run_323 is "keep growing" the full population is eligible for creating children. The lowest part is cut off in this plot.

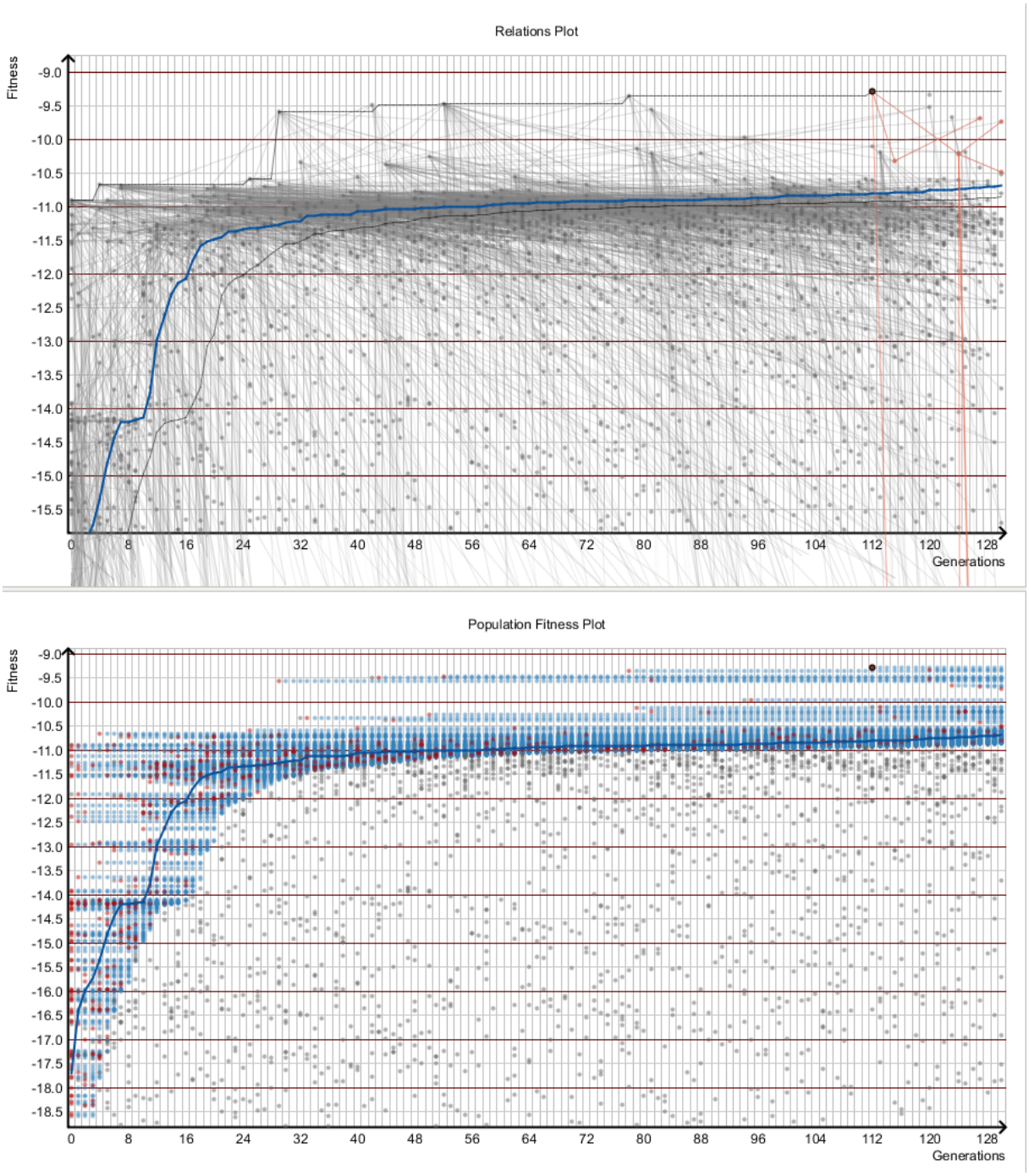


**Figure 4.50:** An overview of the population in Denoptim run internally known as run_325. The plot shows the fittest was not made by EA but instead from original fragment of libraries. Since run_325 is "survival of the fittest" only molecules among the blue and orange is eligible for creating children. The lowest part is cut off in this plot.

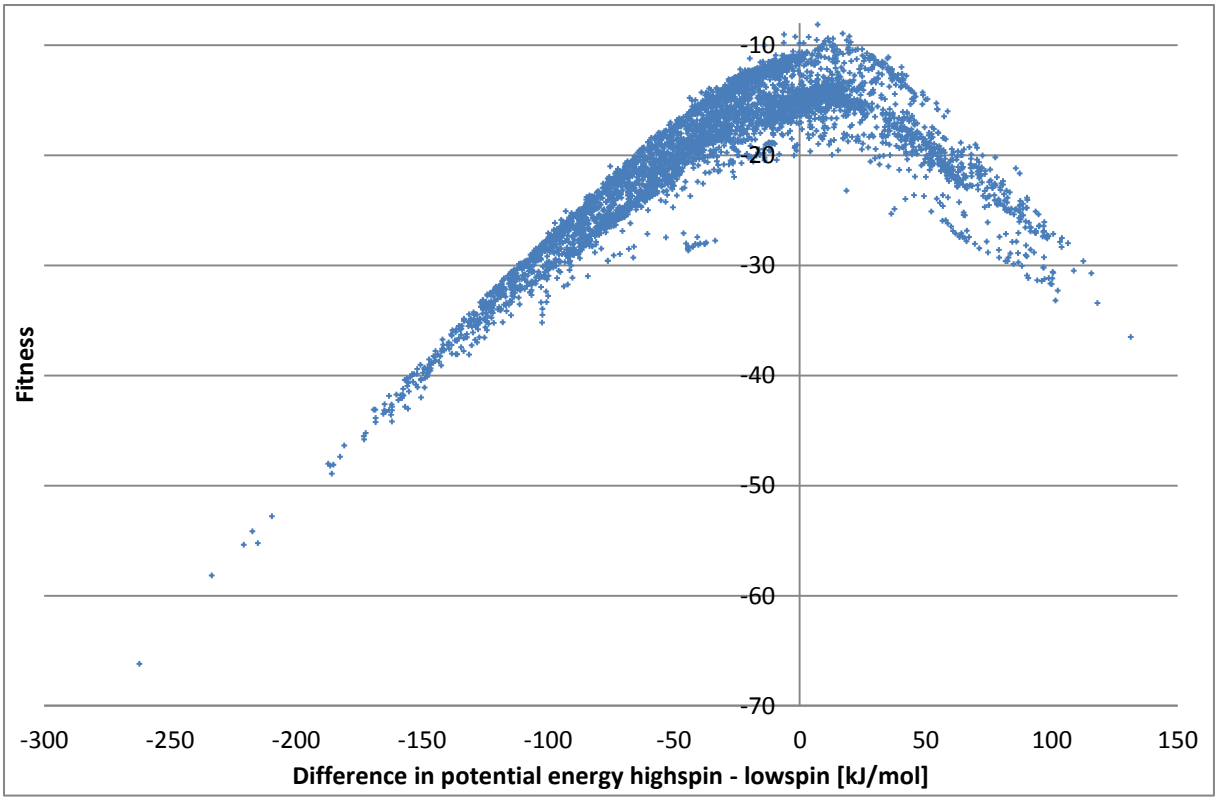


**Figure 4.51:** Plot showing the difference in potential energy by calculating $U_{HS} - U_{LS}$ against fitness in Denoptim run with internal id run_160. This plot includes all the 5108 valid molecules.

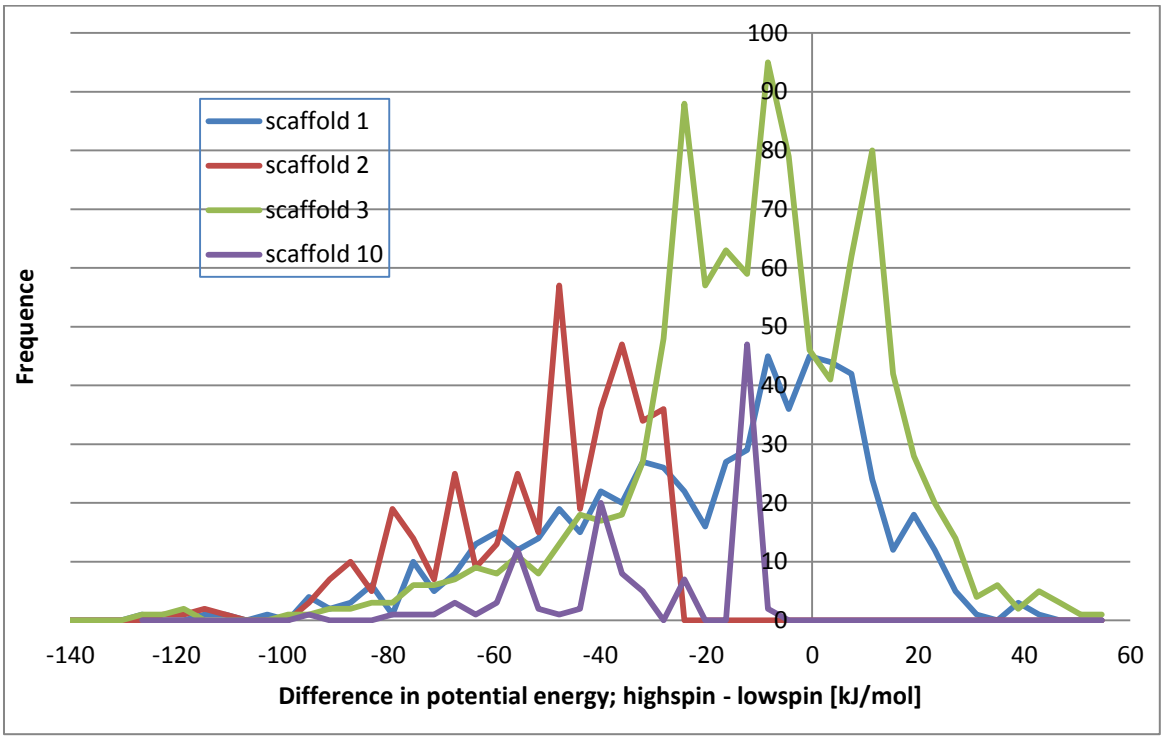


**Figure 4.52:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_160. The histogram includes all good molecules built on the basis of scaffolds 1, 2, 3 and 10.

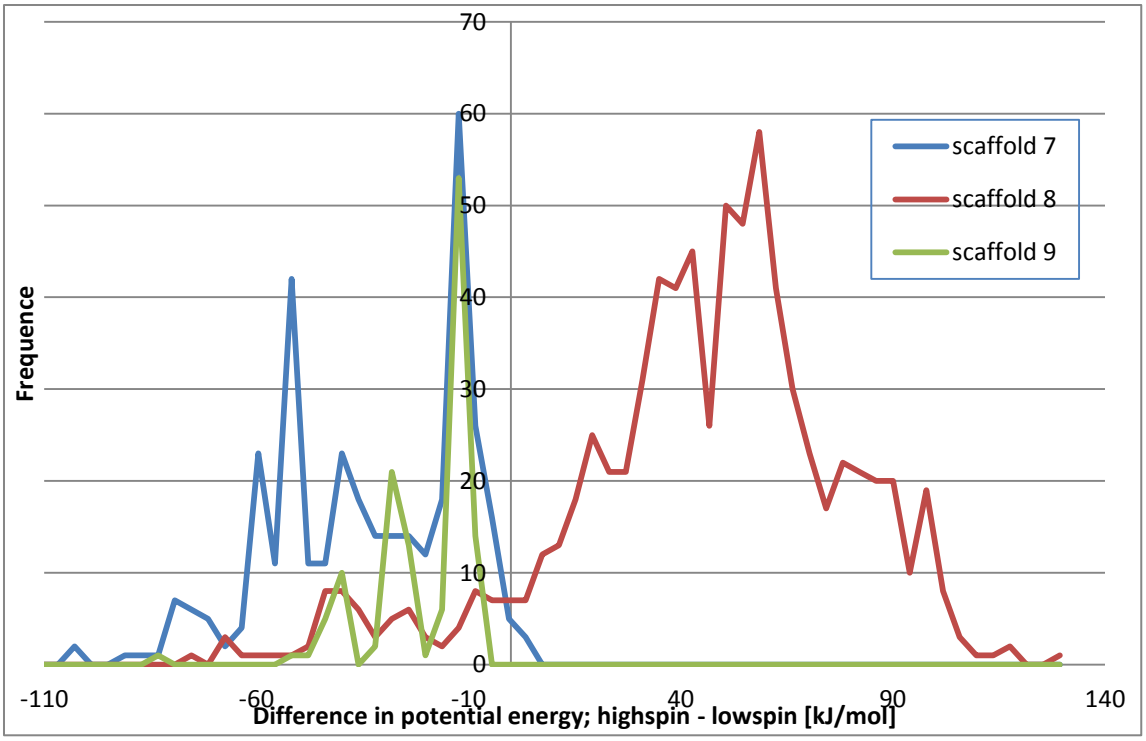


**Figure 4.53:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_160. The histogram includes all good molecules built on the basis of scaffolds 7, 8 and 9.

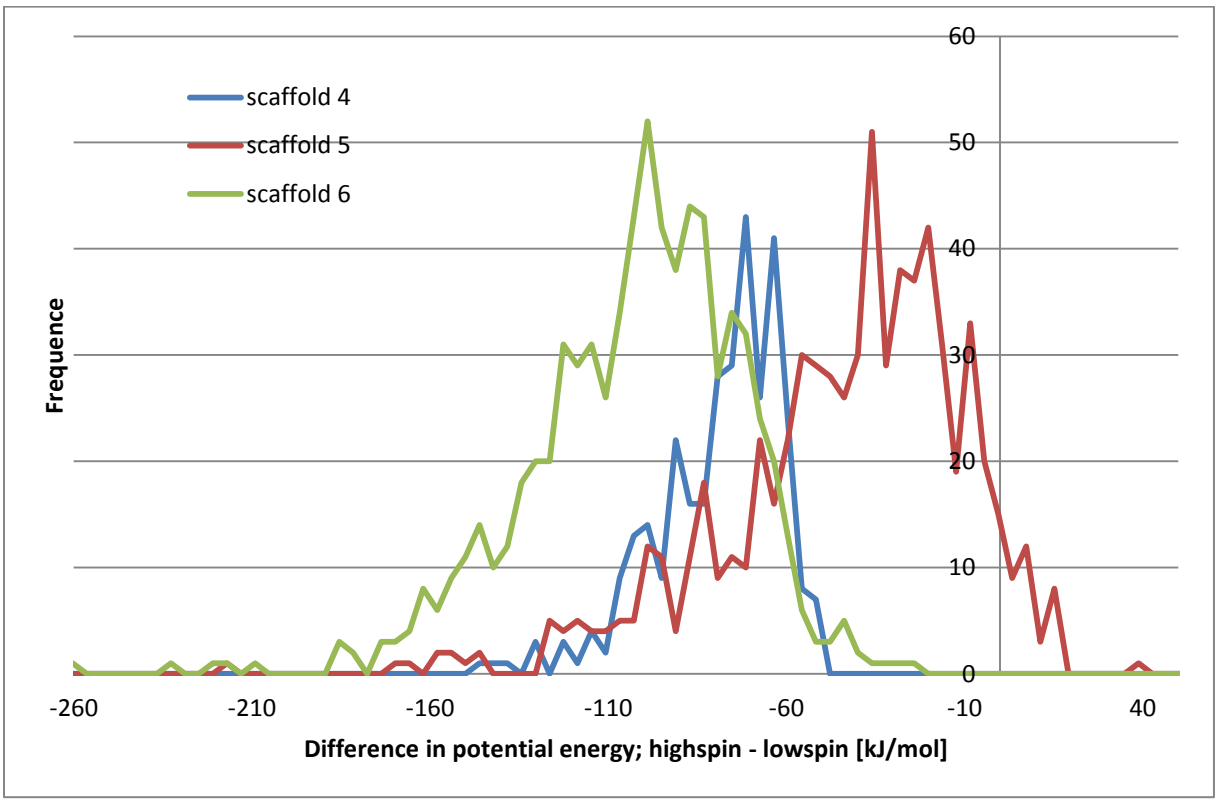


**Figure 4.54:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_160. The histogram includes all good molecules built on the basis of scaffolds 4, 5 and 6.

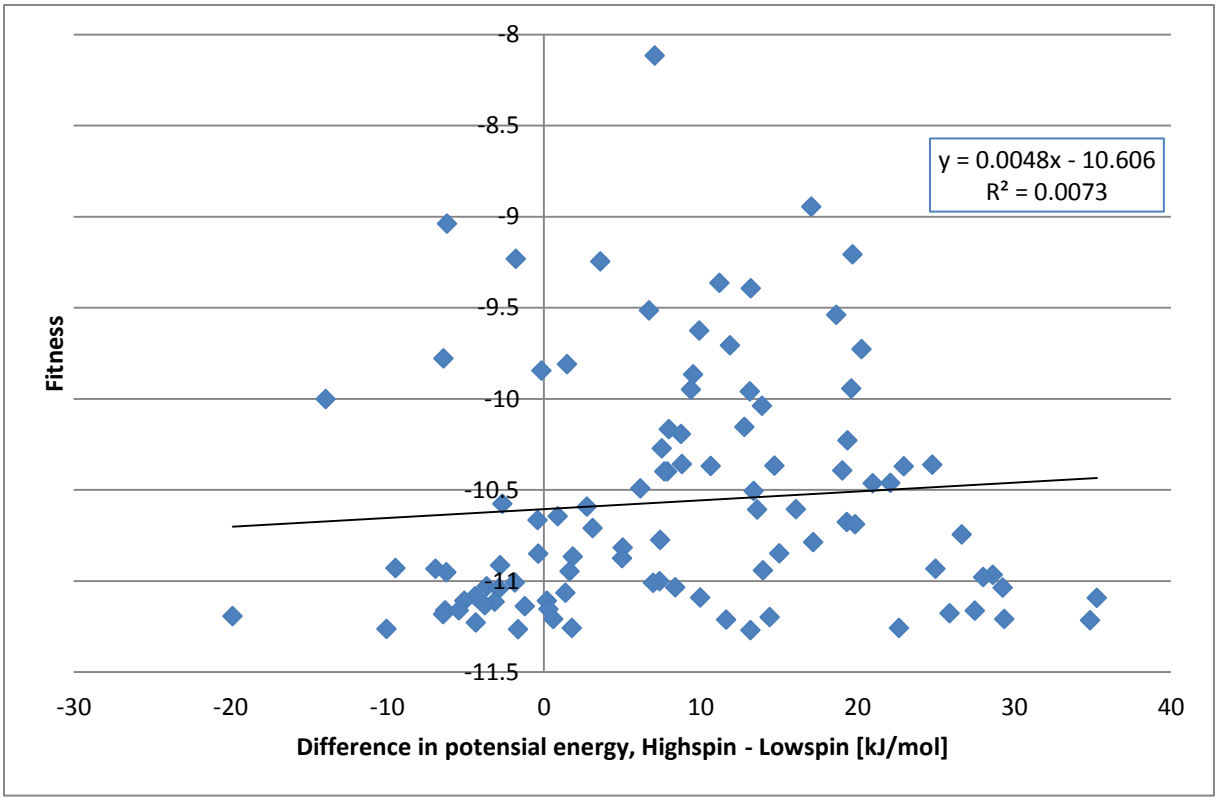


**Figure 4.55:** Plot showing the difference in potential energy by calculating $U_{HS} - U_{LS}$ against fitness in Denoptim run with internal id run_160. This plot shows the 100 fittest molecules in the run and include a trendline with $R^2$ of 0.007

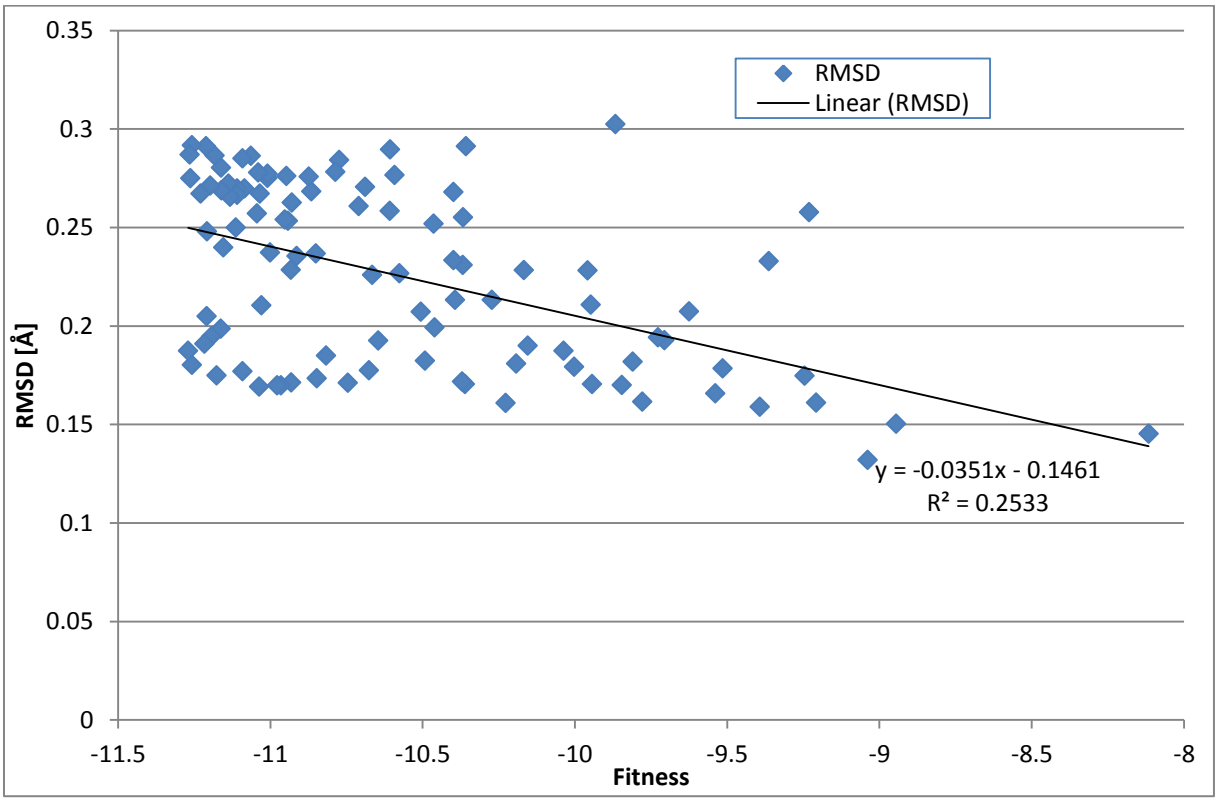


**Figure 4.56:** Plot showing fitness against RMSD for the 100 fittest molecules in Denoptim run with internal id run_160. Also included is the trendline with $R^2$ of 0.253

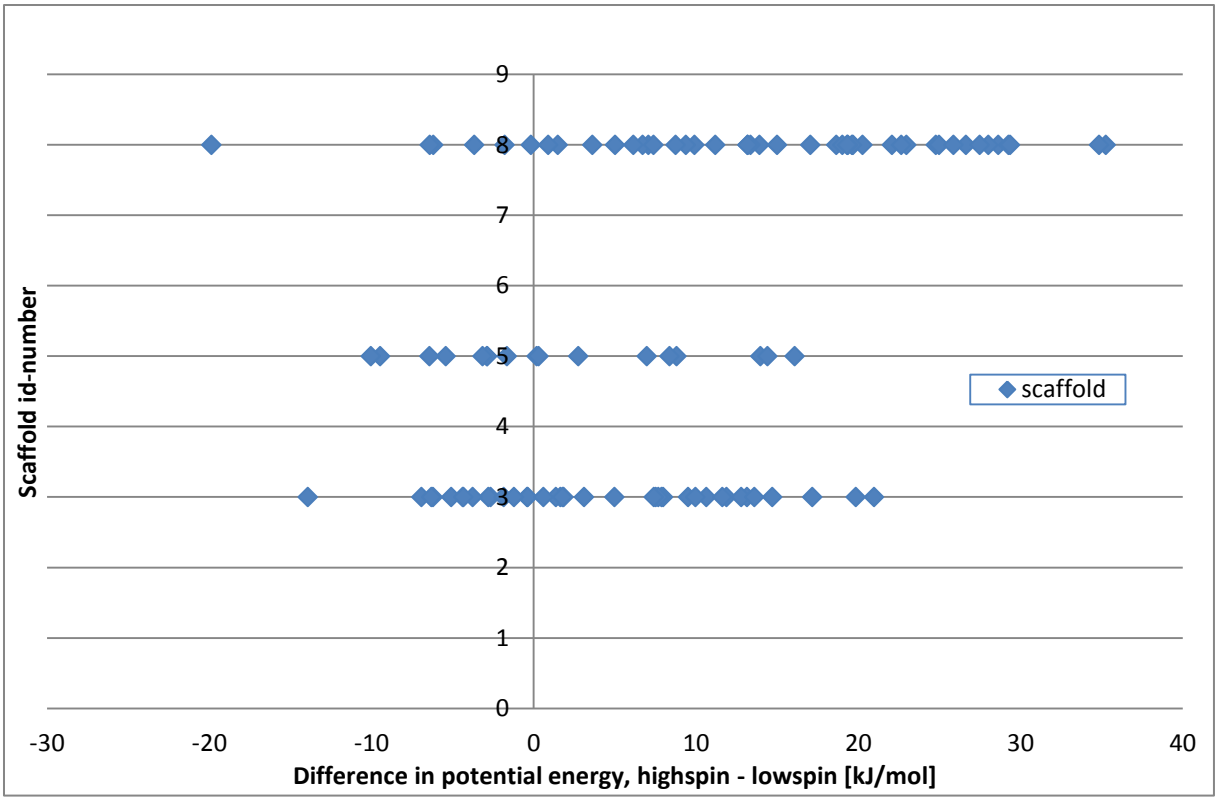


**Figure 4.57:** Plot showing scaffold id-number against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_160. The plot includes the 100 fittest molecules.

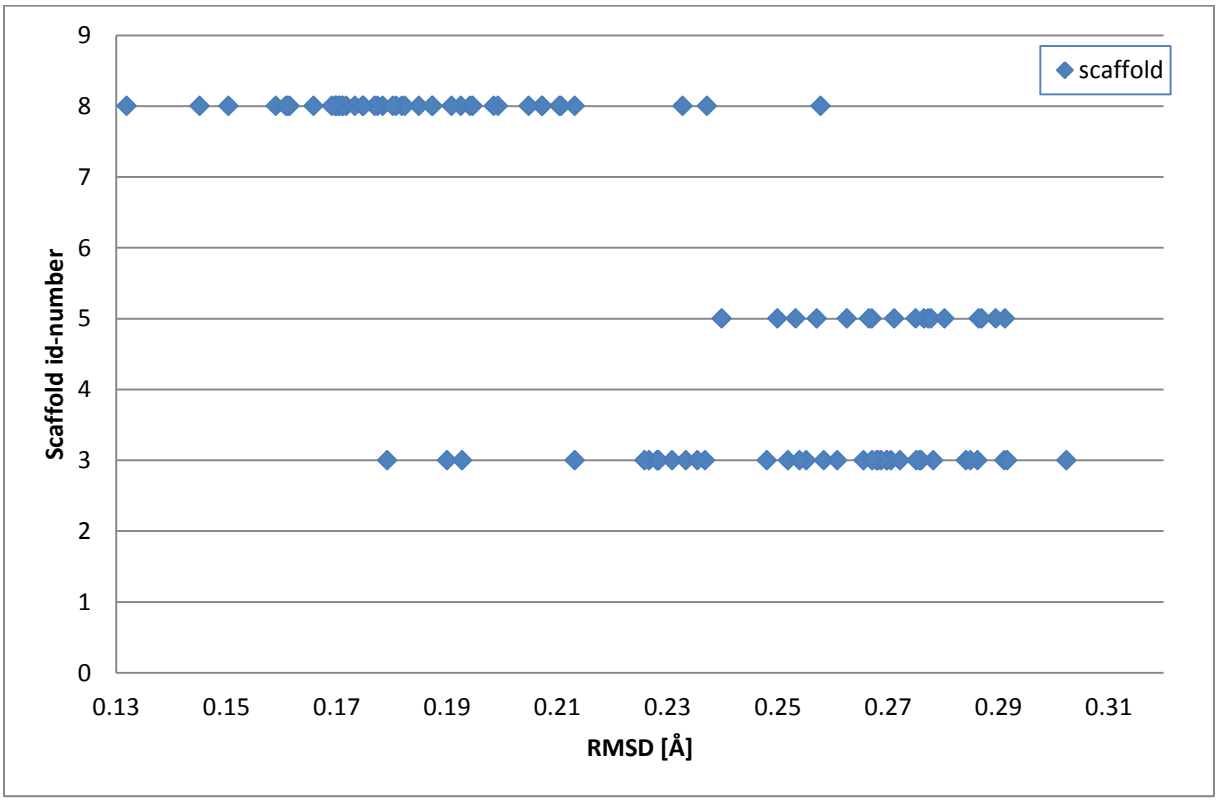


**Figure 4.58:** Plot of scaffold id-number against RMSD for the 100 fittest molecules in Denoptim run with internal id run_160.

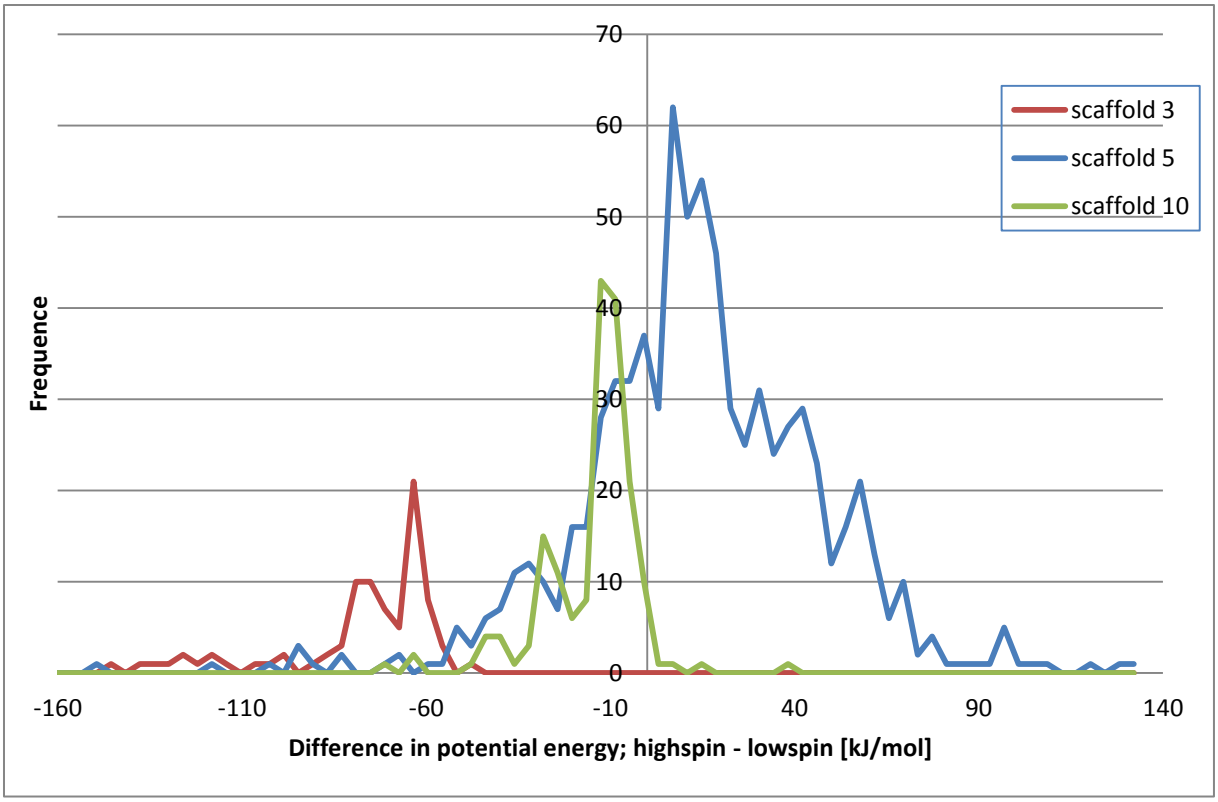


**Figure 4.59:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_201. The histogram includes all good molecules built on the basis of scaffolds 3, 5 and 10.

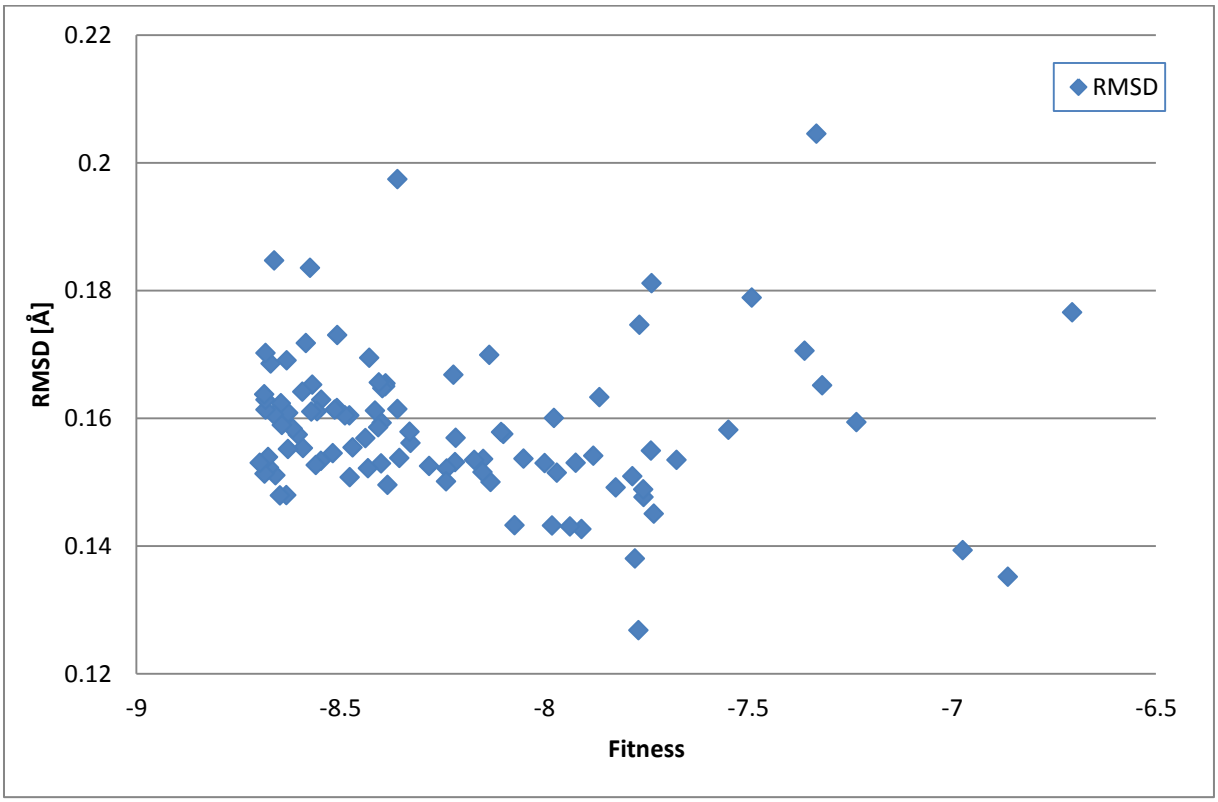


**Figure 4.60:** Plot showing fitness against RMSD for the 100 fittest molecules in Denoptim run with internal id run_201.

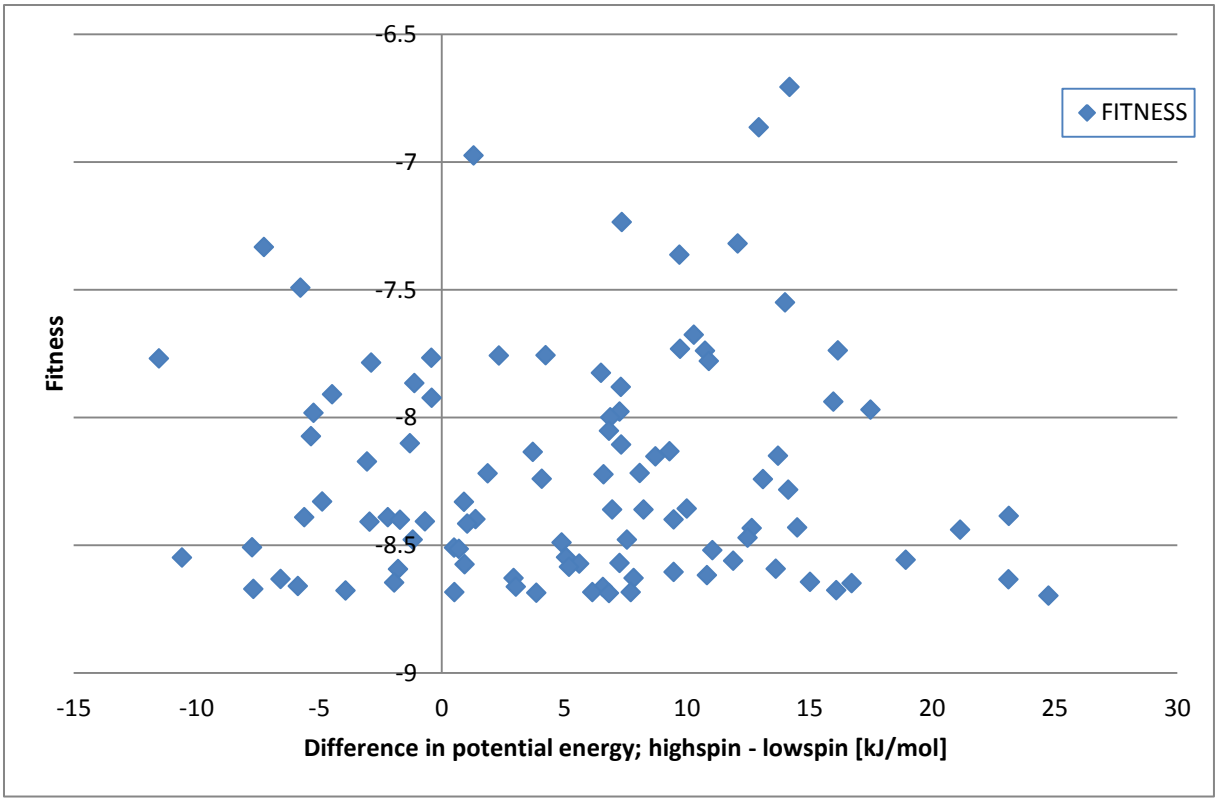


**Figure 4.61:** Plot showing the difference in potential energy by calculating $U_{HS} - U_{LS}$ against fitness in Denoptim run with internal id run_201.

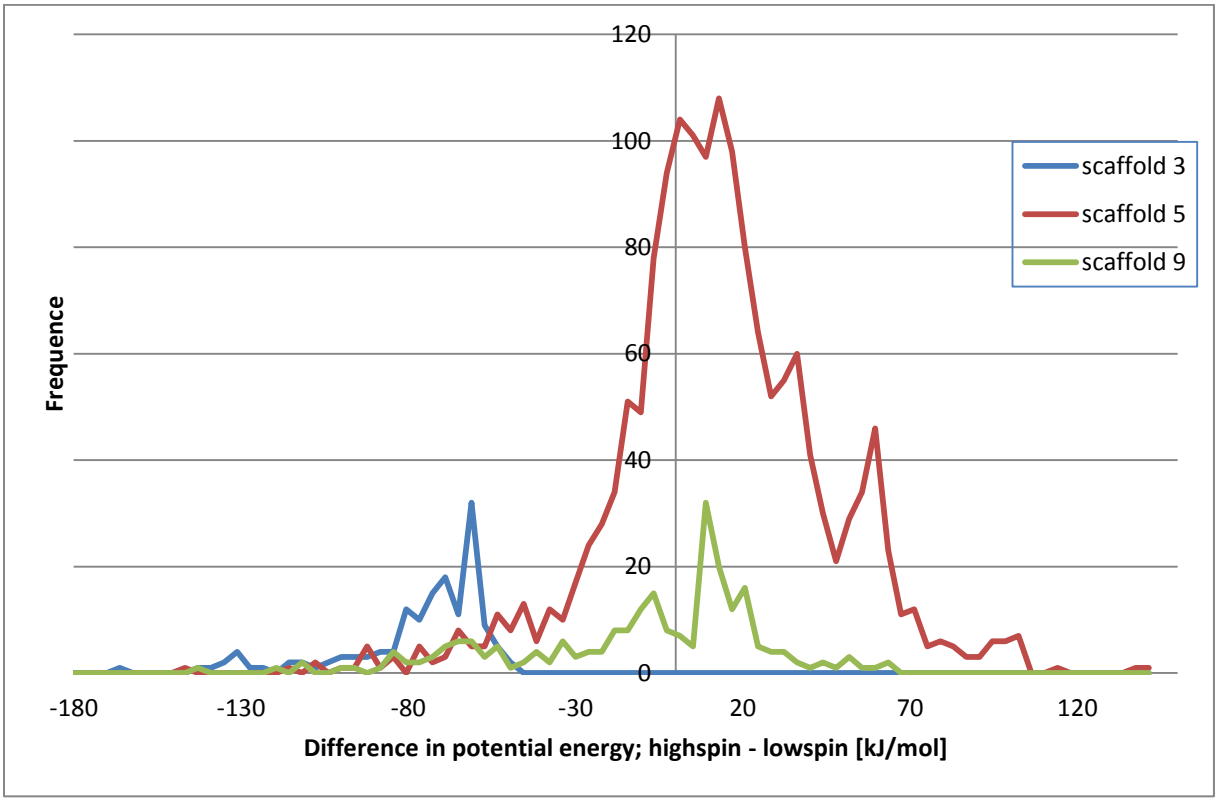


**Figure 4.62:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_310. The histogram includes all good molecules built on the basis of scaffolds 3, 5 and 9.

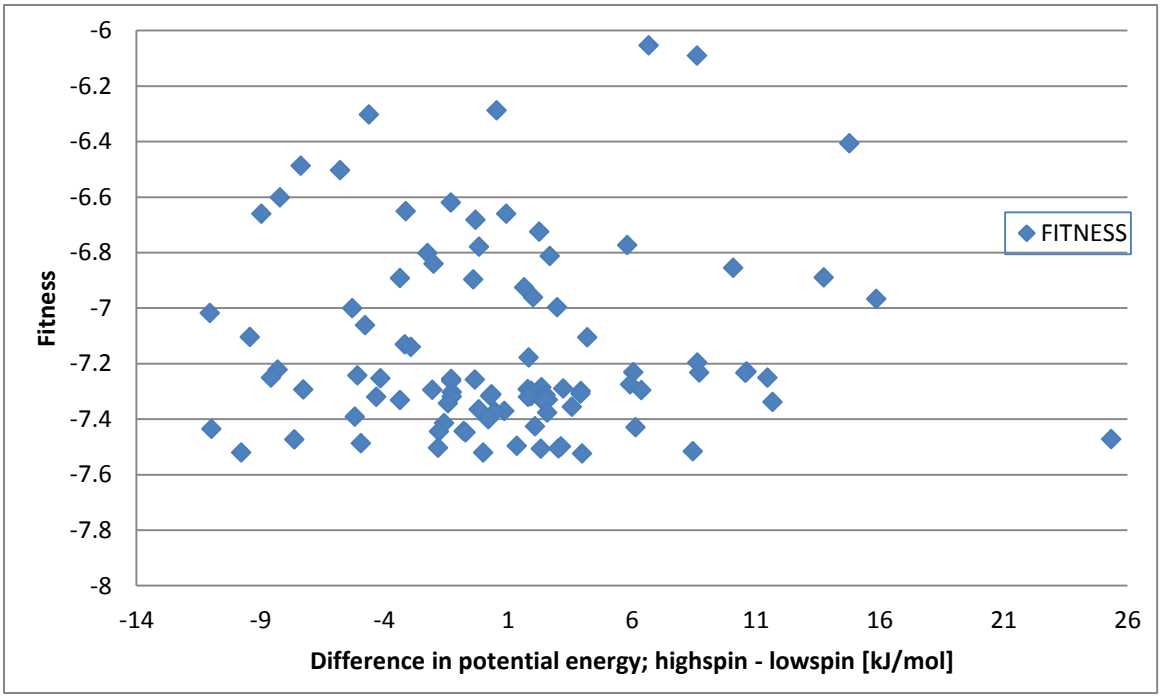


**Figure 4.63:** Plot showing the difference in potential energy by calculating $U_{HS} – U_{LS}$ against fitness in Denoptim run with internal id run_310.

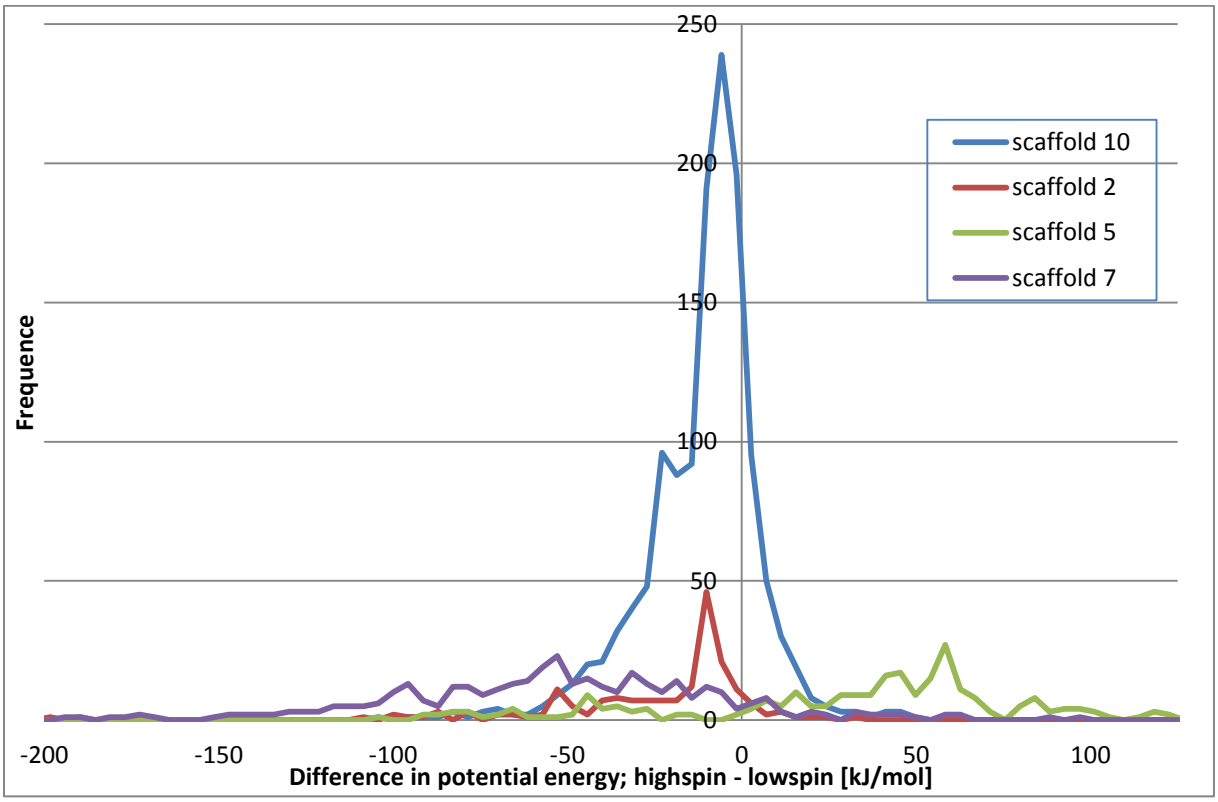


**Figure 4.64:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_325. The histogram includes all good molecules built on the basis of scaffolds 2, 5, 7 and 10.

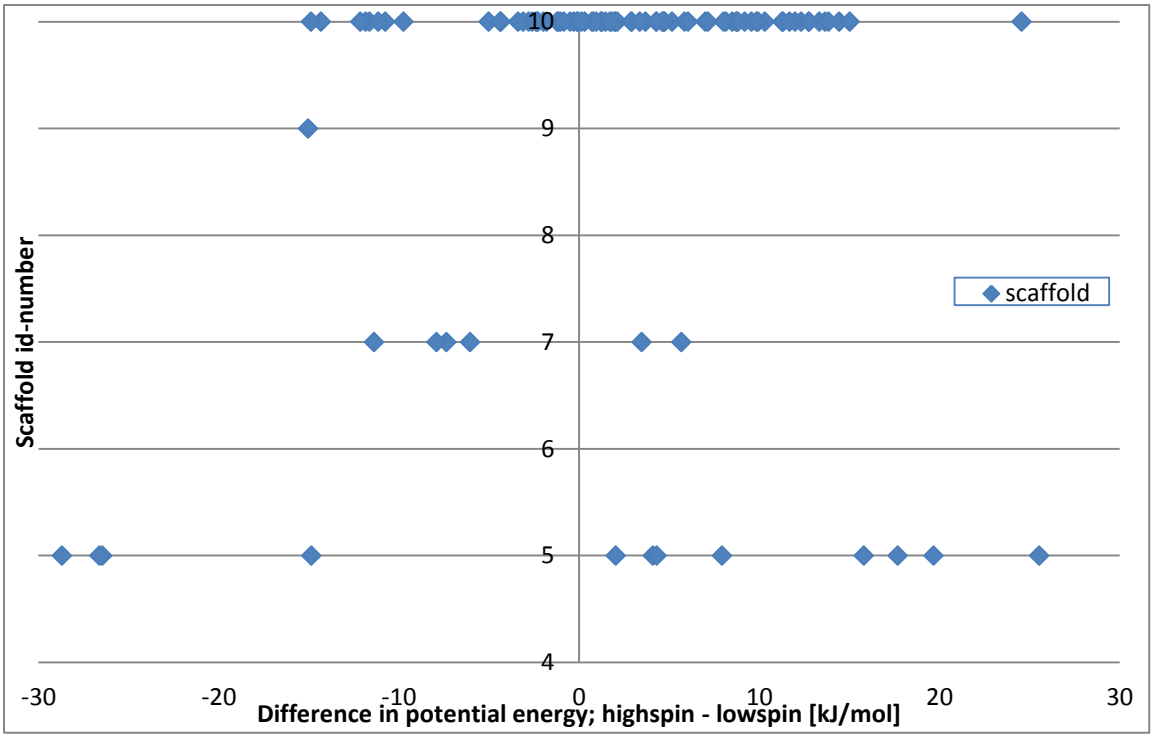


**Figure 4.65:** Plot showing scaffold id-number against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_325. The plot includes the 100 fittest molecules.

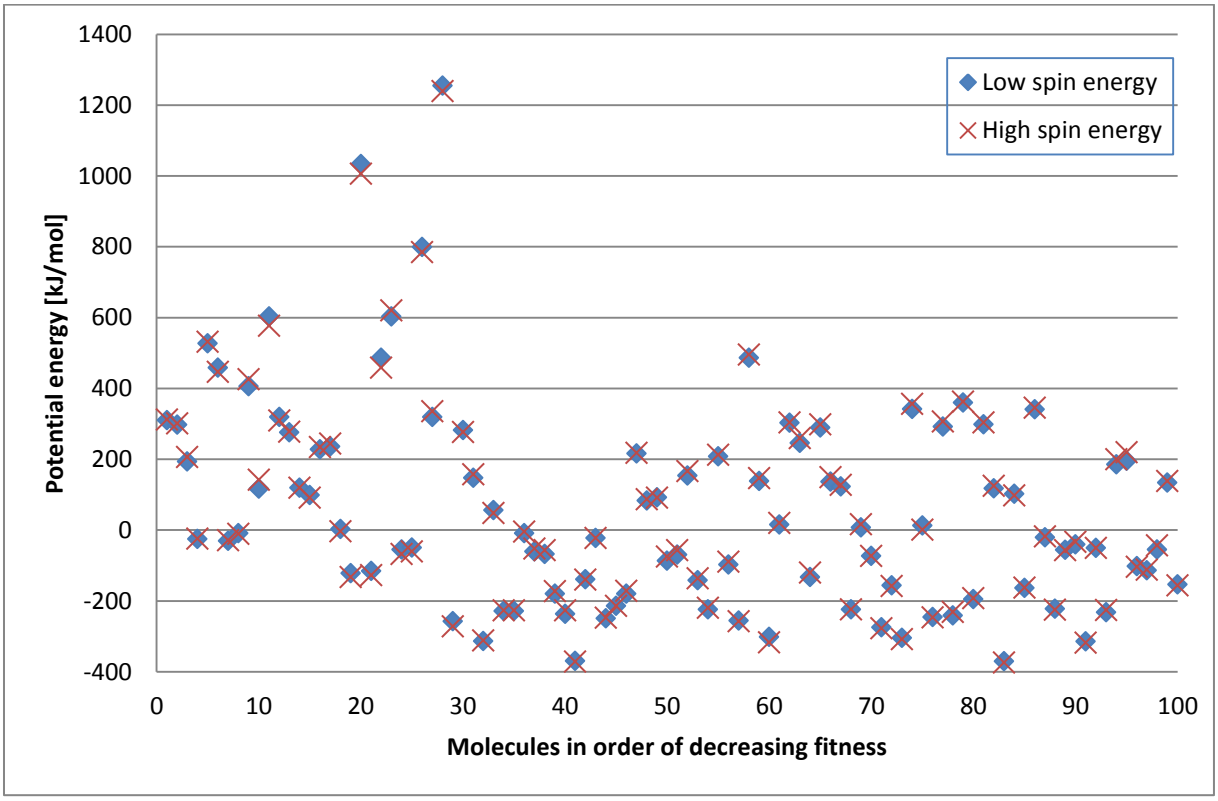


**Figure 4.66:** Plot showing the potential energies for both highspin and lowspin states for the 100 fittests molecules in Denoptim run with internal id run_325. The molecules is sorted in order of decreasing fitness.

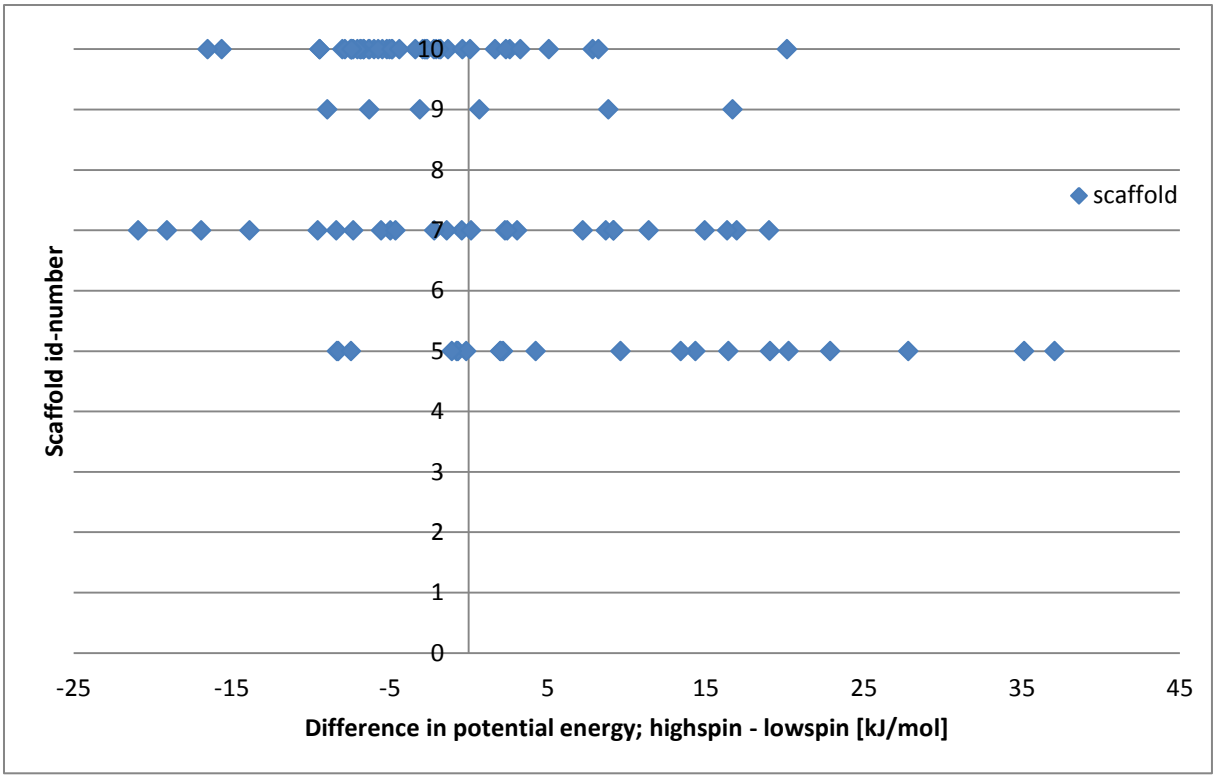


**Figure 4.67:** Plot showing scaffold id-number against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_323. The plot includes the 100 fittest molecules.

## 5. Discussion:

This discussion-section will use the same division into subchapters as used in the result section. this means the 1st. part is programming, the second filterin, scaffolds and fragments, the 3rd. results from conformational and MD on the scaffolds and the 4th. results from Denoptim. Still then applicable using results from a different section will be done, this is especially true for the fitness function since it is used everywhere.

### 5.1: Discussion of the programs and the fitness function.

Before delving into the programs, a look on the fitness function is in order.

#### 5.1.1 Discussion of the fitness function

The fitness function was meant to be low long from the spin-crossover-divide and high near SCO and with the last terms in the function indicating if it's easy to cross between them. A problem here is close to SCO the first term is close to zero while the other term can still be fairly large. This effect is showing up for Denoptim run_160 as shown in figure 4.50 where the $R^2$ is only 0.007 meaning there aren't any correlation between fitness and the difference in potential energy between highspin and lowspin state. By looking on the full run_160 population in figure 4.46 things goes more to a linear state away from SCO indicating the fitness better catches the wanted behaviour. To hopefully improve things decreasing the constant c in the fitness-function as shown in equation 1 will be worth a try.

#### 5.1.2 Discussion of the main LFMM script:

As seen in the flowchart in figure 4.1 the LFMM-functionality is loaded here. This steers all calculations to use this script. By using a script makes it easy to load LFMM if desired something that is more difficult if uses functions. Additionally to handle all the parameters users can specify, handling the logging and remembering to close the logfile at the end and the benefit of pre-loading a database of molecules is all positive in my opinion. Steering all processing through a common script therefore has it's advantages but the inability to use most functions directly is a disadvantage.

Appart for someone possibly wanting a separate script or executable for doing any individual steps steering all processing through this script regardless of run from the command-line, through the graphical DommiMOE-window or by the fitness-script of Denoptim does not seem to give any problems.

In case both spinstates gets calculated in serial this means as shown in figure 4.2 the highspin calculations is always done first. One reason for doing this is highspin is normally faster than lowspin and any molecule going bad often goes bad in both spinstates. Appart for this there is not much to tell about the main script appart for it seems to work as it should as the results shown in chapters 4.2, 4.3 and 4.4 shows.

#### 5.1.3 Discussion on the function marked "do computations" in figure 4.2.

In the flowchart as shown in figure 4.3 many steps is done. The step naturally coming first is loading in a molecule. Any type of filtering on the molecule is done afterwards. In case any LFMM-calculations is done the filtering is enforced but user can in

practice disable all filtering except LFMM only accepts a single transition-metal. If only filtering is desired like for the results in chapter 4.2 all further processing on this molecule is halted as indicated in figure 4.3.

The de-tour to make a LFMM-settings-file is done afterwards followed by one or more of the time-consuming steps. In practice at least one step is always skipped since MD does not need the single point calculations.

Taking a look on the individual steps starting with the flowchart in figure 4.4 a necessary step is to remove any molecule already loaded. Unless any previous molecule is removed there will be 2, 3, 4… molecule present and any calculations on multiple molecules would not give any good answer. The actual code for this step is shown in appendix 9.1. The reason for not reloading the forcefield unless it is necessary is due to this reloading takes significantly longer time than to load a molecule and do filtering on it. To be on the safe side the molecule is always reloaded before creating an LFMM settings-file meaning the forcefield can be reloaded many times for a single molecule. Having to disable the electrostatics after loading should in theory not be needed since LFMM ignores electrostatics but in practice the calculated potential energies is significantly different with electrostatics enabled.

The code in appendix 9.1 also shows how the try…catch works in SVL, by using a task_call and collecting the errorCode.

The flowcharts in figure 4.5 and 4.6 shows the filtering. As mentioned in the introduction it can be necessary to set the ionisation for MOE to select the correct atomtype. During this necessary step also counting how many transition-metals is present is a possibility with no reason to let pass by and a test for one and only one transition-metal is a good first step to include filtering.

The remaining tests in the flowchart figure 4.5 is all light tests and the same is the case for negative carbons and metal bond order in the start of flowchart shown in figure 4.6. The two last steps on the other hand is the potentially most time-consuming and these tests is for this reason left until last.

The generation of a LFMM settings-file is detailed in the flowchart in figure 4.7. As the flowchart reveals the various parameters used here are hard-coded, one of the very few functions still using hard-coded numbers and not allowing getting them as part of the parameters issued to the main script in figure 4.1.

The flowchart for stochastic conformational search is shown in figure 4.8. The logic for allowing multiple loops is to allow doing a short initial conformational search for so multiplying with a constant. If where are no improvement in a conformational search after doubling no further conformational search is done. Demanding 3 or more similar results is one of the configurable parameters.

The step including reloading the best molecule at the end of the flowchart in figure 4.8 is to make sure any further calculations is using the molecule with lowest potential energy and not some other molecule.

In the flowcharts for geometry optimization and single point calculation shown in figure 4.9 and 4.10 a step that looks uneccessary is the "reload lfmmFileName" and "reload LFMM settings file". This is especially the case for single point in figure 4.10 since the file was just created some nanoseconds before. This step is to make sure DommiMOE doesn't suddenly complain about missing parameters or uses the wrong parameters. As for the step with "delete LFMM settings file again" this has to do with the desire of supporting different spin-states and also to immediately support different transition-metals than iron if this is necessary. For this reason metal-name, ionisation and spin-state is included in the filename as described in the flowchart in figure 4.7. The problem here is more than once during running multiple Denoptim-calculations at the same time they failed due to one instance was in the process of creating a new file while another failed to read the now empty file. To work-around this problem adding the time into the filename was done. With maybe thousands of calculations done this would unfortunately leave behind thousands of LFMM-parameter-files and for this reason the file is deleted.

The flowcharts for MD is shown in figure 4.11 and 4.12. Due to the current problems with MD they are not commented any further.

### 5.1.5 Discussion on the main parts of the fitness-script for Denoptim.

The main part of the fitness-script is shown in the flowchart of figure 4.13. This includes a step with the 3D-builder to make sure the stereochemistry on the start-conformation is correct. Afterwards a very short highspin DommiMOE run is done. For one this will filter-out any invalid molecules and for another it enables getting a conformation within LFMM-forcefields potential and this conformation is used to generate the UID that is used by Denoptim to filter for duplicate molecules.

The last part of the fitness-script depends on serial or parallel run, with serial being depicted in figure 4.14 and parallel in figure 4.15. The flowchart for serial in figure 4.14 is fairly simple by just containing a loop to not error-out the Denoptim-run in case the connection to DommiMOE's licensing server is lost. The parallel method as seen in figure 4.15 is more detailed. The reason to include this is that in case it is the lowspin calculation that fails having finished all parts of the highspin calculation would be a waste of time. One problem with parallising is highspin normally will finish faster and this can lead to idle cpu-core. Another weakness is it's missing any recovery from losing connection to licensing server.

## 5.2 Discussion on filtering, scaffolds, fragments and compability matrices.

Based on table 4.1 having to manually filter 1779 molecules looking for possible scaffolds looks like a time-consuming process. While running DommiMOE as filtering does work, since any transition-metal not being a single iron where iron is only directly bonded to nitrogen will lead to any kind of LFMM-calculations to error-out. While DommiMOE successfully tried and failed doing a geometry optimization on 713715 molecules from the CCSD does show the code for handling errors is working. One negative point is among the 627 molecules passing this part there was iron-complexes with only 3 bonded nitrogen and many pyridines and other ring-structures. Since the limited LFMM-forcefield does not correctly handle these types of complexes LFMM successfully wasting time on them is discouraging.

By using all the various filtering-options as shown in flowcharts in figure 4.5 and 4.6 as shown in table 4.1 only 26 molecules passed the filtering. All of these molecules fulfilled the requirements of $Fe^{2+}$ with 6 N-atoms bonded and no pyridines or similar rings. These results shows the minor aim of filtering worked and it also shows why including filtering as part of MOE/DommiMOE is an advantage since without filtering DommiMOE wasted resources on 601 invalid molecules.

While the ten scaffolds was all picked from either a single molecule based on molecular weight, visual inspection and the INCHI-keys, the results as shown in table 4.2 includes an 11th. unique INCHI-key not selected as a scaffold. Based on visual inspection of the molecules in this group containing the molecules with CCSD refcodes CODQAO, EYEWOV, GOXSET, LEVQAF, QIBZUX, RENHUO and XUZVAP there are multiple isomers inside this group. This gives a set of *fac*/*mer*-isomers and an example of both is shown in figure 5.1 and 5.2. In figure 5.1 the *fac*-isomer based on CCSD refcode QIBZUX[42] is shown while figure 5.2 shows *mer*-isomer based on refcode GOXSET. Due to the fact the INCHI-key is the same even the isomers are different there are no easy method to discriminate between molecules. Additionally a molecule built by Denoptim as one isomer can often swap to a different isomer during a conformational search. While using bulky ligands can block this swapping with small ligands with often nothing besides hydrogen or methyl added this will not be the case in this instance. Based on these reasons these molecules was rejected and the number of scaffolds was limited to ten.

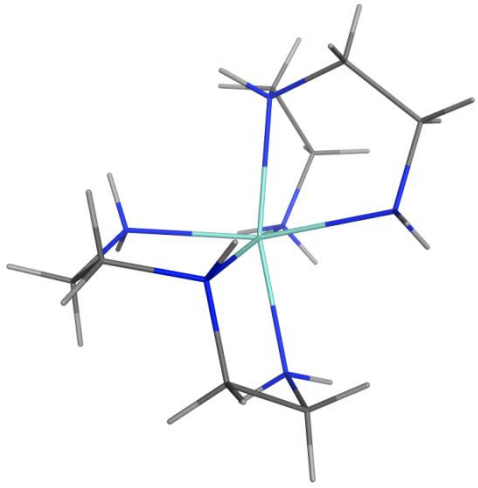

**Figure 5.1:** A molecule with CCSD refcode QIBZUX is a *fac*-isomer.

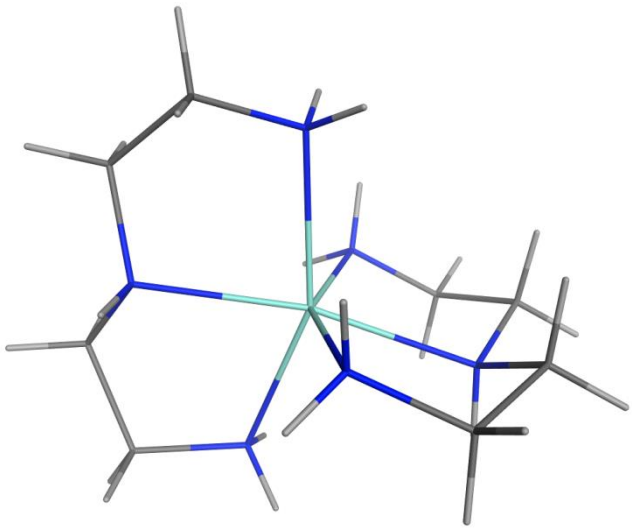

**Figure 5.2:** A molecule with CCSD refcode GOXSET is a *mer*-isomer.

For the 10 choosen scaffolds they all fulfills the desired requirements of being $Fe^{2+}/N_6$ ammine-only ligands. For other types of fragments as depicted in figure 4.19 the choises was based on a desire to use fairly small fragments since being close to nitrogen and the metal has the largest chance of influencing the metal-ligand and by this the SCO. The hydrocarbons is all small and readily synthesizable and the same is the case with aromatic rings. Including hydroxyl extends the chemistry and this can drag some electrons away from nitrogen making the binding to metal weaker. Ether is also fairly common and is included with the last being chlorine. While all of these fragments is easily synthesizable how easily they're added to ammine-ligands without disrupting these too much is unknown at this point.

As for not including other types of atoms, including more nitrogens was decided against since these can start competing with the metal for binding. Strong ligands like carbonyl was also decided against to not disrupt things.

When it comes to the compability matrix, one requirement here is to never connect chlorine or oxygen directly to nitrogen. Other than this nitrogen and carbon is treated separate meaning there will never be any crossover between fragments bonded to nitrogen and carbon.

**5.3 Discussion about conformational searches and MD on ten molecules.**
From the combination of results in table 4.3 one immediately apparent problem is the instability between runs especially when it comes to the difference in potential energies between spinstates. Looking on DETTOL the average from single runs is 4.97 kJ/mol while the standard deviation is 5.71 and this indicates a huge variation.

By comparing the results in table 4.3 against the reference-numbers things looks even bleaker. While HUMBEX is allowed to be different, since it is not compared to the same molecule of the five other molecules with included reference-energies only DETTOL has average on both sides of the reference. While PAZXAP is even strongly on the wrong side of the spinstate, PURYIK and FEBMAB has larger energy-difference than expected while LOTSES has less. Of the 3 remaining molecules where only the spinstate is known KIBWAV should be lowspin but is detected as highspin.

A closer look on some of the other tables is in order before drawing any conclusions, starting with table 4.4 and KIBWAV. The highspin and lowspin energies for KIBWAV is additionally plotted in figure 4.20 and while unstable at short conformational searches is not a big problem the one at 4500 is also markedly off and this is not a good sign. A look at table 4.5 shows the results of multistep conformational search and the huge variations even for the longer-running is negative. The 500_3_new managed to get the same answer 3 times. Looking on both table 4.4 and 4.5 and remembering KIBWAV should be lowspin, of the 43 conformational searches only in one instance, for 400_3_old is the result lowspin.

The superposition between two different lowspin runs is shown in figure 4.21 and this shows some distortion with RMSD being 0.1829 Å. Finally the plot in figure 4.22 between RMSD and conformational search id number is not a good plot, with 15 different pairs to describe 43 conformational searches.

By looking on the plot of TIYPIC in figure 4.23 things looks much better even the RMSD is split in two different groups. By looking on the overlaps first in figure 4.24 things looks good and this incidentally also clearly shows the highspin state has larger bondlengths than lowspin. The overlap in figure 4.25 on the other hand is clearly not good. Rotating around a bond can be a reason for this being so far off since if the part with C3 is held constant while the rest is allowed to turn it will overlap. Based on TIYPIC being one of the most consistent molecules in table 4.3 figure 4.25 shows that despite stable energy there can still be multiple conformations.

PURYIK is shown in figure 4.26 and 4.27 and while the start with short number of conformational search is bad things seems to improve somewhat for longer

searches. How the fitness is good but the potential energy difference between highspin and lowspin as shown for 5000 conformational search in figure 4.27 is on the other hand not so good.

Going to PAZXAP the plot in figure 4.28 is interesting with highspin being very stable while lowspin being more variable. Two different superpositions in the highspin state as shown in figure 4.29 and 4.30 reveals some large deviations from expected structure. This large deviation is possibly the reason for PAZXAP being completely wrong spinstate as shown in table 4.3.

Plots with LOTSES  is shown in figure 4.31 and 4.32 and based in RMSD this shown different geometry even LOTSES has negligible variation in energies as shown in table 4.3.

The plot of KIBWEZ and potential energies in figure 4.33 starts off bad but stabilizes after 1000 conformational steps. the RMSD in figure 4.34 on the other hand still shows some variations.

DETTOL copies the behaviour of KIBWEZ with potential energies as shown in figure 4.35 being unstable at low number of conformational steps but good at longer. The RMSD in figure 4.36 is still variable at longer lengths.

HUMBEX potential energy behaviour in figure 4.37 is faster at stabilizing and the RMSD plot in figure 4.38 shows two distinctly different geometries.

The last molecule is FEBMAB with RMSD plot in both figure 4.39 and 4.40 showing two distinctive geometries. The superposition of lowspin against highspin for both possibilities is shown in figure 4.41 and 4.42.

Based on all the energy-graphs, it seems atleast the energy stability gets better as number of conformational searches passes 1000. All molecules also gives at least two different RMSD even in cases with very stable energies. PAZXAP and KIBWAV on the other hand continues behaving badly and LFMM can not currently handle these two molecules.

### 4.3.1 Discussion of results from molecular dynamics.

While some of the results from conformational search was not as desired, as shown in table 4.6, figure 4.43 and figure 4.44 things is much worse for MD. With only one of 10 molecules managing a lowspin MD to 300 ps without crashing all the time only some of the times in it's current state MD isn't usable. The reason for one of the crashes is shown in figure 4.45 with badly distorted bond-lengths and atoms out of their expected positions.

While using an older version of the customized LFMM-forcefield is a possibility, one point against this is it is less accurate[39] and another is the problems with replicating older results[37].

### 5.4 Discussing results from Denoptim runs.

With the contamination of many of the results by PAZXAP being completely off in some cases the end-result is not very trustworthy. Still a quick overview of the results follows.

As table 4.7 shows, there are some molecules rejected as invalid. These are rejected by the filtering included in the software. Appart for this run_325 have a much larger number of duplicates than the other runs.

From the overview of the full runs as seen in figure 4.46, 4.47, 4.48, 4.49 and 4.50 one point is the "survival of the fittests" often has a fast grow-rate and ends with higher than the "keep growing" runs. For all runs except the run_325 the best molecule was generated by EA.

Seeing on run_160 starting with figure 4.51 a plot of difference in potential energy against fitness shows good behaviour far from the y-axis but the closer to y-axis the more smeared-out.

Figure 4.52, 4.53 and 4.54 shows only one of the scaffolds is clearly lowspin. Figure 4.55 shows the less-than linear nature between difference in potential energy and fitness. The plot of fitness against RMSD in figure 4.56 is on the other hand slightly better.

For the other runs, the plots of difference in potential energy against frequence shows the same scaffold dominates on the highspin-side in run_201, figure 4.59 and in run_310 in figure 4.62. In run_325 as seen in figure 4.64 on the other hand scaffold 10 is dominating.

Figure 4.60 shows fitness against RMSD for run_201 and 4.61 shows fitness against difference in potential energy also for run_201. Neither gives a very linear result.

The difference in potential energy against fitness for run_310 is shown in figure 4.63.

The spread in difference in potential energies against scaffolds for run_325 is shown in figure 4.65 while the same type for run_323 is shown in figure 4.67.

Since one scaffold completely dominated over the others in run_310 no plot of spread against scaffold is included.

The last included is a graph from run_325 showing lowpin and highspin energies is close to eachother.

While this is just some of the main results from the runs, PAZXAP dominating the others as it especially did in run_310 and at the same time knowing the results for PAZXAP is wrong drilling much deeper does not give any extra information.l

The Denoptim runs does on the other hand show the fitness-script works and after finding a solution for PAZXAP things are ready to run.

## 6. Conclusion:

All results from Denoptim-runs, conformational searches, MD and the filtering as described in chapter 4 and discussed in chapter 5 does not reveal any problems due to the software. The results for filtering worked very good and the inclusion of running DommiMOE revealed the software continued working even some molecules in a database was invalid. The software therefore seems to be very robust.

When it comes to conformational searches, needing atleast 1000 steps is adviceable. The problems on PAZXAP and KIBWAV on the other hand is harder to solve.

While the code for running MD works there are multiple problems with MD making this not an option at this point.

## 7. Future work:

For the coding-part, the hardcoded numbers for LFMM settings-file and in fitness-function should be replaced. Another limitation is currently the lack of support for the middle spin-state that can be present for some metals. A 3rd. limitation is the currently no support of metals with different ionisations.

The runs with conformational search revealed problems with some molecules and a possible fix for this is desirable.

## 8. Bibliography:


1. Foscato, M., et al., *Automated Design of Realistic Organometallic Molecules from Fragments.* Journal of Chemical Information and Modeling, 2014. **54**: p. 767-780.
2. Hay, B.P., *De novo structure-based design of anion receptors.* Chemical Society Reviews, 2010. **39**: p. 3700-3708.
3. Schneider, G. and U. Fechner, *Computer-based de novo design of drug-like molecules.* Nature Reviews Drug Discovery, 2005. **4**: p. 649-663.
4. Pirard, B., *The quest for novel chemical matter and the contribution of computer-aided de novo design.* Expert Opinion on Drug Discovery, 2011. **6**(3): p. 225-231.
5. Chu, Y., et al., *An evolutionary algorithm for de novo optimization of functional transition metal compounds.* Journal of the American Chemical Society, 2012. **134**: p. 8885-8895.
6. Venkatraman, V., et al., *Evolutionary de novo design of phenothiazine derivatives for dye-sensitized solar cells.* Journal of Materials Chemistry A, 2015. **3**: p. 9851-9860.
7. Deeth, R.J., *The ligand field molecular mechanics model and the stereoelectronic effects of d and s electrons.* Coord. Chem. Rev., 2001. **212**: p. 11-34.
8. Deeth, R.J., N. Fey, and B. Williams-Hubbard, *DommiMOE: An implementation of ligand field molecular mechanics in the molecular operating environment.* J. Comput. Chem., 2004. **26**(2): p. 123-130.
9. Shatruk, M., et al., *Symmetry-breaking structural phase transitions in spin crossover complexes.* Coord. Chem. Rev., 2015. **289-290**: p. 62-73.
10. Halcrow, M.A., *The foundation of modern spin-crossover.* Chem. Commun. (Cambridge, U. K.), 2013. **49**(93): p. 10890-10892.
11. Tolla, A.S., et al., *Spin-State Tuning in Iron(II) Triazamacrocyclic Complexes.* European Journal of Inorganic Chemistry, 2013. **2013**(12): p. 2115-2121.
12. Rodrigues, T. and G. Schneider, *Flashback forward: reaction-driven de Novo design of bioactive compounds.* Synlett, 2014. **25**(2): p. 170-178.
13. Reymond, J.-L., *The Chemical Space Project.* Accounts of Chemical Research, 2015.
14. Buda, C., et al., *De Novo Structural Prediction of Transition Metal Complexes: Application to Technetium.* Inorg. Chem., 2002. **41**(8): p. 2060-2069.
15. Hay, B.P. and T.K. Firman, *HostDesigner: A Program for the de Novo Structure-Based Design of Molecular Receptors with Binding Sites that Complement Metal Ion Guests.* Inorganic Chemistry, 2002. **41**: p. 5502-5512.
16. Vukovic, S. and B.P. Hay, *De Novo Structure-Based Design of Bis-amidoxime Uranophiles.* Inorganic Chemistry, 2013. **52**: p. 7805-7810.
17. Hay, B.P., et al., *Search for Improved Host Architectures: Application of de Novo Structure-Based Design and High-Throughput Screening Methods To Identify Optimal Building Blocks for Multidentate Ethers.* Journal of the American Chemical Society, 2005. **127**: p. 17043-17053.
18. Steinbeck, C., et al., *The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics.* Journal of Chemical Information and Computer Sciences, 2003. **43**: p. 493-500.
19. *http://dasher.wustl.edu/tinker/*.
20. O'Boyle, N.M., et al., *Open Babel: An open chemical toolbox.* 2011. **3**: p. 33.
21. *Daylight Theory Manual, In Daylight Chemical Information System, Inc.*
22. Clark, A.M., *Accurate Specification of Molecular Structures: The Case for Zero-Order Bonds and Explicit Hydrogen Counting.* J. Chem. Inf. Model., 2011. **51**(12): p. 3149-3157.

23. Paulsen, H. and A.X. Trautwein, *Density functional theory calculations for spin crossover complexes.* arXiv.org, e-Print Arch., Phys., 2012: p. 1-31, arXiv:1206.2247v1 [physics.chem-ph].
24. Saha-Dasgupta, T. and P.M. Oppeneer, *Computational design of magnetic metal-organic complexes and coordination polymers with spin-switchable functionalities.* MRS Bull., 2014. **39**(7): p. 614-620.
25. Ye, H.-Z., C. Sun, and H. Jiang, *Monte-Carlo simulations of spin-crossover phenomena based on a vibronic Ising-like model with realistic parameters.* Phys. Chem. Chem. Phys., 2015. **17**(10): p. 6801-6808.
26. Deeth, R.J., A.E. Anastasi, and M.J. Wilcockson, *An In Silico Design Tool for Fe(II) Spin Crossover and Light-Induced Excited Spin State-Trapped Complexes.* J. Am. Chem. Soc., 2010. **132**(20): p. 6876-6877.
27. Hinchliffe, A., *Molecular modelling for beginners*. 2nd ed. ed. 2008, Chichester: Wiley.
28. Vanommeslaeghe, K., O. Guvench, and A.D. MacKerell, Jr., *Molecular Mechanics.* Curr. Pharm. Des., 2014. **20**(20): p. 3281-3292.
29. Cornell, W.D., et al., *A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules.* Journal of the American Chemical Society, 1995. **117**: p. 5179-5197.
30. Halgren, T.A., *Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94.* Journal of Computational Chemistry, 1996. **17**: p. 490-519.
31. *Molecular Operating Environment (MOE), 2013.08; Chemical Computing Group Inc., 1010 Sherbooke St. West, Suite #910, Montreal, QC, Canada, H3A 2R7, 2013.*
32. *Chemical Computing Group Inc., Molecular Operating Environment (MOE), version 2011.10, https://www.chemcomp.com/MOE-Molecular_Operating_Environment.htm (accessed April 2015).*
33. Slimani, A., et al., *Reparametrization Approach of DFT Functionals Based on the Equilibrium Temperature of Spin-Crossover Compounds.* The Journal of Physical Chemistry A, 2014. **118**: p. 9005-9012.
34. Craig, G.A., O. Roubeau, and G. Aromi, *Spin state switching in 2,6-bis(pyrazol-3-yl)pyridine (3-bpp) based Fe(II) complexes.* Coord. Chem. Rev., 2014. **269**: p. 13-31.
35. Tissot, A., *Photoswitchable spin crossover nanoparticles.* New J. Chem., 2014. **38**(5): p. 1840-1845.
36. Molnar, G., et al., *Emerging properties and applications of spin crossover nanomaterials.* J. Mater. Chem. C, 2014. **2**(8): p. 1360-1366.
37. Foscato, M., R.J. Deeth, and V.R. Jensen, *Integration of Ligand Field Molecular Mechanics in Tinker.* J Chem Inf Model, 2015: p. Ahead of Print.
38. Allen, F.H., *The Cambridge Structural Database: a quarter of a million crystal structures and rising.* Acta Crystallographica Section B Structural Science, 2002. **58**: p. 380-388.
39. Handley, C.M. and R.J. Deeth, *A Multi-Objective Approach to Force Field Optimization: Structures and Spin State Energetics of d6 Fe(II) Complexes.* J. Chem. Theory Comput., 2012. **8**(1): p. 194-202.
40. Tolla, A.S., et al., *Spin-State Tuning in Iron(II) Triazamacrocyclic Complexes.* Eur. J. Inorg. Chem., 2013. **2013**(12): p. 2115-2121.
41. Matsumoto, T., et al., *Nonprecious-Metal-Assisted Photochemical Hydrogen Production from ortho-Phenylenediamine.* Journal of the American Chemical Society, 2013. **135**(23): p. 8646-8654.
42. Jia, D., et al., *New Selenidogermanates with Transition-Metal Complexes as Counterions: Solvothermal Synthesis, Crystal Structures, and Properties of [Mn(en)3]2Ge2Se6 and [Fe(dien)2]2Ge2Se6.* Monatshefte für Chemie - Chemical Monthly, 2007. **138**(3): p. 191-197.

## 9. Appendix:

### 9.1 The function for clearing out any loaded molecules from MOE.

```
/*
** @author Kjell Nedrelid
** @since 2015-02-20
**
** Clear out any loaded molecule from MOE and reload forcefield.
**
** @param forceFieldFileName : The filename there the forcefield is located.
** @param 1 if no errors.
** @param 0 if any errors.
** @return
**/
local function clearMOEloadForceField [forceFieldFileName];
// Initialise parameters.
local result = [];
local errorCode = '';

// Clear out any molecular data, to ensure only one molecule is loaded.
Close ['force':1];

/*
* Loading the forcefield takes a very long time.
* Therefore try if the correct forcefield is already loaded.
*/
[result, errorCode] = task_call ['pot_Info', [], [errmsg:'ignore'] ];
if isEqual [errorCode, 'error'] then
        return 0;
endif;

// Check if result is a tagged vector.
if isTaggedVector [result] then
        // Check if same parameter-file.
        if result.filename == forceFieldFileName then
                return 1;
        endif;
endif;

// Not a tagged vector or wrong file-name, needs to load the forcefield.
[result, errorCode] = task_call ['pot_Load',
                        [forceFieldFileName], [errmsg:'ignore'] ];
if isEqual [errorCode, 'error'] then
        return 0;
endif;

// For doing LFMM-calculations electrostatics should be ignored.
[result, errorCode] = task_call ['pot_Setup', [eleEnable : 0],
        [errmsg:'ignore'] ];
// Check if any errors.
```

```
if isEqual [errorCode, 'error'] then
        return 0;
endif;

// No errors detected.
return 1;
endfunction;
```

### 9.2 A selection of tables and graphs for the conformational search of the 10 molecules also used for building scaffolds.

**Table 9.1:** The grouping of similar RMSD from lowspin conformational searches on the molecule with CCSD refcode DETTOL. The largest standard deviation comes after decreasing the 45x45 matrix of numbers into a 7x45 matrix there each row includes the average numbers collected from the corresponding group members.

| Name | Largest standard deviation | Group members |
|---|---|---|
| Group 1 | 0,0007 | 100; 400; 1000; 200_2_old |
| Group 2 | 0,0007 | 100_2_old; 100_3_new |
| Group 3 | 0,0002 | 500; 900; 1500; 3000; 3500; 5000; 400_2_new; 500_3_old; 500_2_new |
| Group 4 | 0,0003 | 300; 700; 300_3_old |
| Group 5 | 0,0004 | 200;600;800;2000;2500;4000;4500;100_2_new;100_3_new_tight;100_3_old; 200_2_new;200_3_new;200_3_new_tight;200_3_old;300_2_new; 300_2_old;300_3_new;300_3_new_tight;400_2_old;400_3_new; 400_3_old;400_3_new_tight;500_2_old;500_3_new;500_3_new_tight |
| Lowspin geometry optimized | N/A | |
| Original | N/A | |

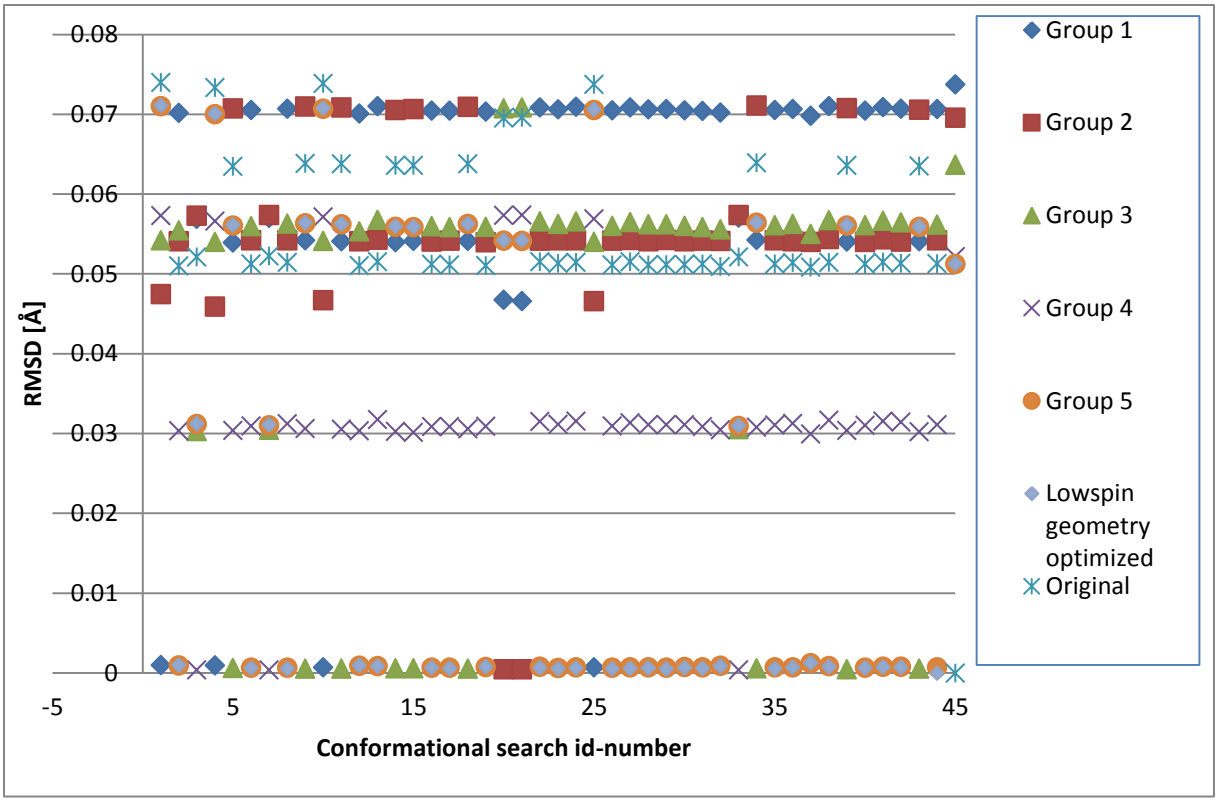


**Figure 9.1:** DETTOL lowspin

**Table 9.2: TIYPIC ls against hs...**

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -855.85 | -720.02 | -869.18 | -755.44 | -13.33 | -14.18 | 0.279 |
| 200 | -855.86 | -720.07 | -869.19 | -755.37 | -13.33 | -14.18 | 0.279 |
| 300 | -855.86 | -719.99 | -869.18 | -755.30 | -13.32 | -14.19 | 0.280 |
| 400 | -855.86 | -720.03 | -869.18 | -755.38 | -13.32 | -14.18 | 0.279 |
| 500 | -855.86 | -720.08 | -869.18 | -755.34 | -13.32 | -14.18 | 0.310 |
| 600 | -855.85 | -720.25 | -869.18 | -755.03 | -13.33 | -14.18 | 0.280 |
| 700 | -855.86 | -719.82 | -869.18 | -755.27 | -13.33 | -14.20 | 0.310 |
| 800 | -855.86 | -719.98 | -869.18 | -755.11 | -13.32 | -14.19 | 0.311 |
| 900 | -855.86 | -720.40 | -869.18 | -755.17 | -13.33 | -14.17 | 0.279 |
| 1000 | -855.86 | -720.15 | -869.19 | -755.14 | -13.33 | -14.18 | 0.311 |
| 1500 | -855.86 | -719.47 | -869.18 | -755.36 | -13.32 | -14.21 | 0.311 |
| 2000 | -855.86 | -720.22 | -869.19 | -755.18 | -13.33 | -14.18 | 0.311 |
| 2500 | -855.86 | -719.82 | -869.19 | -755.41 | -13.33 | -14.19 | 0.279 |
| 3000 | -855.86 | -719.98 | -869.18 | -755.44 | -13.32 | -14.18 | 0.310 |
| 3500 | -855.86 | -720.08 | -869.18 | -755.41 | -13.32 | -14.18 | 0.279 |
| 4000 | -855.86 | -719.93 | -869.18 | -755.30 | -13.32 | -14.19 | 0.279 |
| 4500 | -855.86 | -719.80 | -869.18 | -755.19 | -13.32 | -14.20 | 0.311 |
| 5000 | -855.86 | -719.87 | -869.18 | -755.11 | -13.32 | -14.20 | 0.280 |

**Table 9.3: PURYIK ...**

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -358.84 | -254.93 | -383.92 | -286.92 | -25.08 | -13.08 | 0.214 |
| 200 | -364.67 | -263.29 | -377.78 | -282.08 | -13.11 | -11.50 | 0.210 |
| 300 | -369.65 | -269.73 | -376.76 | -280.70 | -7.10 | -10.94 | 0.210 |
| 400 | -369.65 | -269.77 | -383.89 | -286.91 | -14.24 | -11.60 | 0.212 |
| 500 | -369.65 | -269.99 | -384.31 | -287.71 | -14.67 | -11.61 | 0.211 |
| 600 | -365.32 | -263.26 | -384.34 | -288.05 | -19.02 | -12.19 | 0.211 |
| 700 | -365.27 | -263.18 | -384.31 | -287.65 | -19.04 | -12.22 | 0.211 |
| 800 | -369.69 | -269.80 | -383.95 | -287.22 | -14.26 | -11.59 | 0.211 |
| 900 | -368.91 | -269.73 | -383.91 | -287.14 | -15.00 | -11.63 | 0.211 |
| 1000 | -370.39 | -270.40 | -383.91 | -287.16 | -13.52 | -11.52 | 0.211 |
| 1500 | -370.39 | -270.30 | -383.49 | -286.58 | -13.11 | -11.49 | 0.212 |
| 2000 | -370.39 | -270.24 | -383.88 | -287.30 | -13.49 | -11.52 | 0.211 |
| 2500 | -369.69 | -269.88 | -384.31 | -287.50 | -14.62 | -11.62 | 0.211 |
| 3000 | -370.39 | -270.43 | -383.95 | -287.04 | -13.56 | -11.53 | 0.212 |
| 3500 | -370.39 | -270.13 | -383.92 | -287.15 | -13.53 | -11.54 | 0.211 |
| 4000 | -370.36 | -270.41 | -383.95 | -287.22 | -13.59 | -11.52 | 0.211 |
| 4500 | -370.39 | -270.22 | -383.92 | -287.05 | -13.53 | -11.54 | 0.212 |
| 5000 | -369.69 | -269.97 | -384.34 | -287.81 | -14.64 | -11.61 | 0.211 |

**Table 9.4: PAZXAP ...**

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -125.25 | 202.64 | -164.31 | 60.37 | -39.06 | -31.93 | 0.621 |
| 200 | -121.28 | 127.42 | -164.30 | 60.19 | -43.02 | -27.92 | 0.486 |
| 300 | -125.25 | 202.14 | -164.31 | 60.49 | -39.06 | -31.90 | 0.621 |
| 400 | -121.68 | 126.76 | -164.30 | 60.44 | -42.62 | -27.88 | 0.486 |
| 500 | -125.25 | 203.05 | -164.31 | 60.51 | -39.06 | -31.96 | 0.622 |
| 600 | -125.25 | 202.08 | -164.31 | 60.44 | -39.06 | -31.90 | 0.621 |
| 700 | -121.84 | 127.16 | -164.31 | 60.45 | -42.47 | -27.89 | 0.659 |
| 800 | -125.25 | 202.23 | -164.31 | 60.59 | -39.06 | -31.91 | 0.622 |
| 900 | -121.02 | 127.45 | -164.30 | 60.21 | -43.29 | -27.93 | 0.659 |
| 1000 | -125.25 | 202.33 | -164.31 | 60.61 | -39.06 | -31.92 | 0.622 |
| 1500 | -125.25 | 202.67 | -164.30 | 60.39 | -39.06 | -31.93 | 0.621 |
| 2000 | -125.24 | 202.51 | -164.31 | 60.37 | -39.06 | -31.92 | 0.621 |
| 2500 | -125.25 | 202.62 | -164.31 | 60.59 | -39.06 | -31.93 | 0.621 |
| 3000 | -125.25 | 202.57 | -164.31 | 60.20 | -39.06 | -31.92 | 0.621 |
| 3500 | -121.22 | 127.22 | -164.31 | 60.15 | -43.09 | -27.91 | 0.659 |
| 4000 | -125.25 | 202.64 | -164.31 | 60.50 | -39.05 | -31.93 | 0.622 |
| 4500 | -125.25 | 202.48 | -164.31 | 60.43 | -39.05 | -31.92 | 0.621 |
| 5000 | -125.25 | 202.81 | -164.31 | 60.42 | -39.06 | -31.94 | 0.621 |

**Table 9.5: LOTSES**

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -307.84 | -155.67 | -319.56 | -177.16 | -11.73 | -16.32 | 0.362 |
| 200 | -307.84 | -155.28 | -319.57 | -176.69 | -11.73 | -16.37 | 0.518 |
| 300 | -307.84 | -155.57 | -319.57 | -176.76 | -11.73 | -16.35 | 0.363 |
| 400 | -307.83 | -155.78 | -319.57 | -176.28 | -11.74 | -16.36 | 0.520 |
| 500 | -307.84 | -155.62 | -319.57 | -176.43 | -11.73 | -16.36 | 0.365 |
| 600 | -307.84 | -155.49 | -319.56 | -176.28 | -11.72 | -16.38 | 0.520 |
| 700 | -307.84 | -155.33 | -319.57 | -176.77 | -11.74 | -16.36 | 0.364 |
| 800 | -307.84 | -155.55 | -319.57 | -176.40 | -11.72 | -16.37 | 0.519 |
| 900 | -307.84 | -155.42 | -319.57 | -176.18 | -11.73 | -16.38 | 0.365 |
| 1000 | -307.84 | -155.43 | -319.57 | -176.57 | -11.73 | -16.36 | 0.364 |
| 1500 | -307.84 | -155.57 | -319.57 | -176.88 | -11.73 | -16.34 | 0.363 |
| 2000 | -307.84 | -155.13 | -319.58 | -176.68 | -11.74 | -16.38 | 0.365 |
| 2500 | -307.84 | -155.12 | -319.57 | -176.87 | -11.74 | -16.37 | 0.518 |
| 3000 | -307.84 | -155.61 | -319.57 | -176.07 | -11.73 | -16.38 | 0.520 |
| 3500 | -307.84 | -155.84 | -319.57 | -176.40 | -11.73 | -16.35 | 0.364 |
| 4000 | -307.84 | -155.39 | -319.57 | -176.39 | -11.73 | -16.37 | 0.519 |
| 4500 | -307.84 | -155.43 | -319.57 | -176.77 | -11.73 | -16.35 | 0.519 |
| 5000 | -307.84 | -155.58 | -319.57 | -176.48 | -11.73 | -16.36 | 0.365 |

**Table 9.6: KIBWEZ...**

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -48.57 | -42.64 | -123.57 | -2.99 | -75.01 | -21.44 | 0.221 |
| 200 | -62.04 | -68.78 | -126.66 | 49.83 | -64.63 | -21.55 | 0.444 |
| 300 | -58.07 | -72.79 | -138.20 | -8.09 | -80.13 | -22.88 | 0.291 |
| 400 | -57.92 | -71.07 | -139.67 | -7.54 | -81.76 | -23.29 | 0.227 |
| 500 | -63.12 | -93.02 | -155.98 | -22.88 | -92.85 | -25.81 | 0.299 |
| 600 | -70.69 | -93.63 | -137.79 | 3.13 | -67.09 | -20.57 | 0.334 |
| 700 | -68.32 | -93.16 | -155.98 | -22.68 | -87.67 | -24.66 | 0.300 |
| 800 | -70.37 | -85.92 | -134.93 | -7.92 | -64.56 | -19.48 | 0.163 |
| 900 | -63.04 | -99.44 | -156.11 | -14.75 | -93.07 | -26.16 | 0.292 |
| 1000 | -63.04 | -99.76 | -155.96 | -22.65 | -92.92 | -25.88 | 0.253 |
| 1500 | -70.37 | -86.23 | -156.33 | -17.03 | -85.96 | -24.44 | 0.274 |
| 2000 | -70.69 | -93.27 | -156.34 | -17.07 | -85.65 | -24.40 | 0.300 |
| 2500 | -70.69 | -93.58 | -156.33 | -16.75 | -85.64 | -24.41 | 0.302 |
| 3000 | -70.69 | -94.08 | -158.22 | -17.99 | -87.53 | -24.84 | 0.274 |
| 3500 | -70.37 | -86.40 | -158.21 | -18.30 | -87.83 | -24.86 | 0.281 |
| 4000 | -70.69 | -93.57 | -158.22 | -18.08 | -87.53 | -24.83 | 0.273 |
| 4500 | -70.69 | -93.71 | -158.21 | -18.39 | -87.53 | -24.82 | 0.238 |
| 5000 | -70.69 | -93.52 | -158.21 | -18.19 | -87.52 | -24.83 | 0.273 |

**Table 9.7: DETTOL**

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -180.01 | -1.84 | -176.30 | -68.04 | 3.71 | -15.77 | 0.313 |
| 200 | -197.10 | -21.58 | -176.30 | -68.15 | 20.79 | -15.97 | 0.323 |
| 300 | -192.51 | -32.79 | -191.27 | -85.71 | 1.23 | -14.52 | 0.301 |
| 400 | -180.01 | -1.74 | -176.30 | -68.47 | 3.70 | -15.76 | 0.328 |
| 500 | -197.10 | -21.34 | -191.27 | -85.49 | 5.82 | -15.50 | 0.307 |
| 600 | -197.09 | -21.16 | -191.27 | -85.35 | 5.82 | -15.52 | 0.302 |
| 700 | -192.51 | -32.68 | -191.27 | -85.51 | 1.24 | -14.53 | 0.307 |
| 800 | -197.10 | -21.17 | -191.08 | -86.00 | 6.02 | -15.49 | 0.288 |
| 900 | -197.10 | -21.22 | -191.27 | -85.42 | 5.83 | -15.52 | 0.307 |
| 1000 | -180.01 | -1.24 | -191.27 | -85.84 | -11.26 | -16.23 | 0.308 |
| 1500 | -197.09 | -21.00 | -191.27 | -86.11 | 5.82 | -15.50 | 0.303 |
| 2000 | -197.09 | -21.23 | -191.27 | -85.93 | 5.82 | -15.49 | 0.301 |
| 2500 | -197.10 | -20.95 | -191.27 | -85.89 | 5.82 | -15.51 | 0.300 |
| 3000 | -197.09 | -20.95 | -191.27 | -85.86 | 5.82 | -15.52 | 0.307 |
| 3500 | -197.10 | -21.20 | -191.27 | -85.84 | 5.83 | -15.50 | 0.290 |
| 4000 | -197.10 | -21.48 | -191.27 | -85.74 | 5.83 | -15.49 | 0.301 |
| 4500 | -197.10 | -21.10 | -191.27 | -85.71 | 5.83 | -15.51 | 0.303 |
| 5000 | -197.10 | -20.96 | -191.27 | -85.46 | 5.83 | -15.53 | 0.301 |

**Table 9.8: IPIWAF**

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -263.17 | -192.82 | -325.32 | -116.33 | -62.15 | -23.33 | 0.542 |
| 200 | -263.17 | -192.88 | -325.33 | -116.29 | -62.15 | -23.33 | 0.539 |
| 300 | -263.17 | -192.27 | -325.32 | -116.34 | -62.15 | -23.34 | 0.540 |
| 400 | -263.18 | -192.47 | -325.33 | -116.27 | -62.15 | -23.34 | 0.540 |
| 500 | -263.18 | -192.85 | -325.32 | -116.31 | -62.15 | -23.33 | 0.541 |
| 600 | -263.17 | -192.76 | -325.32 | -116.34 | -62.15 | -23.33 | 0.543 |
| 700 | -263.18 | -192.47 | -325.33 | -116.26 | -62.15 | -23.34 | 0.541 |
| 800 | -263.18 | -192.37 | -325.33 | -116.16 | -62.15 | -23.35 | 0.542 |
| 900 | -263.17 | -192.44 | -325.33 | -116.16 | -62.16 | -23.35 | 0.540 |
| 1000 | -263.18 | -192.66 | -325.33 | -116.14 | -62.15 | -23.34 | 0.542 |
| 1500 | -263.17 | -192.39 | -325.33 | -116.18 | -62.16 | -23.35 | 0.540 |
| 2000 | -263.18 | -192.32 | -325.33 | -116.25 | -62.15 | -23.34 | 0.541 |
| 2500 | -263.17 | -192.47 | -325.33 | -116.27 | -62.15 | -23.34 | 0.541 |
| 3000 | -263.17 | -192.44 | -325.33 | -116.21 | -62.16 | -23.35 | 0.540 |
| 3500 | -263.18 | -192.80 | -325.33 | -116.09 | -62.15 | -23.34 | 0.541 |
| 4000 | -263.18 | -192.28 | -325.33 | -116.22 | -62.15 | -23.35 | 0.541 |
| 4500 | -263.18 | -192.33 | -325.33 | -116.22 | -62.15 | -23.35 | 0.541 |
| 5000 | -263.18 | -192.73 | -325.33 | -116.24 | -62.15 | -23.34 | 0.541 |

**Table 9.9: HUMBEX**

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -107.12 | 5.18 | -179.36 | -73.16 | -72.24 | -22.08 | 0.407 |
| 200 | -185.85 | -40.16 | -179.36 | -73.20 | 6.49 | -13.63 | 0.434 |
| 300 | -185.85 | -39.93 | -179.36 | -72.74 | 6.49 | -13.66 | 0.435 |
| 400 | -153.25 | 25.87 | -179.36 | -73.23 | -26.11 | -17.53 | 0.395 |
| 500 | -185.85 | -40.05 | -179.36 | -73.17 | 6.49 | -13.63 | 0.358 |
| 600 | -185.85 | -39.99 | -179.36 | -72.86 | 6.49 | -13.65 | 0.435 |
| 700 | -185.86 | -40.50 | -179.36 | -72.97 | 6.50 | -13.62 | 0.434 |
| 800 | -185.85 | -40.28 | -179.36 | -72.70 | 6.49 | -13.64 | 0.435 |
| 900 | -185.85 | -40.19 | -179.36 | -72.59 | 6.49 | -13.65 | 0.436 |
| 1000 | -185.85 | -40.34 | -179.36 | -73.26 | 6.49 | -13.61 | 0.357 |
| 1500 | -185.86 | -39.93 | -179.36 | -72.96 | 6.50 | -13.65 | 0.434 |
| 2000 | -185.85 | -40.36 | -179.36 | -73.15 | 6.49 | -13.62 | 0.358 |
| 2500 | -185.85 | -40.36 | -179.36 | -72.79 | 6.49 | -13.63 | 0.358 |
| 3000 | -185.85 | -40.23 | -179.36 | -72.67 | 6.49 | -13.64 | 0.435 |
| 3500 | -185.86 | -40.31 | -179.36 | -73.00 | 6.49 | -13.63 | 0.358 |
| 4000 | -185.85 | -40.12 | -179.36 | -72.80 | 6.49 | -13.65 | 0.435 |
| 4500 | -185.86 | -40.14 | -179.36 | -72.69 | 6.49 | -13.65 | 0.359 |
| 5000 | -185.86 | -39.98 | -179.36 | -72.75 | 6.49 | -13.66 | 0.435 |

**Table 9.10: FEBMAB**

| # conf. search | $U_{LS}$ [kJ/mol] | $U_{HS\_in\_LS\_conf.}$ [kJ/mol] | $U_{LS}$ [kJ/mol] | $U_{LS\_in\_HS\_conf.}$ [kJ/mol] | $U_{HS} - U_{LS}$ [kJ/mol] | FITNESS | LS against HS RMSD [Å] |
|---|---|---|---|---|---|---|---|
| 100 | -710.85 | -563.84 | -739.78 | -603.23 | -28.93 | -17.33 | 0.313 |
| 200 | -710.85 | -563.77 | -739.78 | -603.35 | -28.93 | -17.33 | 0.412 |
| 300 | -710.85 | -563.60 | -739.78 | -603.14 | -28.93 | -17.34 | 0.313 |
| 400 | -710.85 | -563.80 | -739.77 | -603.23 | -28.92 | -17.33 | 0.413 |
| 500 | -710.86 | -563.72 | -739.78 | -603.30 | -28.93 | -17.33 | 0.313 |
| 600 | -710.85 | -563.48 | -739.78 | -603.19 | -28.93 | -17.35 | 0.314 |
| 700 | -710.86 | -563.80 | -739.78 | -603.22 | -28.92 | -17.33 | 0.413 |
| 800 | -710.85 | -563.60 | -739.78 | -603.45 | -28.93 | -17.33 | 0.313 |
| 900 | -710.85 | -564.08 | -739.78 | -603.29 | -28.93 | -17.32 | 0.313 |
| 1000 | -710.85 | -563.84 | -739.78 | -603.40 | -28.94 | -17.32 | 0.412 |
| 1500 | -710.86 | -563.65 | -739.78 | -603.33 | -28.92 | -17.33 | 0.313 |
| 2000 | -710.85 | -563.48 | -739.78 | -603.51 | -28.93 | -17.33 | 0.313 |
| 2500 | -710.85 | -563.86 | -739.78 | -603.51 | -28.93 | -17.32 | 0.412 |
| 3000 | -710.86 | -563.81 | -739.78 | -603.53 | -28.93 | -17.32 | 0.314 |
| 3500 | -710.85 | -563.74 | -739.78 | -603.44 | -28.93 | -17.33 | 0.412 |
| 4000 | -710.85 | -563.57 | -739.78 | -603.44 | -28.93 | -17.33 | 0.314 |
| 4500 | -710.85 | -563.83 | -739.78 | -603.50 | -28.93 | -17.32 | 0.313 |
| 5000 | -710.85 | -563.51 | -739.78 | -603.28 | -28.93 | -17.34 | 0.314 |

## 9.3

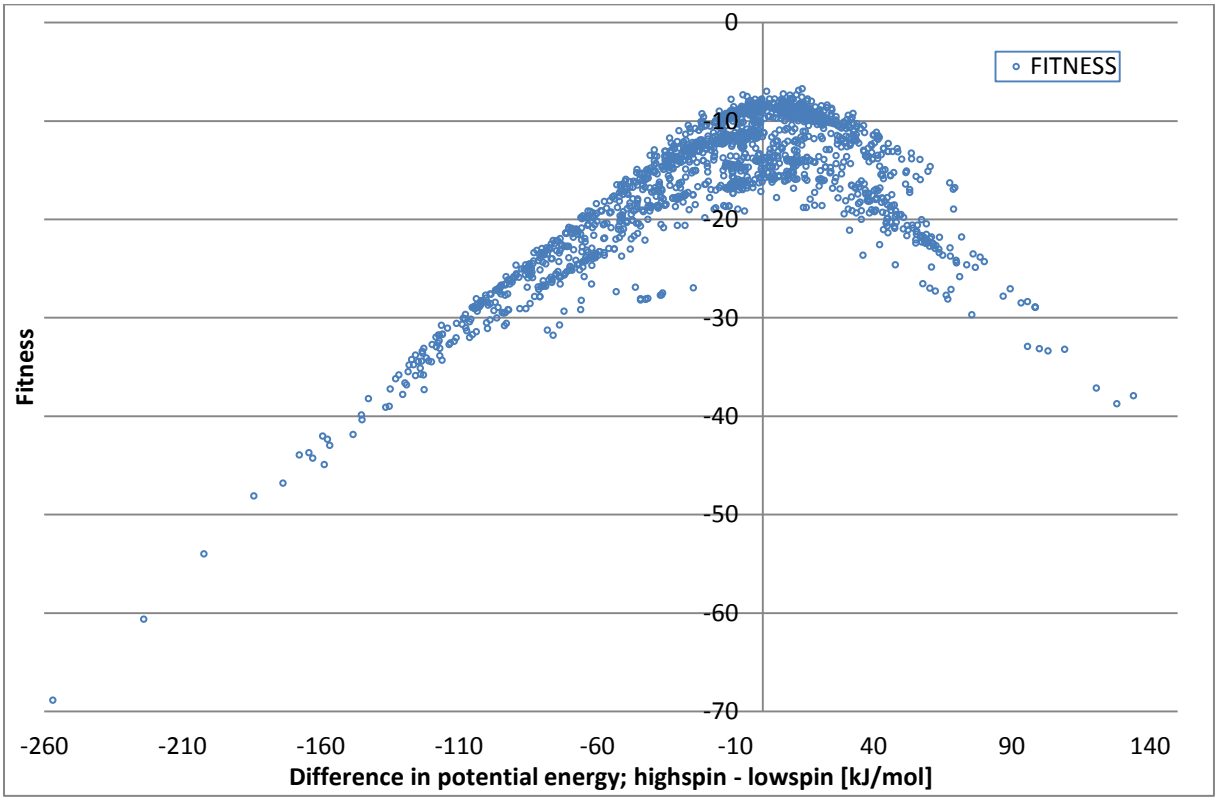


**Figure 9.2:** Plot showing the difference in potential energy by calculating $U_{HS} – U_{LS}$ against fitness in Denoptim run with internal id run_201. This plot is based on all the valid molecules.

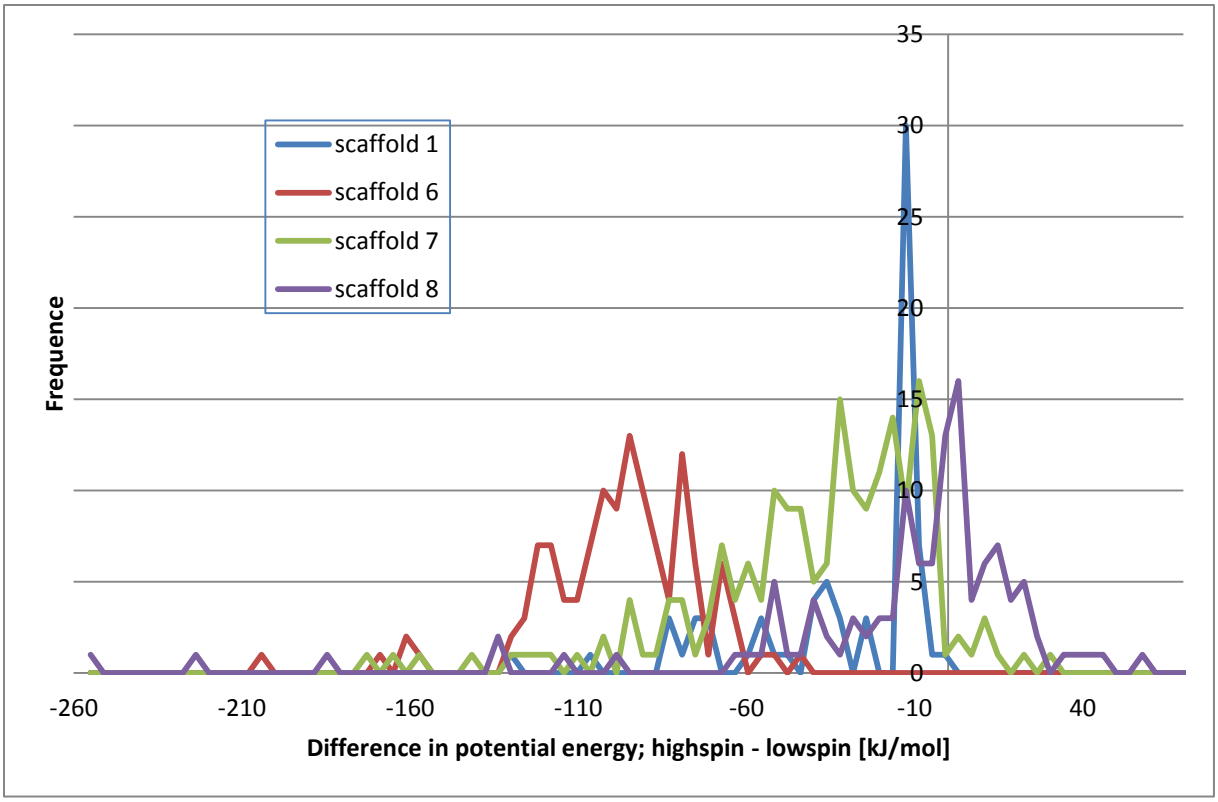


**Figure 9.3:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_201. The histogram includes all good molecules built on the basis of scaffolds 1, 6, 7 and 8.

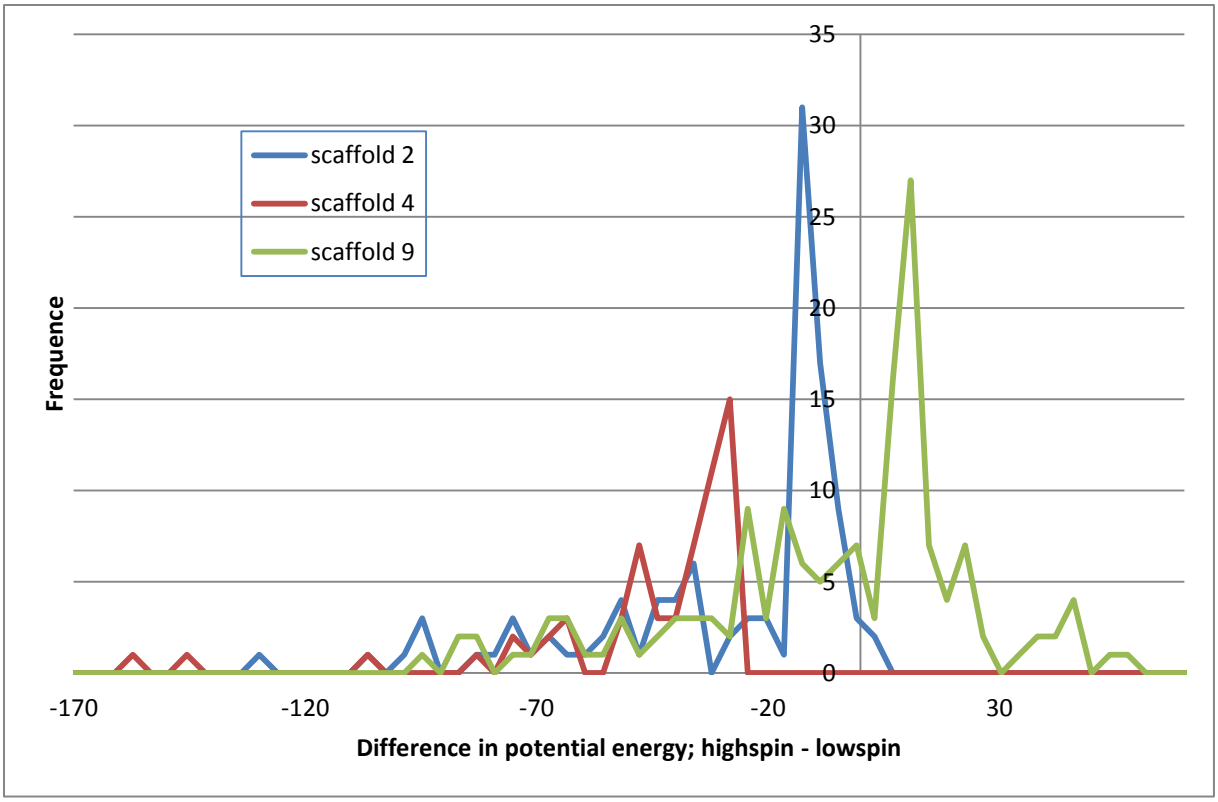


**Figure 9.4:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_201. The histogram includes all good molecules built on the basis of scaffolds 2, 4 and 9.

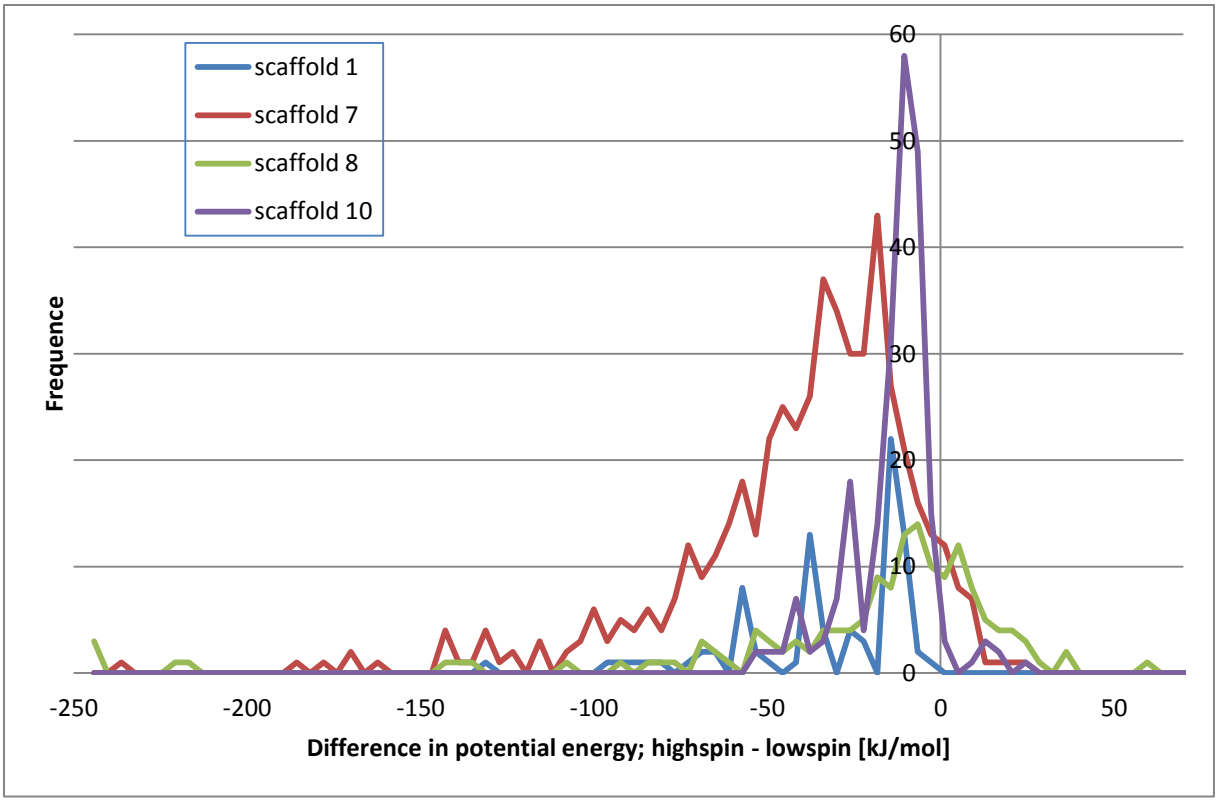


**Figure 9.5:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_310. The histogram includes all good molecules built on the basis of scaffolds 1, 7, 8 and 10.

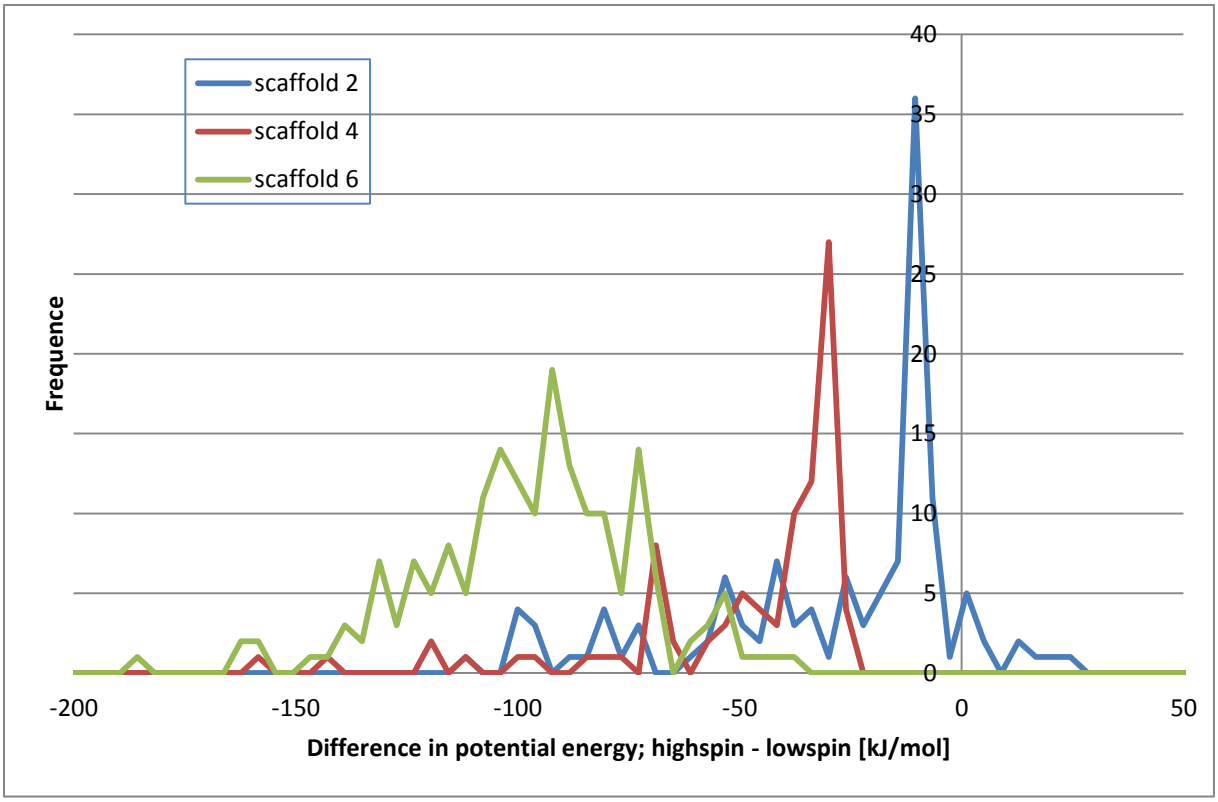


**Figure 9.6:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_310. The histogram includes all good molecules built on the basis of scaffolds 2, 4 and 6.

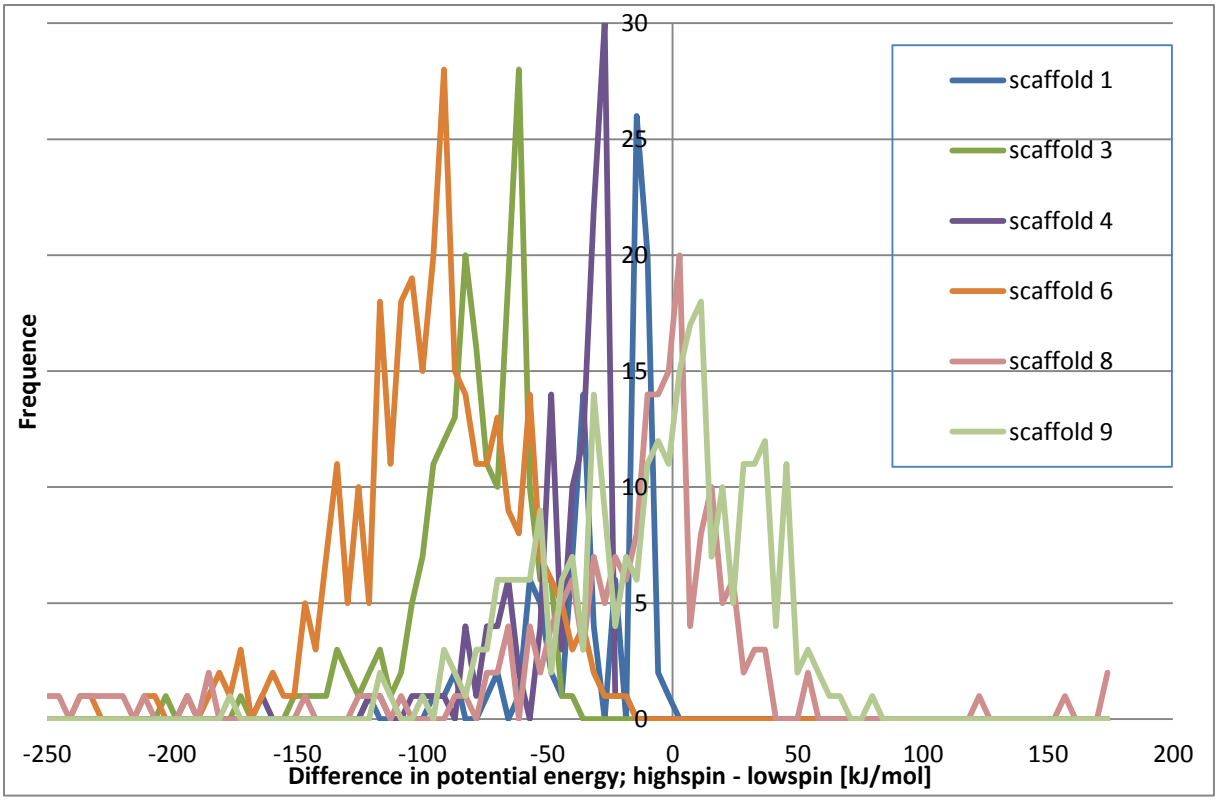


**Figure 9.7:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_325. The histogram includes all good molecules built on the basis of scaffolds 1, 3 and 4, 6, 8 and 9.

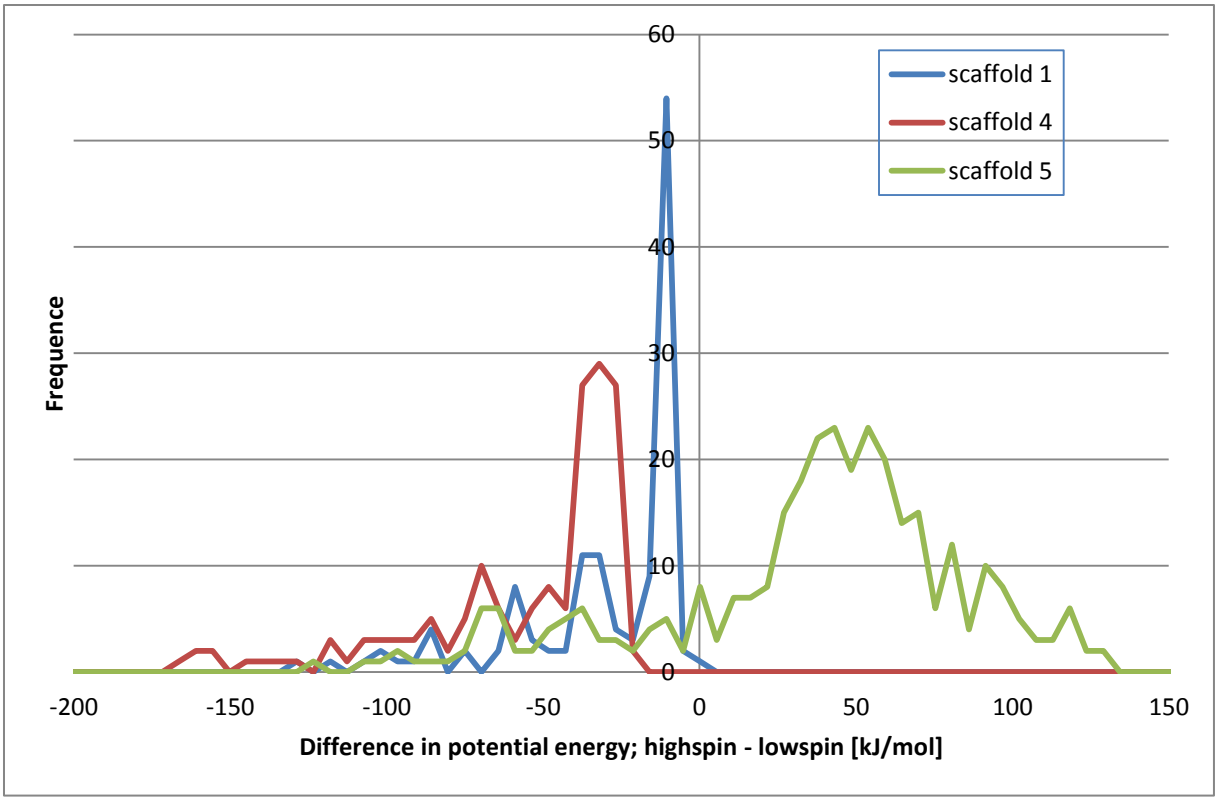


**Figure 9.8:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_323. The histogram includes all good molecules built on the basis of scaffolds 1,4 and 5.

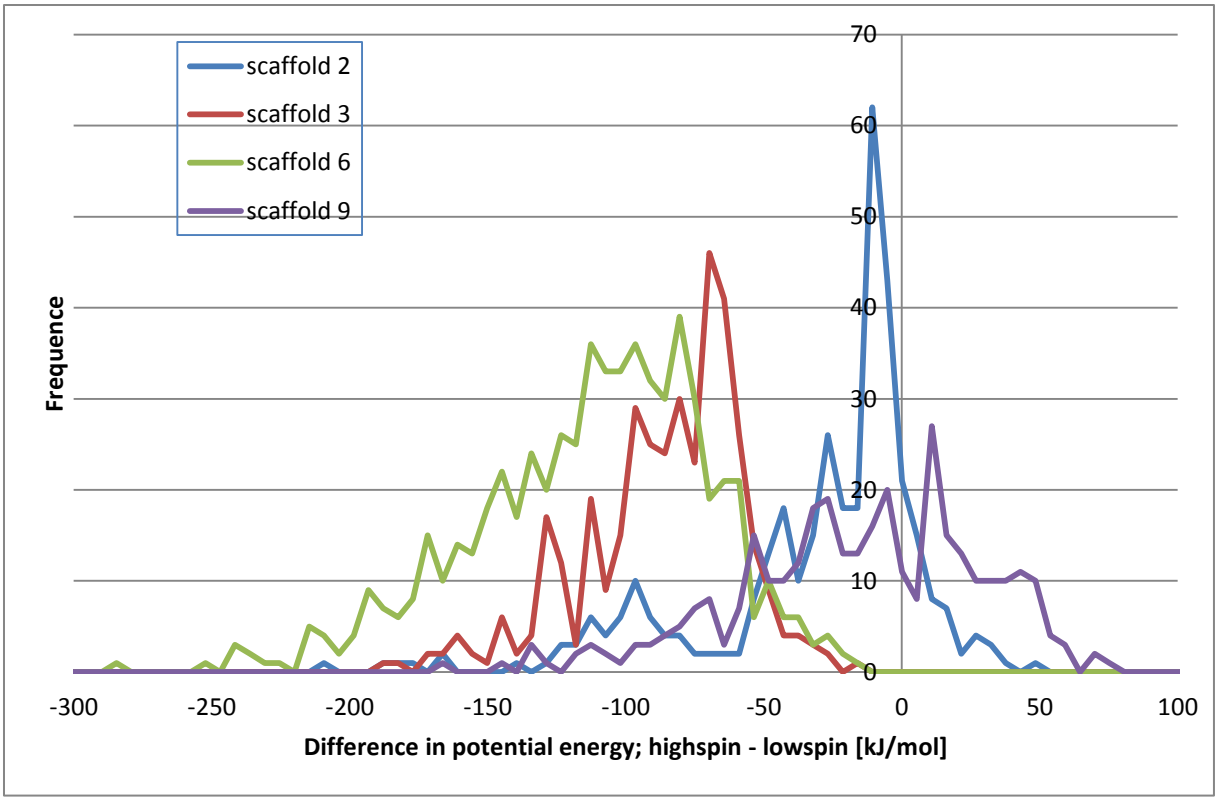


**Figure 9.9:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_323. The histogram includes all good molecules built on the basis of scaffolds 2,3, 6 and 9.

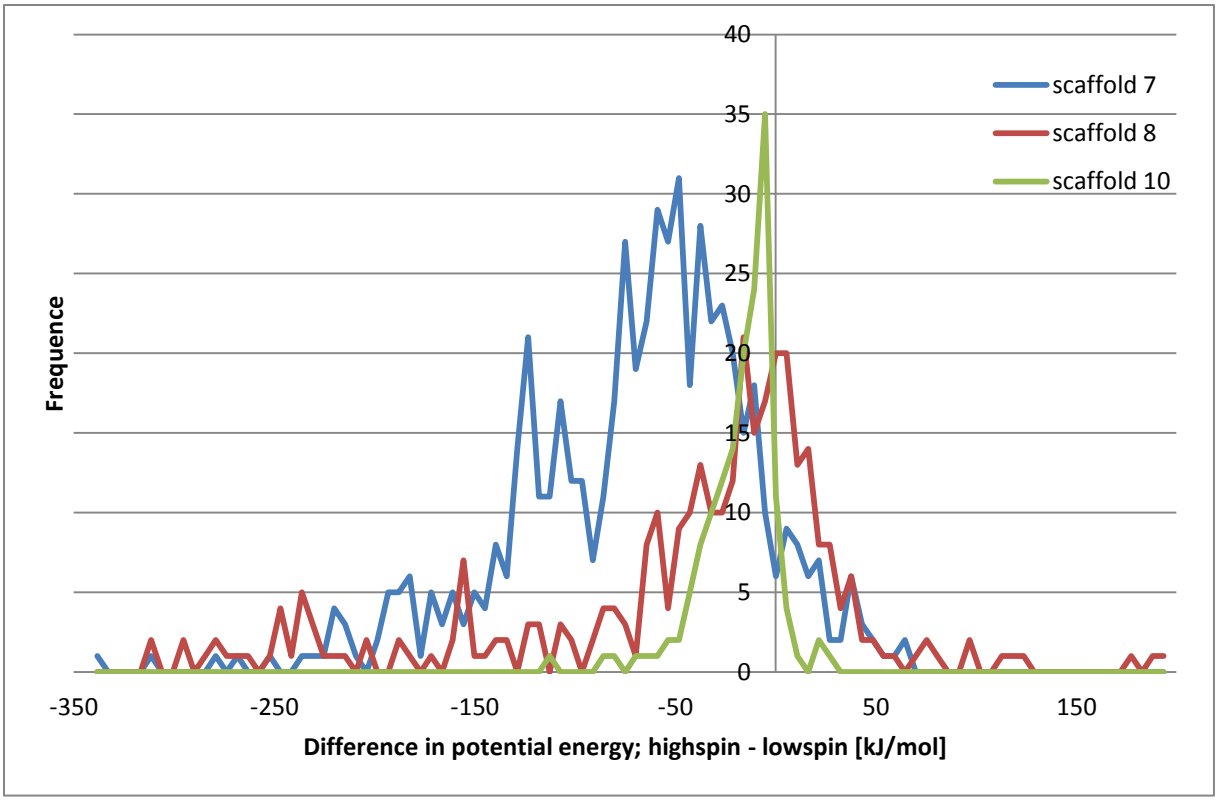


**Figure 9.10:** Histogram showing the distribution against the difference in potential energy $U_{HS}$ - $U_{LS}$ for the Denoptim run with internal id run_323. The histogram includes all good molecules built on the basis of scaffolds 7, 8 and 10.