

EBERHARD-KARLS-UNIVERSITÄT TÜBINGEN
Wilhelm-Schickard-Institut für Informatik
Lehrstuhl Rechnerarchitektur

Diplomarbeit

**Active Structure Learning using Genetic
Algorithms and Kernel Functions**

Christoph Schwörer

Betreuer: Prof. Dr. rer. nat. Andreas Zell
Wilhelm-Schickard-Institut für Informatik

Prof. Dr. rer. nat. Karl-Heinz Wiesmüller
EMC Microcollections GmbH

Begonnen am: 13th January 2010

Beendet am: 12th July 2010

Erklärung

Hiermit versichere ich, diese Arbeit selbstständig verfasst und nur die angegebenen Quellen benutzt zu haben.

Tübingen am 12th July 2010

Christoph Schwörer

Kurzfassung.

Current 3D QSAR approaches attempt to build models base not only on 1D or 2D descriptors of molecules like weight, charge or molecular graphs, but also on 3D sensitive information like the conformation or information about the molecules surface. A basic assumption on building these 3D QSAR models is that the best results are attained by using the best available (i.e., the obtained active structure) data. In this work I tried to find such a best achievable 3D QSAR model by the means of optimizing a model over a set of conformations using a genetic algorithm and three different kernel methods. The intent was to see if these resulting models would include the active structures. For the generation of the sets of conformations I used two different approaches. The first being a precomputation of the conformations the second an implicit generation concurrent to the optimization. The results will show that the model with the best generalization and prediction accuracy in most cases do not include the active conformation but conformations with a minimal average pairwise distance to all other possible conformations of the respective molecules.

Contents

1	Introduction	1
2	Background Information	4
2.1	Kernel Functions	4
2.2	Support Vector Regression	4
2.3	Rotation with quaternions	7
2.4	RMSD calculation with quaternions	7
2.5	Genetic Algorithm	9
2.6	Quantitative Structure-Activity Relationship	11
3	Materials and Methods	13
3.1	Overall process	13
3.2	Radial Distribution Function	15
3.3	Kernel	17
3.3.1	Probability Product Kernel	17
3.3.2	Radial Basis Function Kernel	18
3.3.3	Atom Pair Kernel	19
3.4	Dataset	21
3.5	Conformation Sampling	21
3.5.1	Precomputed Conformation Sampling	21
3.5.2	Implicit Conformation Sampling	24
3.6	SVR	27
4	Results	28
4.1	Precomputed Conformation Sampling	28
4.1.1	Initial Runs	31
4.1.2	Reduced Dataset with PPK and RBF Kernel	33
4.1.3	Reduced Dataset with Atom Pair Kernel	35
4.1.4	Alternative Parameters for the Product Probability Kernel	37
4.1.5	Alternative Parameters for APK	39
4.1.6	Increased Mutation Rate	41
4.1.7	Alternative Conformation Sampling	42
4.1.8	Alternative Mutation Operator	45
4.1.9	Reruns	47
4.2	Implicit Conformation Sampling	47
4.2.1	Initial Runs	49
4.2.2	Reduced Dataset and Fixed Conformation	51
5	Discussion	52
6	Prospects	54

Bibliography	55
---------------------	-----------

1 Introduction

In drug design one of the major goals is to find new lead structures. Lead structures already show a certain affinity towards the intended target but express unwanted side effects or lack certain properties. For example they may be toxic or have a low bioavailability. Without a detailed understanding of the biochemical processes responsible for the activity the search for such a new lead structure is non-trivial.

The usual process is to simply try a huge combination of different chemical compounds *in vitro* and observe their activity. But the combinatorial possibilities of this strategy can explode even for small systems. For instance the number of compounds needed to place 10 substituents on the four open positions of an asymmetrically disubstituted benzene ring system is approximately 10,000.

Therefore this classical *screening* process was automatized and combinatorially optimized in the last decades to *high throughput screening* (HTS) which allowed for a systematical search in greater databanks with hundreds of thousands of entries. But still this process makes up a large amount of the development-costs and -time. Further the chemical compounds needed for the synthesis are often rare and hard to come by in the purity needed for reliable results.

One way to optimize this exhaustive search and give an indication of the right direction is to develop a model that quantitatively relates variations in biological activity to changes in molecular properties which can be easily obtained for each compound. One of the first to build such a model was Corvin Hansch correlating lipophilicity and polarity with biological activity in his Hansch method [Han69]. But there exist many other approaches to this *Quantitative Structure-Activity Relationship* (QSAR) principle, which mostly differ in their use of molecular descriptors and mathematic models such as *Partial Least Squares* or *Principal Component Analysis*. The QSAR models developed in this work are based on kernels which are evaluated by *Support Vector Regression*.

In the recent years several models have been developed using 3D descriptors of molecules. These 3D descriptors are important, because to build a model and gain understanding for the binding process it is not enough to know of the single component and values of a molecule but also to know their 3D dimensional arrangement. As one can see for example on fig 1.1 where a single molecule can take on several conformations. To know which of these conformation is the active conformation can improve the modeling process and the understanding of the chemical processes leading to the activity.

One thing all QSAR methods have in common is the basic assumption that the biological activity is an additive function of the molecular properties (2D or 3D) of the substituents and groups of the respective structure. Not only the mere presence of those groups is essential but also their three dimensional arrangement.

This leads to the expectation that on using 3D descriptors only good and correct training data including 3D information of the molecules leads to a good model of the activity. But, what if this doesn't hold true? What if a better model can be created *not* using the actually correct physical data? The question is if the reverse of this expectation is always valid, thus if the model quality is bijective to the training data quality.

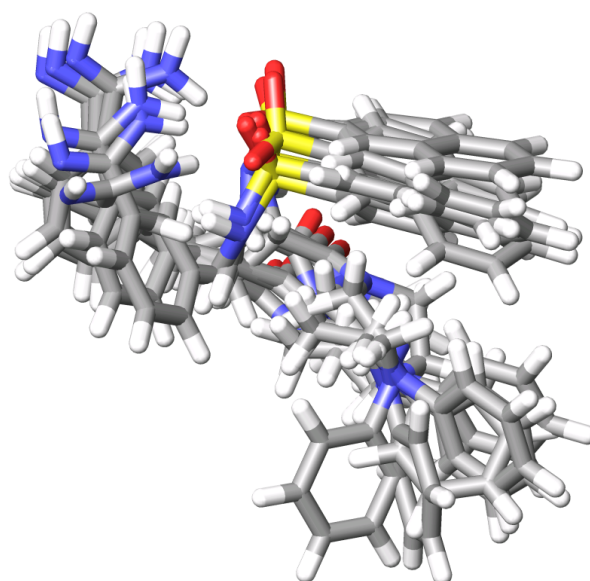
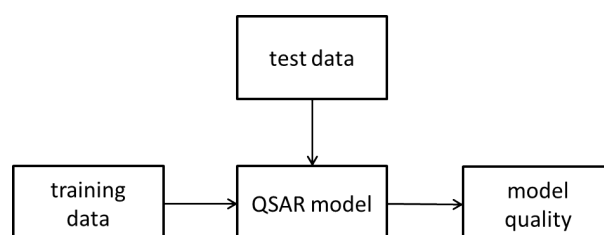
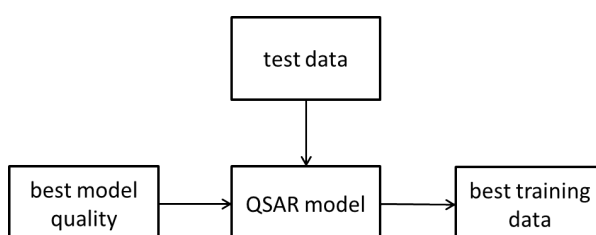


Figure 1.1: This figure shows an overlay of six conformations of the same thrombin inhibitor. One can see the high flexibility of the lower ring system.



(a) This figure shows the standard process for building a QSAR model.



(b) This figure shows the reverse process of optimizing a QSAR model with respect to the training data quality

Figure 1.2: These figures show the standart and the process used in this work to build a QSAR model. To optimize the model quality with respect to the varying training data the process has been reversed.

To test, whether one can find such a model I will reverse the QSAR approach (see figure 1.2 . Therefore optimizing the QSAR model to predict activity by the means of altering the training dataset, where for each data point several values are given including the actual physical ones. While the input data points vary (i.e. their molecular descriptors) their target function value

(i.e. the activity) stays the same.

The intention of this experiment is to see if the best attainable model includes the actual active structure or the artificially created one. To this end I compiled a data set and created a set of conformers for each molecule. Then I concurrently optimized the activity prediction over the whole training dataset not to favor one molecule over the other by successively optimizing one after the other. A good way to handle multidimensional optimization with several datasets is to use a *genetic algorithm* which I did in this case.

In this work I will show the methods used for the generation of the the dataset, the optimization and the evaluation. Further I will present the results and discuss their significance. Finally I will give a perspective of further work which can be done on this topic.

2 Background Information

2.1 Kernel Functions

Detecting linear relations has been the focus of much research in statistics and machine learning for the last few decades and the resulting algorithms are well understood, well developed and efficient [Sew07]. However, many models of natural processes aren't linear. So, if a problem is non-linear, instead of trying to fit a non-linear model, one can map the problem from the *inputspace* \mathcal{X} to a new higher-dimensional space called the *featurespace* \mathcal{F} and then use a linear model in the feature space. This mapping can be achieved by doing a non-linear transformation. For example the function ϕ can be given as

$$\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \text{ with } \phi(x_1, x_2) = (x_1^2, \sqrt{2x_1x_2}, x_2^2) \quad (2.1)$$

While this function is a very simple one, other functions can easily become computationally impracticable for both polynomial features and higher dimensionality. This is grounded on the fact that the number of different monomial features of degree p is $\binom{d+p-1}{p}$, with $d = \dim \mathcal{X}$ [Vap95] (e.g. $p = 7$, $d = 28 \cdot 28 = 784$, corresponds to a total of approximately $3,7 \cdot 10^{16}$ features).

The key to an efficient computation is the observation made by [BGV92] that

$$\left\langle \left(x_1^2, \sqrt{2x_1x_2}, x_2^2 \right), \left(x_1'^2, \sqrt{2x_1'x_2'}, x_2'^2 \right) \right\rangle = \langle x, x' \rangle^2 \quad (2.2)$$

which allows the use of *kernel functions* where ϕ must not be explicitly known as long as the function corresponds to a dot product in the *Featurespace* \mathcal{F}

$$k(x, x') := \langle \phi(x), \phi(x') \rangle \quad (2.3)$$

2.2 Support Vector Regression

Many multi variant systems assume that there is a linear relation between X and Y which holds for all samples. In chemoinformatics this assumption does not hold true and causes a variety of problems on the prediction of unknown data points. One way to solve these occurring problems is to use non-linear learning methods such as *support vector regression* (SVR). The support vector algorithm is a non-linear generalization of the *Generalized Portrait* algorithm developed in Russia in the 1960's [VL63] [VC64]. Its groundwork, the statistical learning theory, or *VC theory*, has been developed over the last half century by Vapnik and Chervonenkis [VC74] [Vap82] [Vap95]. The VC theory defines properties of learning machines, enabling them to generalize to unseen data.

Given a set of training data $\{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathcal{X} \times \mathbb{R}$ with \mathcal{X} denoting the space of input patterns (e.g. $\mathcal{X} = \mathbb{R}^d$) the goal of ε -SV regression is to find a function $f(x)$ with a maximum

deviation of ε from the actually received targets y_i for all the training data. In addition f should be as flat as possible. The form of a linear function f is given as

$$f(x) = \langle w, x \rangle + b \quad \text{with } w \in \mathcal{X}, b \in \mathbb{R} \quad (2.4)$$

with $\langle \cdot, \cdot \rangle$ denoting the dot product in \mathcal{X} and flatness meaning a small w . To attain this we minimize the euclidean norm $\frac{1}{2} \|w\|^2$ which can be formally written as a convex optimization problem:

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|w\|^2 \\ & \text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{cases} \end{aligned} \quad (2.5)$$

The above formula is viable for all problems where a function f actually exists that approximates all pairs (x_i, y_i) with precision ε . If this is not the case, or if we want to allow some errors, according to [CV95] one can introduce slack variables ξ_i, ξ_i^* leading to the formula:

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (2.6)$$

Where the constant $C > 0$ defines the trade off between the flatness of f and the amount up to which deviations larger than ε are tolerated. This is the same as dealing with a ε -intensive loss function $|\xi|_\varepsilon$ denoted by:

$$|\xi|_\varepsilon := \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{else} \end{cases} \quad (2.7)$$

Figure 2.1 depicts the use of ξ and ε . Extending support vector machines to solve non linear problems is possible by using a standard dualization approach utilizing Lagrange multipliers as described in [Fle89] leading to the following formula:

$$\begin{aligned} L := & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n (\eta_i \xi_i + \eta_i^* \xi_i^*) \\ & - \sum_{i=1}^n \alpha_i (\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b) \\ & - \sum_{i=1}^n \alpha_i^* (\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle - b) \end{aligned} \quad (2.8)$$

With L being the Lagrangian and $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ the Lagrangian multipliers. Thus they have to satisfy the constraints

$$\eta_i, \eta_i^*, \alpha_i, \alpha_i^* \geq 0 \quad (2.9)$$

To gain an optimal result one can infer from the saddle point condition that the partial deriva-

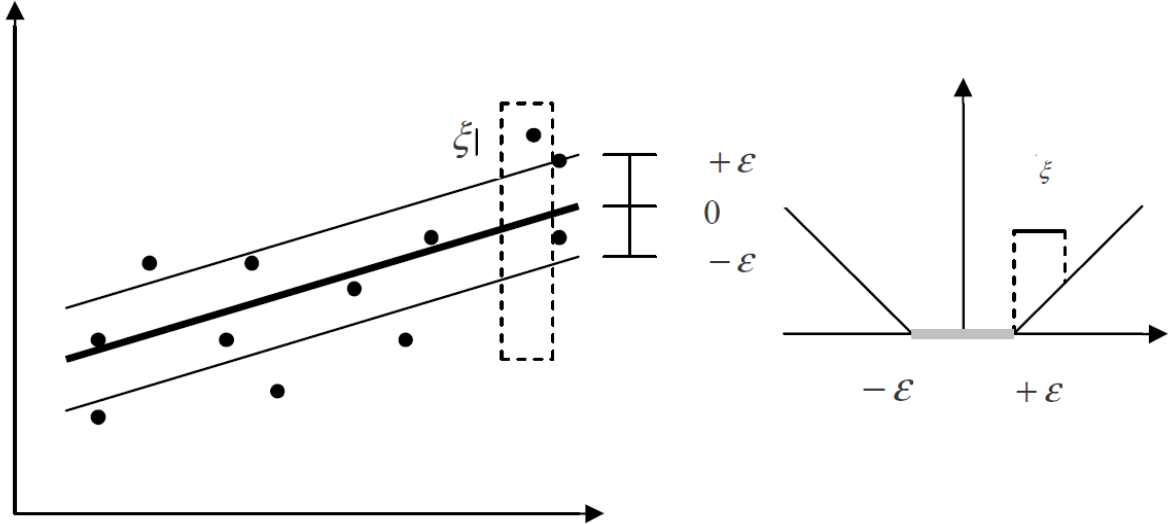


Figure 2.1: The image shows the use of ξ and ε in a support vector regression. Data points with a distance smaller than ε are not considered an error. For data points with a distance larger than ε the parameter ξ decides whether they are tolerated or not.

tives of L have to vanish

$$\begin{aligned}
 \frac{\partial L}{\partial b} &= \sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0 \\
 \frac{\partial L}{\partial w} &= w - \sum_{i=1}^n (\alpha_i - \alpha_i^*) x_i = 0 \\
 \frac{\partial L}{\partial \xi_i} &= C - \alpha_i - \eta_i = 0 \\
 \frac{\partial L}{\partial \xi_i^*} &= C - \alpha_i^* - \eta_i^* = 0
 \end{aligned} \tag{2.10}$$

Substituting eq 2.7 into eq. 2.6 leads to the dual optimization problem:

$$\begin{aligned}
 &\text{maximize} \quad \begin{cases} -\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle \\ -\varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \end{cases} \\
 &\text{subject to} \quad \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]
 \end{aligned} \tag{2.11}$$

Having already eliminated η_i, η_i^* we can further reformulate (7) to $\eta_i^{(*)} = C - \alpha_i^{(*)}$ so that follows

$$w = \sum_{i=1}^n (\alpha_i - \alpha_i^*) x_i, \text{ thus } f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b. \tag{2.12}$$

The fact that the data x_i only contributes in form of the dot product allows the introduction of kernel functions in such a way that

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{2.13}$$

This allows the prediction of unknown data points via

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) k(x, x_i) + b \tag{2.14}$$

2.3 Rotation with quaternions

Quaternions are an extension of the complex numbers invented by William Rowan Hamilton in 1843[Ham66] and formally introduced to computer graphics by the publication of Shoemaker [Sho85] [Har94]

Quaternions encode rotations by a set of 4 real numbers (or 2 complex numbers), while a linear representation of a rotation requires a 3×3 Matrix, thus 9 numbers. Further Quaternions occupy a smooth, seamless isotropic space which is the generalization of the surface of a sphere. This means that one doesn't need to take special care in avoiding singularities (e.g., the gimbal lock, where two rotation axes collapse into one making the interpolation irreversible).

The four-dimensional space \mathbb{H} is spanned by the real axis and three additional orthogonal axes, spanned by the vectors \mathbf{i} , \mathbf{j} , \mathbf{k} called the *principal imaginaries*, which obey Hamilton's rule

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \quad (2.15)$$

Where the three dimensional vectors $\mathbf{i}, \mathbf{j}, \mathbf{k}$ signify

$$\begin{aligned} \mathbf{i} &= (1,0,0) \\ \mathbf{j} &= (0,1,0) \\ \mathbf{k} &= (0,0,1). \end{aligned} \quad (2.16)$$

A quaternion $q = r + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ consists of a real part r and a pure part $x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ and can be written as a three dimensional vector and a scalar

$$q = (a, \mathbf{b}) \quad (2.17)$$

The sum of two quaternions is given as

$$q_1 + q_2 = (a_1 + a_2) + (\mathbf{v}_1 + \mathbf{v}_2) \quad (2.18)$$

and their product as

$$q_1 q_2 = a_1 a_2 - \mathbf{b}_1 \cdot \mathbf{b}_2 + a_1 \mathbf{b}_2 + a_2 \mathbf{b}_1 + \mathbf{b}_1 \times \mathbf{b}_2 \quad (2.19)$$

where the multiplication of two quaternions $q_1 q_2$ with unit length (i.e. absolute value = 1) and q_2 being a *pure quaternion* (i.e. with $a = 0$) causes a rotation of \mathbf{b}_2 around the axis described by \mathbf{b}_1 for $\cos^{-1} 2\phi$ degrees. Where ϕ is the desired rotation angle.

2.4 RMSD calculation with quaternions

In various cheminformatic situations the problem arises of finding the best superposition of one rigid object onto another. For example to give a similarity measure for two proteins or in case of this work two conformations of the same molecule. One method is finding the best rotation and translation to minimize the *root mean square deviation* (RMSD) [Kab76] with examples are given by [Dia76] and [McL72]. A prerequisite for this method is a given assignment of the points matched on each other. Usually such an assignment is already given (e.g., the canonical atom numbering of two different conformations).

The mathematical problem can be stated as follows: [Cou04]

“given a ordered set of vetors \mathbf{y}_k (target) and a second set \mathbf{x}_k (model), $1 \leq k \leq N$, find a orthogonal transformation \mathcal{U} and a translation \mathbf{r} such that the residual E (weighted by w_k)

$$E := \frac{1}{N} \sum_{k=1}^N w_k |\mathcal{U} \mathbf{x}_k + \mathbf{r} - \mathbf{y}_k|^2 \quad (2.20)$$

is minimized. ”Where the weight factor w_k allows to lay the emphasis on certain parts of the structure in question.

While Kabsch’s method uses Lagrange multipliers, Mackay proposed a method in 1984 [Mac84] using quaternions to calculate the rotation matrix. One disadvantage of Mackay’s method was that, using a linear form of the least square errors, the results could be false where objects had different relative orientations in space. In 1989 Kearsley developed a method, solving the non-linear least square error problem with an eigenvalue determination through the use of quaternions [Kea89]. The proof that both, Kabschs and Kearsleys methods lead to the same result was brought by Coutsiass *et al.* in 2005 [Cou05].

If \mathbf{x}_k and \mathbf{y}_k are considered as *pure quaternions*, with $x_k := (0, \mathbf{x}_k)$ and $x_k^c = -x_k$ the rotation $\mathcal{U}(q)$ can be written as

$$(0, \mathcal{U}(q)\mathbf{x}_k) = qx_kq^c \quad (2.21)$$

And the residual function is transformed using quaternions to

$$E_q = \frac{1}{N} \sum_{k=1}^N (qx_kq^c - y_k)(qx_kq^c - y_k)^c \quad (2.22)$$

An expansion and a multiplication by N leads to

$$\begin{aligned} NE_q &= \sum_{k=1}^N (qx_kq^c)(qx_kq^c)^c + y_k y_k^c (qx_kq^c) y_k^c - y_k (qx_kq^c)^c \\ &= \sum_{k=1}^N (x_k x_k^c + y_k y_k^c + (qx_kq^c)y_k + y_k(qx_kq^c)) \end{aligned} \quad (2.23)$$

where the normalization $qq^c = 1$ and the property of pure quaternions $x^c = -x$ has been used. qx_kq^c and y_k being pure quaternions and with a, b pure $ab + ba = 2(-\mathbf{a} \cdot \mathbf{b}, \mathbf{0}) = 2([ab]_0, \mathbf{0})$ the last two terms in eq. 2.23 can be combined as follows

$$(qx_kq^c)y_k + y_k(qx_kq^c) = 2([y_k(qx_kq^c)]_0, \mathbf{0}) \quad (2.24)$$

This means that only the 0th component is non-zero. Because of the associativity of the quaternions one can write $y_k(qx_kq^c) = (y_kqx_k)q^c$ and define $x_k := y_kqx_k$ wich leads to the 4-vector form of z_k, \mathcal{Z}_k with $\mathcal{Z}_k = \mathcal{A}_L(y_k)\mathcal{A}_R(x_k)\mathcal{Q}$ with $\mathcal{A}_L, \mathcal{A}_R$ defined as follows

$$\mathcal{A}_R(p) = \begin{pmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & p_3 & -p_2 \\ p_2 & -p_3 & p_0 & p_1 \\ p_3 & p_2 & -p_1 & p_0 \end{pmatrix}, \quad \mathcal{A}_L(p) = \begin{pmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{pmatrix} \quad (2.25)$$

All together we can write

$$\begin{aligned}
-2\mathbf{y}_k^T \mathcal{U}(q) \mathbf{x}_k &= 2[y_k(qx_kq^c)_0] \\
&= 2[z_kq^c]_0 \\
&= 2(z_{k0}q_0 + \mathbf{z}_k \cdot \mathbf{q}) \\
&= 2\mathcal{Q}^T \mathcal{Z}_k \\
&= 2\mathcal{Z}^T \mathcal{A}_l(y_k) \mathcal{R}(x_k) \mathcal{Q}
\end{aligned} \tag{2.26}$$

followed by the residue

$$NE_q = \sum_{k=1}^N (|\mathbf{x}_k|^2 + |\mathbf{y}_k|^2) - 2\mathcal{Q}^T \mathcal{F} \mathcal{Q} \tag{2.27}$$

with

$$\mathcal{F} := - \sum_{k=1}^N \mathcal{A}_L(y_k) \mathcal{A}_R(x_k) \tag{2.28}$$

leading to the full form of the matrix \mathcal{F} in terms of the correlation matrix \mathcal{R}

$$\mathcal{F} = \begin{pmatrix} R_{11} + R_{22} + R_{33} & R_{23} - R_{32} & R_{31} - R_{13} & R_{12} - R_{21} \\ R_{23} - R_{32} & R_{11} - R_{22} - R_{33} & R_{12} + R_{21} & R_{13} + R_{31} \\ R_{31} - R_{13} & R_{12} + R_{21} & -R_{11} + R_{22} - R_{33} & R_{23} + R_{32} \\ R_{12} - R_{21} & R_{13} + R_{31} & R_{23} + R_{32} & -R_{11} - R_{22} + R_{33} \end{pmatrix} \tag{2.29}$$

In this way the problem can be reduced to finding the extreme of a quadratic form $\mathcal{Q}^T \mathcal{F} \mathcal{Q}$ for the four variables $q_i, i \in \{0, 1, 2, 3\}$ subject to the constraint $\mathcal{Q}^T \mathcal{Q} = 1$. Here $\mathcal{Q}^T \mathcal{F} \mathcal{Q}$ is the standard Rayleigh quotient for a symmetric matrix \mathcal{F} , where the maximum value of $\mathcal{Q}^T \mathcal{F} \mathcal{Q}$ is equal to its largest eigenvalue which leads to the following problem

$$\mathcal{F} \mathcal{Q} = \lambda \mathcal{Q} \tag{2.30}$$

which in turn leads to the following expression for the best RMSD Value

$$e_q = \sqrt{\min_{\|\mathbf{q}\|=1} E_q} = \sqrt{\frac{\sum_{k=1}^N (|\mathbf{x}_k|^2 + |\mathbf{y}_k|^2) - 2\lambda_{max}}{N}} \tag{2.31}$$

2.5 Genetic Algorithm

In cheminformatics one often encounters optimization problems with several variable parameters. Traditional optimization methods such as steepest decent often fail at this task because they often run into a local optimum. To get around this problem Prof. John Holland developed the class of *Genetic Algorithms* (GA's) at the University of Michigan during the 60's and 70's [Hol75].

Genetic algorithms belong to the class of stochastic search methods. Their distinctive feature is, that instead of operating on a single solution like most other stochastic search methods, they operate on a whole set of solutions. The term *Genetic Algorithm* is a tribute to their basic operations which derive from natural evolutionary processes, such as inheritance, mutation, selection, and crossover.

Given a problem P with parameters x_1, \dots, x_n the first step is to initialize a first set of solutions, called population $M(0)$. Each single solution is called individual m and is represented by a bit string called chromosome (see fig x). The initial value of each parameter is chosen at random within its predefined range.

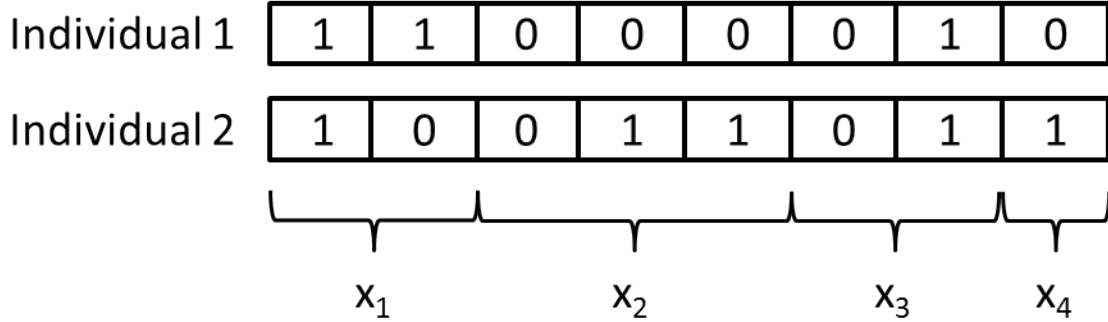


Figure 2.2: This figure shows two individuals with parameters x_1, \dots, x_4 encoded as a series of binary representations of different length.

The second step is to evaluate each individual (i.e. solution) in the current population $M(t)$ for its fitness. This is done by applying the individuals parameter values to a fitness function (which in most cases is the initial problem function) and assigning the function result as fitness value $u(m)$. This means that the parameters of individuals with higher fitness values lead to a better result of the problem function.

The third step is to assign each of the current individuals a selection probability $p(m)$ which depends on the individuals fitness value $u(m)$. This selection probability determines if an individual is chosen for mating. There are several methods of assigning selection probabilities like roulette wheel selection (the likelihood of picking an individual is proportional to the individual's score), tournament selection (a number of individuals are picked using roulette wheel selection, then the best of these are chosen for mating), and rank selection (pick the best individual every time). Moreover it is important not to use a method which always picks the individuals with the best fitness because then the population will quickly converge to these individuals narrowing the search space.

The fourth step is to generate a new population $M(t + 1)$ using the individuals selected in step three to produce offspring applying the already mentioned genetic operators mutation and crossover with a predefined probability (see figure 2.3(a) and 2.3(b) for genetic operators).

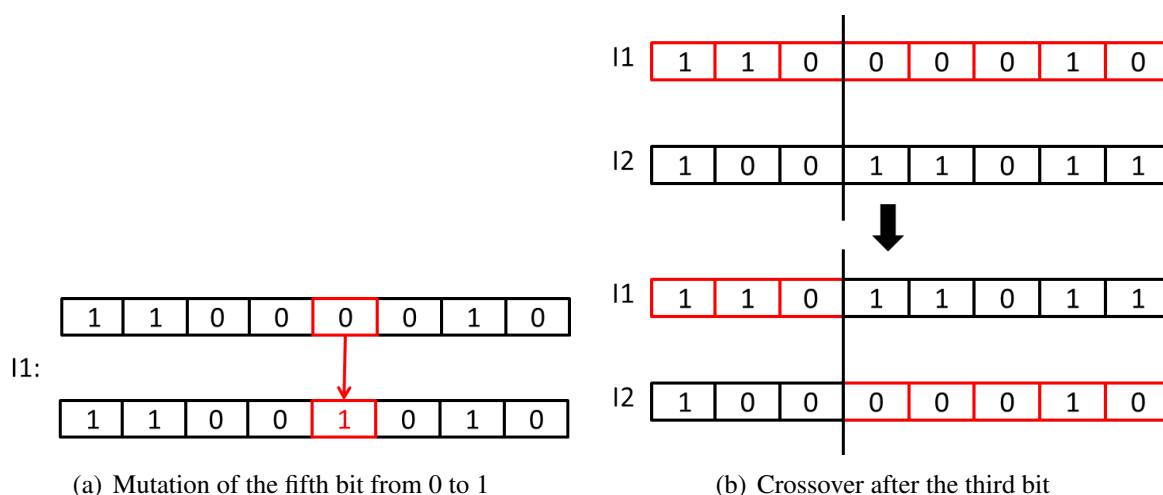


Figure 2.3: This figure shows the two genetic operators mutation and crossover. These allow to generate new individuals from already existing ones and to introduce new sets of parameters with possible better fitness values.

Steps two to four are then repeated until one of three possibilities occur. The best fitness in the current population reaches a given limit, the best fitness does not increase over several, predefined generations, or the steps two to four are repeated for a specific number of times.

2.6 Quantitative Structure-Activity Relationship

For the development of a new drug it is important not only to know its chemical formula but also its conformation. The underlying principle for that is the so called *lock and key principle* postulated by Emil Fischer in 1894 [Fis94] stating that an active compound has to be spacial complementary to its target to form a complex. But as we know today there are several other factors that influence the building of an active complex. Those can be direct features of the molecules, like hydrophobicity, partial atomic charge, binding sites etc., or there can be influences from the surrounding solution (e.g., water) so that a ligand changes its conformation in the binding process. These considerations lead to the expansion of the *lock and key principle* to the *induced fit theory* in 1958 [Kos58][Kos94]

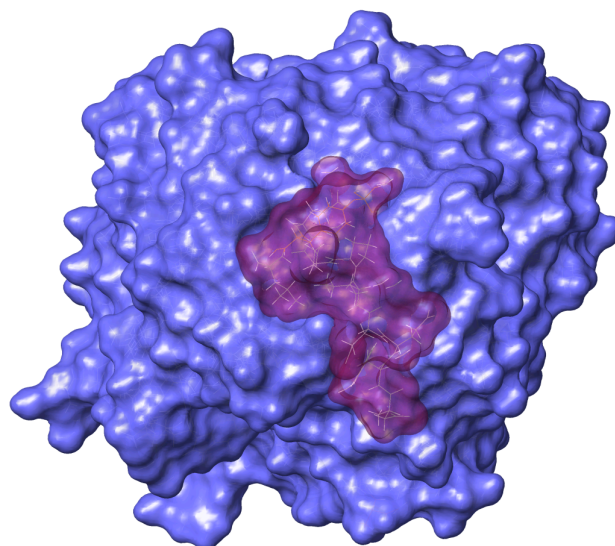


Figure 2.4: This figure shows a Thrombin-Hirudin complex. The Hirudin(magenta) being the key to the Thrombin(blue) lock.

Also new to this theory was the introduction of flexible binding sites which can account for differences in specificity and affinity. This leads to the conclusion that the biological activity is a direct function of the ligands three dimensional structure which in turn is the fundamental premise for the *quantitative structure-activity relationship* (QSAR) [SOW04]. QSAR Methods attempt to represent the relationship between structural attributes of molecules and their biological activity. In the beginning QSAR models were used to retrospectively analyze the activity modulation of molecules in a specific subset. But in the last decade QSAR models have been increasingly used for predictions on novel derivatives of well known ligands [Eki04]. To be applicable to such a use the applied QSAR models must be able to generalize and predict activities correctly beyond the chemical space defined by the given training data.

To that end a large number of methods has been described in the literature since the beginning of the research on QSAR. The early methods implemented only 2D features of molecules (e.g. the connection table of a molecule), while newer ones often include 3D features like the chemical properties of molecules in their bioactive conformation [SJ93] [OW91].

3 Materials and Methods

In this chapter the two main strategies applied to the problem and the overall process will be explained in detail and their function will be exemplified. The parameters used for the experiments and their progress will be given. The implementation of the algorithms or the use of external programs or code will be described. All algorithms were written in Java.

3.1 Overall process

Because this work consist of a concatenation of different machine learning and chemoinformat-ical methods I will first give an overview of the whole process and then explain the appointed methods in depth.

The aim was to see if the best models for an activity prediction included the actual active structures of the given molecules or if a better model could be found without them.

In this work I used two different approaches. The first was to precompile a set of conformers for each molecule maximizing the coverage of the conformer space and the second was to create random new conformers during the optimization process. From the set of precompiled conformers 100 (or the maximum available if lower then 100) were chosen equally distributed over the calculated relative energy range for each molecule and used as the training set. In both approaches the optimization was done by a genetic algorithm. The deciding facts for using a heuristic (in this case the genetic algorithm) were that both a full search of the optimization space isn't feasible for 100 molecules each with at least 100 conformation and that the solution hyperplane is very jagged and there was no information about a starting point.

The information about the molecules conformation were encoded in the GA's genes, either as a direct reference to the whole conformation (in the precomputed approach) or as single dihedral angles for each rotatable bond in each molecule (in the implicit approach).

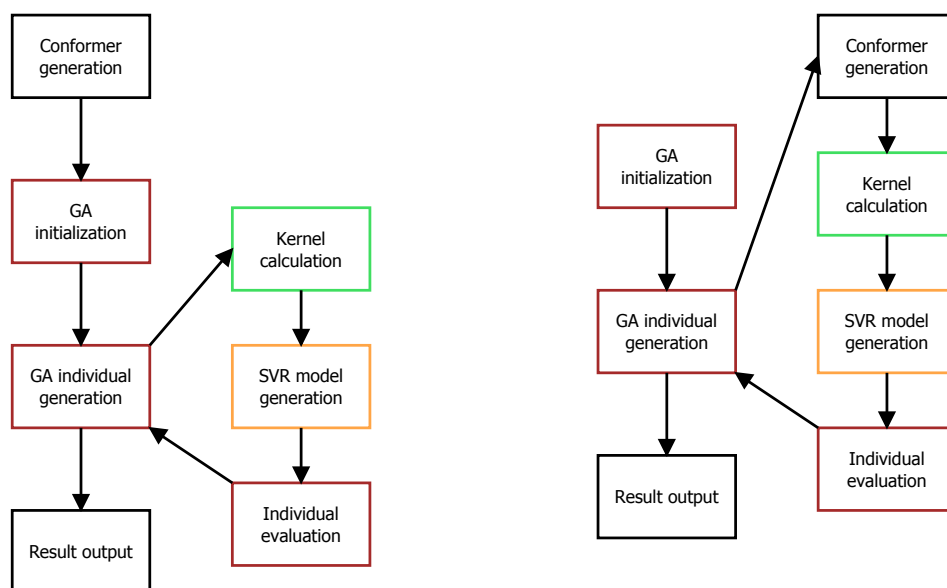
After each generation of the GA the fitness of it's individuals was calculated. In this case each individual corresponded to a set of conformers for which a kernel matrix using one of the following kernel methods were used.

- Probability Product Kernel (PPK)
- radial basis function(RBF)
- Atom Pair Kernel (APK)

The first two of which were working on the RDF of a given molecule and the third one working directly on 3D model of the conformation. Each kernel matrix therefore consisted of similarity measures between the molecules. And for each molecule pKi value was known. These informations were used to build a SVR model to predict the activity of an unknown molecule in relation to it's similarity to the molecule in the training set. For each model a set of best

parameters was searched using 5 repetitions of *leave-one-out* convoluted with a 5-fold cross-validation. These best parameters were used to compute the MSE of the model which in turn served as the fitness value for each individual.

The next generation of individuals in the GA was then generated using standard GA operators such as mutation and cross over. The individual selected to mate for the next generation according to their fitness value.



(a) In the first approach the conformer sampling was done before the optimization process

(b) In the second approach the conformer sampling was done implicitly as part of the optimization process by mutating the conformers

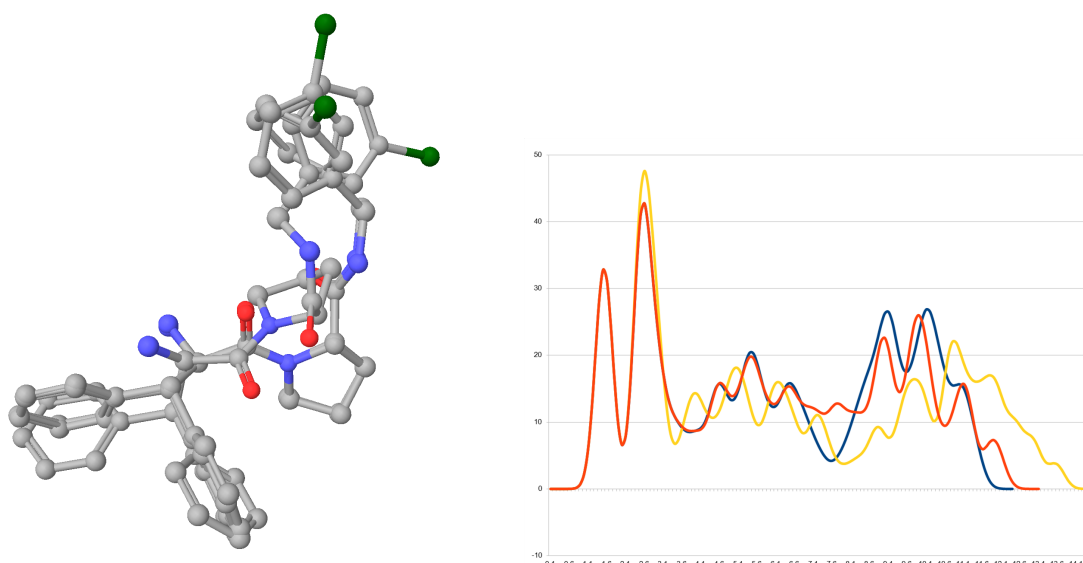
Figure 3.1: These two figures show the different procedures of the two approaches. Both consist of four frameworks indicated by the different colors. The conformer generation (either with MacroModel or implicit), the GA (with JavaEva2) which runs the optimization loop, the kernel matrix computation and the SVR modeling (with libsvm).

The process of generating new generations, the calculation of each kernel matrix and the evaluation via the SVR was then repeated 200 times. The development of the MSE for each individual and the RMSD between the conformations of the individual and the known active structure was calculated.

3.2 Radial Distribution Function

An important prerequisite for the computation of active structures with respect to the different conformations is keeping some kind of knowledge about the 3D structure of the molecules throughout the whole process. Therefore a molecular representation is needed that guarantees 3D sensitivity. To do so there are some prerequisites for a structure code

- independence from the numbers of atoms, i.e. the size of the molecule,
- unambiguity regarding the three-dimensional arrangement of the atoms and
- invariance against translation and rotation of the entire molecule



(a) Overlay of three different conformations of the same molecule (b) The RDF for the three molecules shown on the left

Figure 3.2: These figures show an overlay of three conformations of the same Thrombin inhibitor and their RDF. While the internal distances of the ring systems stay the same (i.e. the peaks representing the ring systems at $r \approx 1.5$ and $r \approx 2.6$ overlap for all three molecules) their relative spatial position vary (i.e. the peaks representing the distances of the ring systems among themselves at $r \approx 6$ to $r \approx 12$ are set off)

One method that meets all of the above requirements and which I used in this work is a derivation of the *3D-Molecule Representation based on Electron diffraction* (3D-MoRSE) [Sch96] [Sel97], the *radial distribution function* [Gas96] [Gas97]. In general this function gives the probability to find a pair of atoms in the given molecule with similar properties in the distance r to each other.

$$g(r) = f \sum_{i=1}^{N-1} \sum_{j>i}^N A_i A_j e^{-B(r-r_{ij})^2} \quad (3.1)$$

where f is the scaling factor and N is the number of atoms. The exponential term consists of the distance r_{ij} between two atoms i, j and the smoothing factor B for the probability distribution which will be explained later. A_i and A_j are the characteristic Atom properties. The properties used in this work are standard properties of the JoeLib2 framework, for example:

- | | |
|-----------------------------|-------------------------------|
| • Electro-topological state | • Electronegativity (Pauling) |
| • Partial charge | • Intrinsic state |
| • Atom mass | • Free electron count |
| • Electron affinity | • Hybridisation |
| • Van-der-Waals volume | • Heavy atom valence |
| • Electrogometrical state | • Implicit valence |

This distribution function allows to embed a lot of additional information, e.g. bond distances, ring types, planar and non-planar systems and atom types, all of which are important in calculating the similarity of two molecules or as in this case the similarity of two conformers of the same molecule.

An important factor in using the *radial distribution function* is the resolution of the 3D model of the molecule on which the formula is applied. Using exact distances stands in contrast to physical reality and further restricts the application of any ability to interpolate for better results. Even though if one wants to compute the similarity of two conformers using paired atomic distances a certain amount of fuzziness is necessary to account for flexibility and errors in the initial measurement. Therefor the width of the peaks in the radial distribution function is determined by the factor B . As an approximation the value of B can be given as a relation between B and the chosen step size Δr [Hem99] by

$$B \approx (\Delta r)^{-2} \quad (3.2)$$

In this work I started with a value of $B = 1000$ for my computations. But on realizing that even slight changes had a large effect on similarity values I successively lowered it up to a value of $B = 10$ where only rotations of whole ring systems had a noticeable effect on similarity. The step size Δr was always set to value of $\Delta r = 0.1\text{\AA}$.

Implementation

In this implementation the function was internally represented by a vector of double values each representing the value of the RDF at point $g(r)$ with $r \in 0.1\mathbb{N}$. The length of the vector, and therefore the range of the function with y values ≥ 0 was predetermined by measuring the longest distance of atom pairs in a molecule over all molecules in the dataset and adding 2\AA as security margin. The preceding scaling factor f was not used (i.e. always set to $f = 1$).

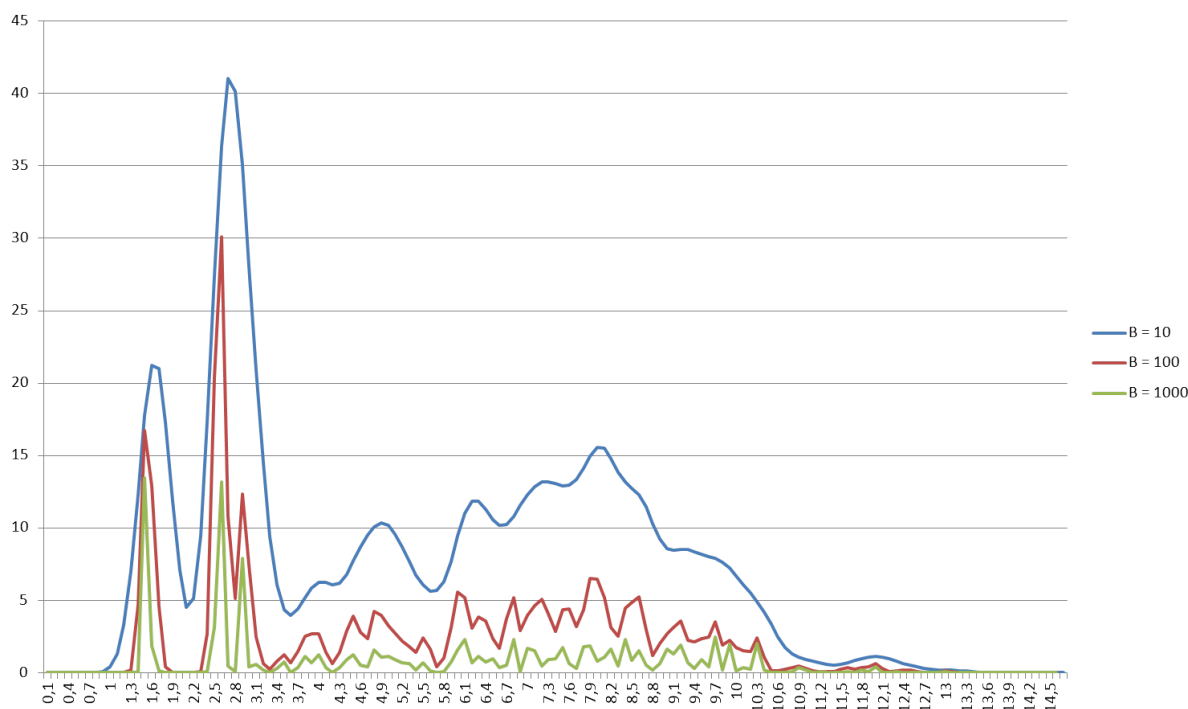


Figure 3.3: This figure shows the overlay of three RDF diagrams of the same molecule with three different values for B: 10, 100, 1000. One can see that with increasing B the smoothness decreases but the information value increases.

3.3 Kernel

3.3.1 Probability Product Kernel

One of the two methods used in this work to give a 3D sensitive representation of a molecule was the *radial basis function* (RBF). This function can be regarded as a distinct distribution of atom pairs in the given molecule.

Typical kernels compute a generalized inner product between two input objects χ and χ' which is equivalent to applying a mapping function ϕ to each object and then computing a dot product between $\phi(\chi)$ and $\phi(\chi')$ in a Hilbert space [Jeb04]. This kernel considers the case of a mapping $\phi(\chi)$ being a probability distribution $p(x|\chi)$, restricting the Hilbert space to the space of distributions embedded in the Hilbert space.

In this work the probability distribution $\phi(x|\chi)$ is given as the RDF function which leads to the definition of the probability product kernel as follows

Definition Let p and p' be probability distributions on a space X and ρ be a positive constant. Assume that $p^\rho, p'^\rho \in L_2(X)$, i.e. that $\int_X p(x)^{2\rho} dx$ and $\int_X p'(x)^{2\rho} dx$ are well defined (not infinity).

The **probability product kernel** (PPK) between distributions p and p' is defined as

$$k^{prob}(p, p') = \int_X p(x)^\rho p'(x)^\rho dx = \langle p^\rho, p'^\rho \rangle_{L_2}. \quad (3.3)$$

Furthermore it is well known that $L_2(X)$ is a Hilbert space. Hence the defined kernel is positive definite for any set of \mathcal{P} of probability distributions over X such that $\int_X p(x)^{2\rho}$ is finite for any

$p \in \mathcal{P}$.

Implementation

The first idea was to implement the computation of the probability product kernel with the numerical integration of the given RDF functions via Simson's rule (see figure 3.4)

$$\int_a^b f(x)dx \approx \frac{a-b}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]. \quad (3.4)$$

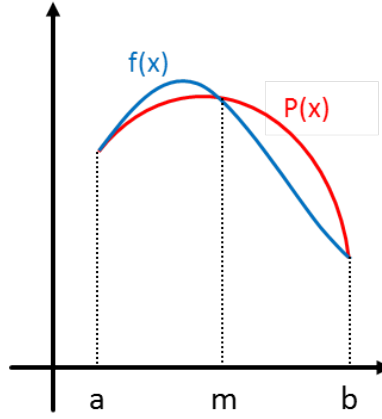


Figure 3.4: This figure shows the approximation of a function $f(x)$ by a quadratic interpolation $P(x)$.

The RDF was interpolated by Simpson's rule in steps of 0.01 which led to an exact calculation of the integral up to the 6th decimal place and also allowed to freely choose the factor ρ in the PPK formula.

But the first tests on this implementation showed that the computation of a single kernel value could take up to 10 seconds resulting in maximum total of 1.5 hours per kernel matrix. Being unfeasible due to the enormous amount of need computational power I decided to fix the parameter ρ with $\rho = 1$. With this the kernel takes the form of the expectation of one distribution under the other:

$$k(p, p') = \int p(x)p'(x)dx = E_p[p'(x)] = E_{p'}[p(x)] \quad (3.5)$$

This is also called the *expected likelihood kernel*.

3.3.2 Radial Basis Function Kernel

Another method of measuring similarity between the two result vectors A and B of the RDF is the use of a radial basis function. A radial basis function (RBF) kernel, also known as an isotropic stationary kernel [HG04], is defined by a function $\psi : [0, \inf) \rightarrow \mathbb{R}$ such that

$$k(x, x') = \psi(\|x - x'\|) \quad (3.6)$$

where $x, x' \in X$ and $\|\cdot\|$ denotes the Euclidean norm. The use of a special RBF kernel, the Gaussian RBF kernel has been suggested in [Guy93] with

$$k(x, x') = \exp\left(-\frac{\sum_1^n \|x_i - x'_i\|^2}{2\sigma^2}\right) \quad (3.7)$$

where x_i and x'_i are the single data points in the result vectors of the RDF. And σ defining the width of the sphere surrounding the corresponding training pattern [Cha05].

The issue on implementing this kernel was to find a viable value for the σ parameter in the above formula. On choosing σ to low the patterns will tend to be very similar over-fitting the model and taking away its ability to generalize outside its bounds. While choosing σ to high will have opposite effect letting the patterns appear very dissimilar and under-fitting the model. So finding a optimal value for σ is more about finding an acceptable trade-off between over-fitting in dense areas and under-fitting in sparse areas.

3.3.3 Atom Pair Kernel

While the preceeding kernel were based on a RDF representation another method to compare the 3D structure of two molecules or the different conformations of the same molecule is to represent the molecule as a trie data. For that I use a derivate of the *optimal assignment of atom pairs* [Jah09].

This method is based on a matrix $D = \left\lfloor \frac{d_{ij}}{b} \right\rfloor$ of binned geometrical distances between the three-dimensional coordinates of atoms i, j . Where d_{ij} are the atomic distances and b is the binning factor. The matrix D is used a a lookup table for the information needed to build a trie containing all the geometrical information for all atom pairs from a fixed atom i to any other atom. Where a trie is a prefix based search tree that can be applied to any symbolic pattern with a reading direction.

At the beginning the trie of atom i only consists of the root labeled with the hash code of the atomic symbol i . To fill the trie patterns of the form

$$\text{hash}(\text{symbol}(i)), d_{ij}, \text{hash}(\text{symbol}(j)) \quad (3.8)$$

are inserted successively as ordered triplets. An example of a local atom pair environment and the corresponding trie is shown in figure 3.5.

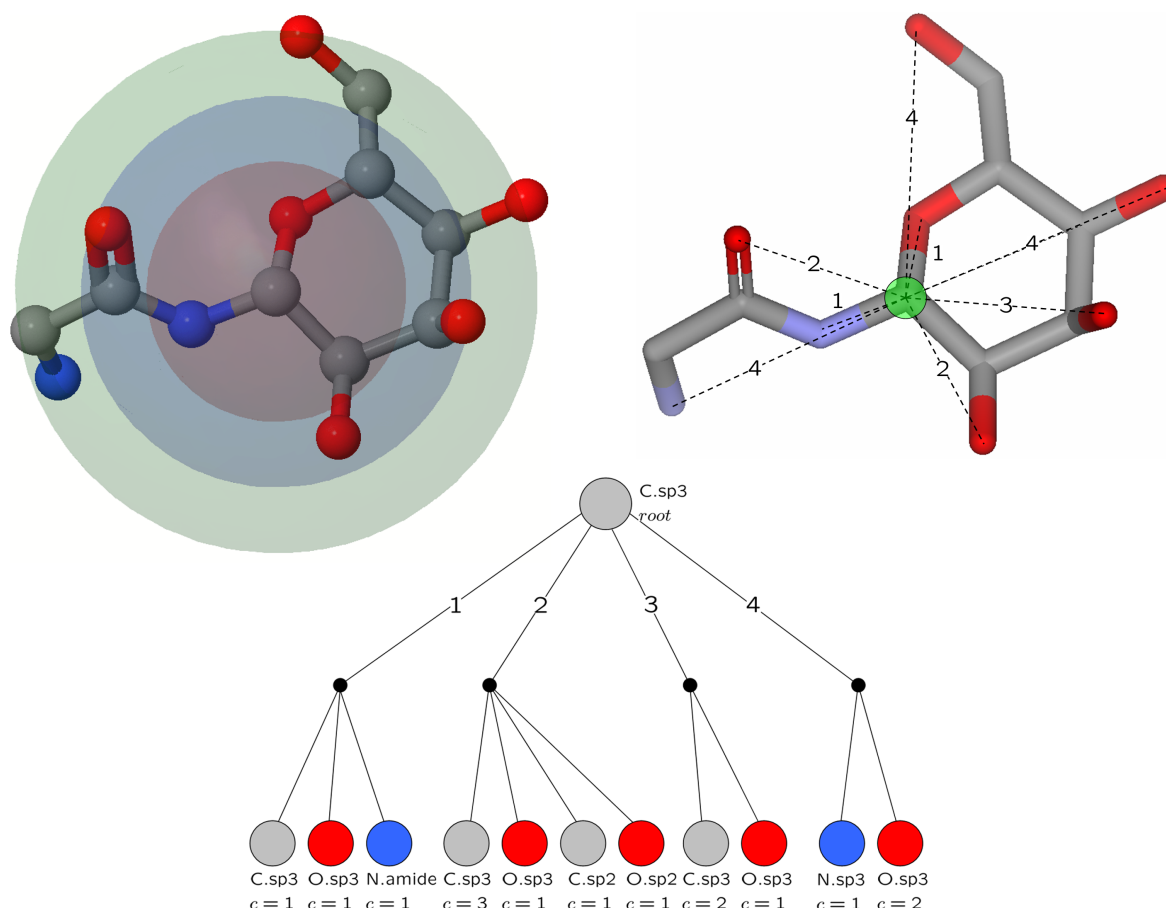


Figure 3.5: **Binned geometrical distances, spheres and trie.** The upper left figure shows the spheres of the binned geometrical distances 1.0, 2.0 and 3.0Å for the centered carbon atom. The sphere of the binned geometrical distance of 0.0Å (distances in the range [0.0; 1.0)) is not visualized as individual sphere because it contains no atoms. The upper right figure illustrates the resulting local atom pair environment of binned geometrical distances. For simplicity, only the distances to non-carbon atoms are displayed. The lower figure visualizes the corresponding trie of geometric atomic distances of the annotated atom in the upper figures. The root and leaves are labeled with the corresponding atom type. The leaves contain additionally the total number of occurrences in the local atom pair environment.[JZ10]

The representation of a local atom environment as tries allows the comparison of two local atom environments by comparing the tries. This can be achieved by applying a well known similarity measurement like the Tanimoto coefficient

$$T(A,B) = \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B} \quad (3.9)$$

In this case let L_A, L_B be two sets of local atom pair environments of two molecular graphs A, B and $l_{A_i} \in L_A, l_{B_j} \in L_B$ the tries i, j of the nominal features (atom pair environments of atoms i, j).

Then the Tanimoto coefficient can be defined as

$$\text{Sim}(l_{A_i}, l_{B_j}) = \frac{|l_{A_i} \cap l_{B_j}|}{|l_{A_i} \cup l_{B_j}|} \quad (3.10)$$

Implementation

The implementation used in this work was based on the *Chemistry Development Kit* (CDK) [Ste03] [Ste06] implemented by [Jah09].

The single arbitrary parameter b was initially set to $b = 0.1$ and subsequently set to $b = 0.2$ to account for errors in measurement of the crystal structure.

3.4 Dataset

The dataset used in the experiments consisted of two parts. A precompiled set of 88 molecules taken from [Boe99] and a smaller set of 12 molecules compiled for this work. All of the molecules in the dataset were thrombin inhibitors with a known pK_i value. However only the 12 molecules in the compiled dataset had crystallographic determined active structures. The active structures were gained by taking the crystal structure analysis of thrombin with the respective ligand and extract the bound ligand from the whole structure.

The first step therefore was to search for all potential thrombin inhibitors in the scBDP¹ [Kel06]

The second step was to find an entry with the identical structural formula in the Binding Database² [XG02] for information about pK_i Values and publications.

The third and final step was to download the crystallographic analysis given by the PDB ID from the Protein Data Bank³ [Ber77] and to extract the bound ligand with Schrödinger's Maestro program. Thus these 12 ligands will from now on be referenced by their originating PDB ID. They are depicted in figure 3.6 and their data and publications is shown in table 3.1.

The 88 precompiled structures were only available as structural formulas so they had to be converted into a valid 3D conformation. To achieve this they were converted with the CORINA program [Sad94].

Thrombin inhibitors were chosen both for their high flexibility and the fact that the interactions of inhibitors and Thrombin are well investigated and there are several well documented studies including crystal structures.

3.5 Conformation Sampling

3.5.1 Precomputed Conformation Sampling

The first strategy to be pursued was to precompute a set of conformers for all molecules, pick a subset of 100 of these conformers per molecule (or less, if less than 100 were available) and use the genes in the GA as indices for the molecules to choose from. Therefore a mutation operation in the GA lead not only to a single change in the conformation but could lead to a whole different one.

¹<http://bioinfo-pharma.u-strasbg.fr/scPDB/>

²<http://www.bindingdb.org>

³<http://www.rcsb.org/pdb>

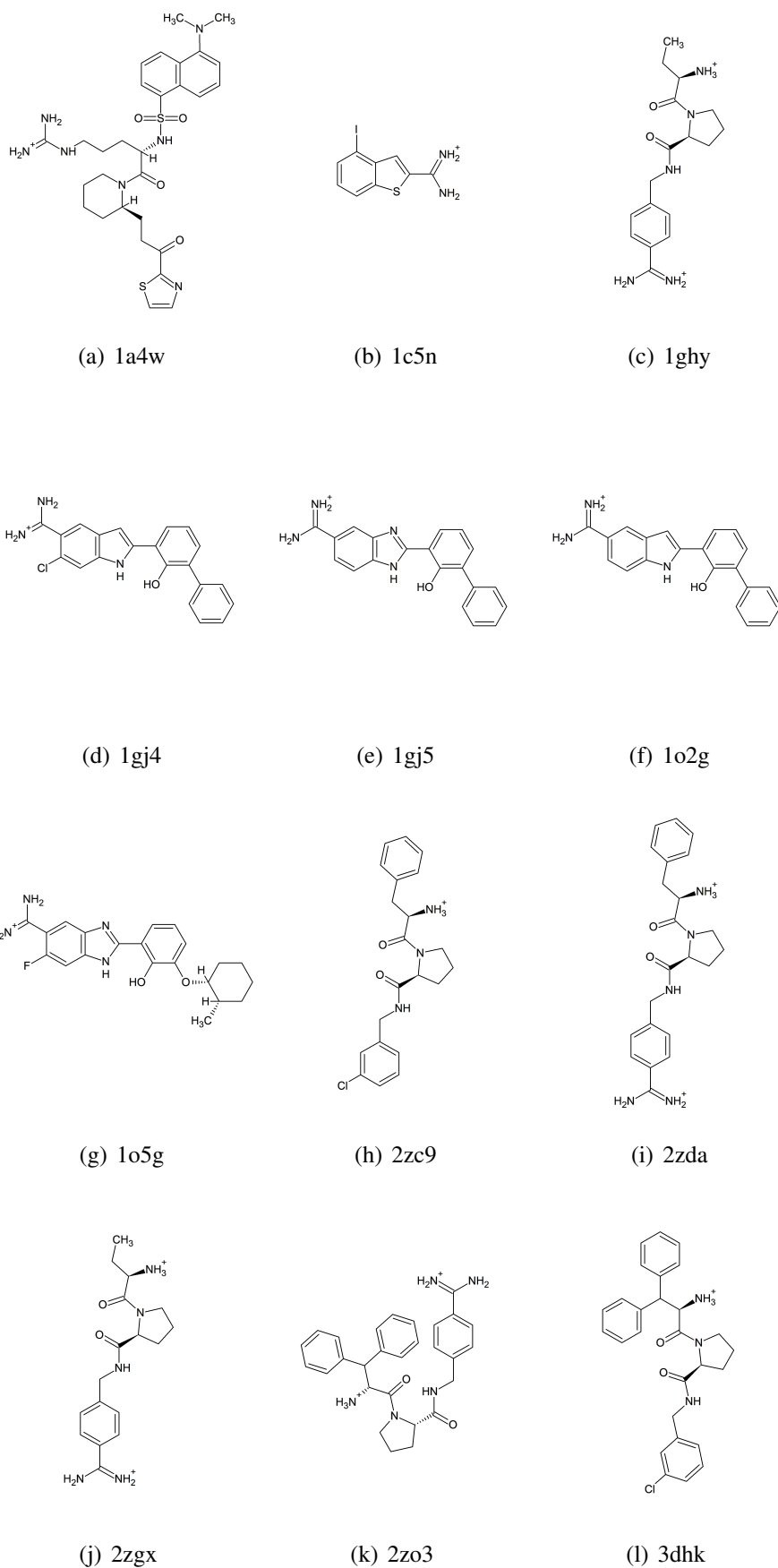


Figure 3.6: These figures show the 12 molecules with known active structure used in this work. They are labeled with the PDB ID they were extracted from.

PDB ID	pKi value	resolution (in Å)	first published in
1a4w	7.796	1.80	[Mat96]
1c5n	4.699	1.50	[Kat00]
1ghy	5.071	1.85	[Kat01a]
1gj4	4.222	1.81	[Kat01b]
1gj5	6.347	1.73	[Kat01b]
1o2g	6.495	1.58	[Kat03]
1o5g	4.957	1.75	[Kat04]
2zc9	7.327	1.58	[Bau09]
2zda	8.398	1.73	[Bau09]
2zgx	6.745	1.80	[Bau09]
2zo3	10	1.70	[Bau09]
3dhk	6.744	1.73	[Bau09]

Table 3.1: This table gives information of all used molecules for which the crystal structure was known

The conformations themselves were generated with the ConfGen program [WS10] which is based on the molecular modeling Program MacroModel [SI08b].

The first step the program takes is to identify *variable features* which are rotatable bonds, flexible ring systems and invertible nitrogens. ConfGen generally identifies a bond as rotatable if the following criteria are met:

- It is a single bond
- It doesn't lie within a ring
- Neither of the atoms connected by the bond is terminal (i.e. has no other bonds to it)
- Neither end of the bond is a CH_3 , NH_2 or NH_3^+ group
- Neither atom in the bond is bonded to two or three atoms that are all equivalent and are arranged with two- or three-fold rotational symmetry.

Ring conformers are generated using the same template based facility available in LigPrep [SI08a], Glide [Fri04], MacroModel [SI08b], or Phase [Dix06]. It is designed to generate a complete set of accurate, low energy ring conformation identifying individual rings with a *smallest set of smallest rings* (SSSR) method [Zam76]. When a ring system is identified it is compared to a set of 1252 templates to find the most similar template. This template is then used to calculate the relative energies of the ring within the molecule. There are N_{ri} combinations of ring conformations for a whole molecule:

$$N_{ri} = 2^{N_i} \prod_r N_{cr} \quad (3.11)$$

where N_i is the number of invertible nitrogen atoms, r runs over all flexible ring systems and N_{cr} is the number of templates selected to use for each individual ring system.

Each of the generated set of ring conformers is then processed as follows. First the potential of each rotatable bonds connecting the ring systems are calculated using a derivative of OPLS [Jor88] [Jor96] including a quick check of Lennard-Jones potentials of all atoms on one side of the bond to all on the other side to avoid local Van-der-Waals clashes. Then the potential

parameter	intermediate	comprehensive
maximum number of search steps	1000	1000
search steps per rotatable bond	75	75
minimum heavy atom RMSD (Å) for distinct conformer	1	0.5
minimum dihedral angle difference for polar hydrogens (°)	60	60
maximum relative energy for flexible rings (kcal/mol)	2.39	23.9
maximum number of ring conformations per ligand	16	128
maximum number of ring conformations per ring	8	64
maximum relative ConfGen energy (kcal/mol)	25	119.5
energy threshold for periodic torsions (kcal/mol)	5.74	5.74
restraint potentials for weak torsions in MacroModel (kcal/mol)	239	239
restraint potential half width (°)	10	10
suppress hydrogen-bond electrostatics in MacroModel	Yes	Yes
maximum relative energy all-atom energy in MacroModel (kcal/mol)	25	119.5

Table 3.2: This table shows the parameters used to generate the two datasets. The intermediate parameter set is more restrictive and almost certainly only picks energetic minima while the comprehensive parameter set allows for the algorithm to pick a conformation lying between two optima.

minima are computed and used to create sets of rotational bonds surrounding the molecular core (i.e. the part of the molecule remaining if every outer rotational bond is severed).

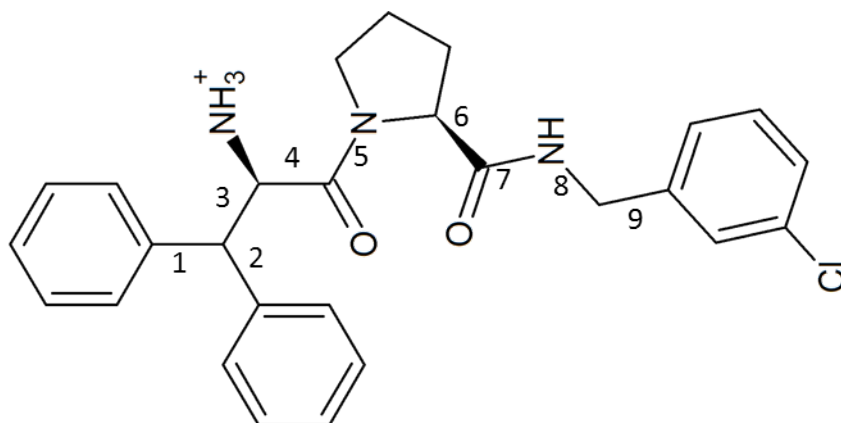
For each combination of ring system conformation, invertible nitrogen atom geometry and minima of rotatable bond dihedral angle all molecule conformations are compiled and, if the sum of all relative potential (to the one with the least energy) energies doesn't exceed a preset limit, the conformation is added to the resulting set of conformers.

In this work I used two sets of parameters for the algorithm described above. One restrictive and one permissive. While the restrictive parameter set only generated conformers where the rotatable bonds took up a local minimum energetic state the permissive one allowed more freedom. Thus conformations were picked lying in between local optima and allowing the GA to successively change more easily from one conformation to another. For the exact parameters used in the conformer sampling see table 3.2

3.5.2 Implicit Conformation Sampling

The second strategy to be pursued was to not use the precomputed conformation sampling but to generate a new set of conformations from generation to generation in the genetic algorithm. Therefore the encoding of the single individuals in the GA had to be different. In contrast to the

optimization of the precomputed conformation set, where each ‘gene’ represented the conformation ID of a whole molecule, here a ‘gene’ only represented a single rotatable bond within a molecule. While a mutation on one gene in the GA meant to pick a whole new conformation of the concerned molecule with possibly every rotatable bond affected, the mutation of a gene in the implicit conformation sampling only meant the alteration of a single rotatable bond. In addition it had to be ensured that the crossover operator didn’t cut in the middle of the encoding of a molecule but only at the end of one and the beginning of the other. Doing a crossover in the middle of a molecule could lead to an invalid conformation because it couldn’t be guaranteed that the molecule wouldn’t fold back on itself overlapping one or more atoms.



Bond Id	1	2	3	4	5	6	7	8	9
Dihedral Angle	10°	118°	70°	251°	188°	22°	103°	34°	62°

Figure 3.7: The figure shows an example of a molecule with nine rotatable bonds and the corresponding encoding as a gene for the GA. The denoted angles are the dihedral angles for a unique set of deterministically calculated atoms ‘surrounding’ the bond

But before encoding a molecule in the GA one had to know the exact number of rotatable bonds. For that each bond was inspected and had to meet a list of criteria to count as rotatable. The criteria used were the ones already implemented in the JoeLib2 framework:

- The atom at the beginning of the bond has to have a heavy atom valence of > 1
- The atom at the end of the bond has to have a heavy atom valence of > 1
- The bond order has to be 1
- The bond mustn’t lie in a ring system
- The atom at the beginning of the bond mustn’t have a hybridization of 1
- The atom at the end of the bond mustn’t have a hybridization of 1

If these criteria were met, the bond was added to the molecule’s rotatable bond list.

The unit with which the rotations were encoded was 1° (i.e. degree) where degree refers to the dihedral angle. An angle of 0° refers to the original crystallographic conformation.

For the first generation of the GA a initial set of conformations was computed picking a random value for the dihedral angle of each rotatable bond of each molecule in the dataset.

After each occurring mutation in the GA the according molecule was computed again with the new degree value. Where the new value was in reference to the original 0° value and not to the currently applied one.

To compute a rotation around a rotatable bond one has to rotate each atom belonging to one of either of the two bipartite graphs formed by splitting the molecular graph at the designated bond. The bipartite graph was calculated using a stack, adding the beginning atom of the bond and then recursively adding every atom bound to the ones already on the stack (except for the atom at the end of the designated rotatable bond) until no new atoms could be found. This was possible because no bond were allowed to be rotatable if they were in a ring system and no molecules with macrocycles were in the dataset.

The actual rotation was achieved by applying a quaternion to each of the atoms in the bipartite graph with the center of the coordinate system being the atom at the beginning of the bond.

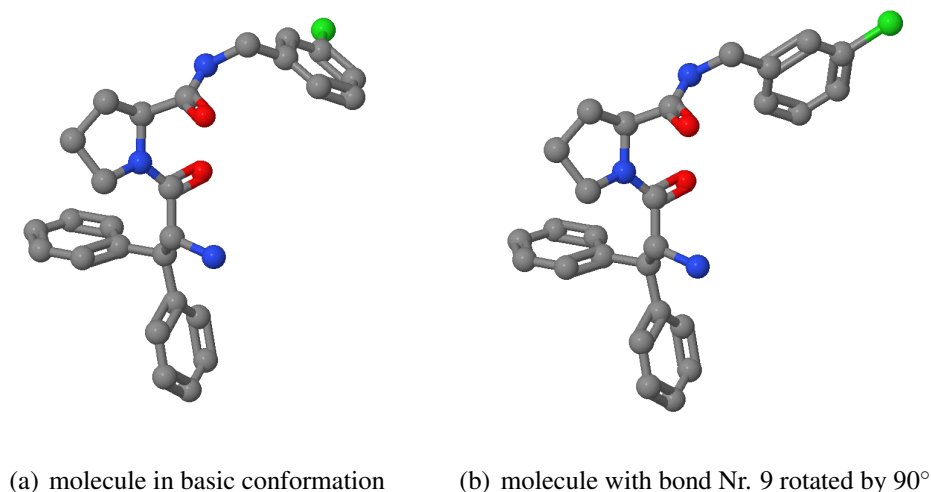


Figure 3.8: These two figures show an example of a rotation around one rotatable bond.

Both, at the initial random initialization of the conformations and at every mutation event it has to be ensured that the generated conformation is valid (i.e. no atoms or bonds overlap or lie too close to each other). Therefore for every new conformation all pairwise atom distances have to be calculated. The chosen value for a lower bound (gathered by calculating the average minimal distance for non-bound atoms over the whole original dataset) was 2\AA while distances of covalent bonds were ignored.

3.6 SVR

In this work I used the libSVM implementation by Chang and Lin [CL01]⁴. To compute the MSE a leave-one-out approach was applied. A model of the dataset (i.e represented by the kernel matrix) was built n times (with n being the size of the dataset) always with one different data point left out. For these datasets a five-fold cross-validation (inner fold) was run 5 times (inner runs). The inner fold was used to determine the best parameters of the regression (i.e values of parameters ϵ and c yielding the best performance on the validation dataset). The inner runs were used to the best model, with the just computed best parameters. The best model was then used to predict the currently left out data point. The set of parameters can be seen in table 3.3.

computation method	leave one out
inner folds	5
inner repetitions	5
c begin	-1
c end	5
ϵ begin	-7
ϵ end	-2

Table 3.3: This table shows the parameters used to calculate the MSE for the regression on the datasets.

⁴<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

4 Results

In this chapter I will present the results, interpret and discuss them. The results are divided by precomputed and implicit conformation sampling. Each of those two parts is further split by the used kernel methods and parameters. The results are mostly presented in chronological order to try and replicate my line of thought.

Evaluation Method

To explain the values shown in the following diagrams I will give a short explanation for each of them. The meaning of the values remains the same for every ‘run’-diagram in this work.

MSE

The ‘avg MSE’ value shown in each diagram is the average MSE value for one generation (i.e. 100 individuals). Where MSE is the best found *Mean Square Error* for each regression. In the corresponding table I will show the respective numerical values. The ‘Best individual MSE’ relates to the absolute minimum found by at least one individual.

RMSD

The ‘avg RMSD’ value is the average RMSD value for the conformation of the 12 molecules with known active structure encoded by the current individual to their respective active structure. The ‘Best individual RMSD’ relates to the individual with the lowest RMSD averaged over the 12 molecules in my dataset to their respective active structure.

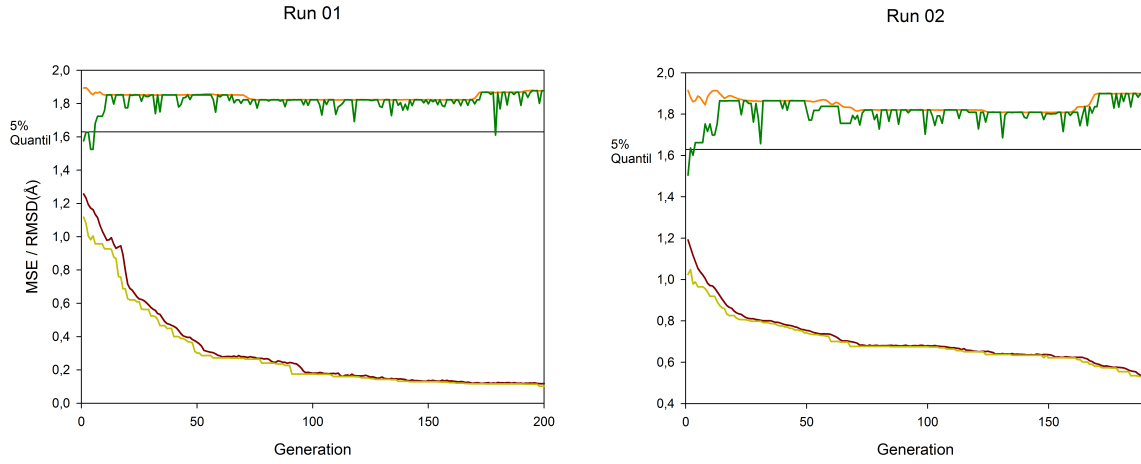
5% Quantile

The 5% quantile demarks the value where every point below this line lies in the lowest 5% of all possible values for the average RMSD. Its value is exactly 1.629. This was computed picking 20000 random combinations from the conformer sets of each molecule and building the average RMSD to their respective active structures. From this normal distribution the p -Quantile with $p = 0.05$ was calculated using the standard formula $x(p) = \mu + \sigma \cdot z(p)$ where μ is the expectation and σ^2 the variance and $z(0.05)$ was looked up in the normal distribution table.

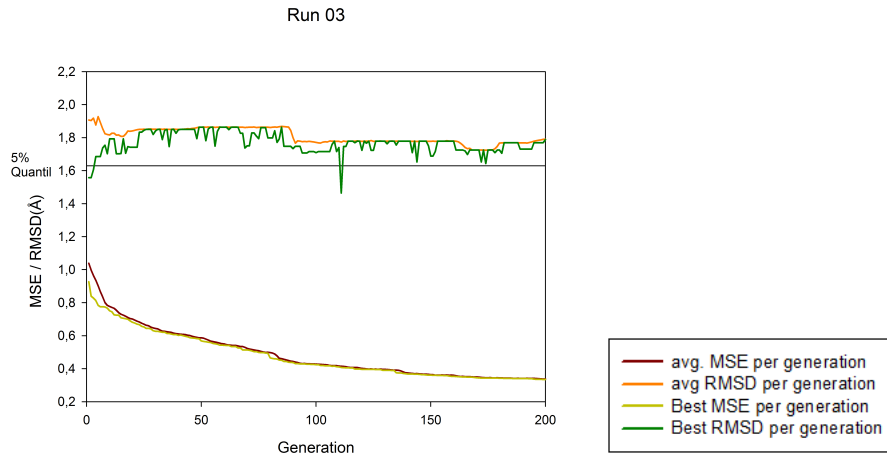
4.1 Precomputed Conformation Sampling

For the first experiment, the optimization of the QSAR model I used the dataset described earlier. The dataset first included all conformers produced with ConfGen where every molecule had a different amount of conformers created. To reduce the extent of the combinatorial size I picked the lower of either 100 or the number of conformers originally created. The selection method was to pick the conformers equally distributed over their relative energy to the

conformer with the absolute lowest energy to guarantee an equal distribution over the conformational space of each molecule. Because the output sets for each individual conformer were already sorted by their relative energy I simply had to pick every n -th conformer. Where $n = \text{number of conformers available} / 100$.



(a) This figure shows the avg. MSE, avg RMSD, best MSE and best RMSD for Run 01 where the PPK Kernel was used with parameter $B = 1000$ (b) This figure shows the avg. MSE, avg RMSD, best MSE and best RMSD for Run 02 where the RBF Kernel was used with parameters $B = 1000$; $\sigma = 100$



(c) [This figure shows the avg. MSE, avg RMSD, best MSE and best RMSD for Run 02 where the RBF Kernel was used with smoothing factor = 0.1

Figure 4.1: These figures show the results of the first three runs. One can see that the optimization works fine due to the MSE declining while the average RMSD only declines in Run03 using the APK but still doesn't reach the 5% quantile.

Paramter / Run Nr.	01	02	03
Kernel method	PPK	RBF	APK
RDF B factor	1000	1000	-
RBF Sigma factor	-	100	-
Smoothing factor	-	-	0.1
Mutation Probability	0.1	0.1	0.1
Mutate first 12 only	no	no	no
Conformation Sampling	intermediate	intermediate	intermediate
Start avg. MSE	1.256	1.191	1.039
End avg. MSE	0.118	0.536	0.337
Diff. Start/End	1.138	0.655	0.702
Best avg. MSE	0.109	0.536	0.337
Best individual MSE	0.117	0.526	0.335
Start avg. RMSD	1.893	1.915	1.907
End avg. RMSD	1.878	1.900	1.790
Diff. Start/End	0.015	0.015	1.117
Best avg. RMSD	1.822	1.808	1.724
Best individual RMSD	1.525	1.505	1.465

Table 4.1: This table shows the parameters and the results for Run01, Run02 and Run 03. Parameters denoted by ‘-’ are not available for the chosen kernel method

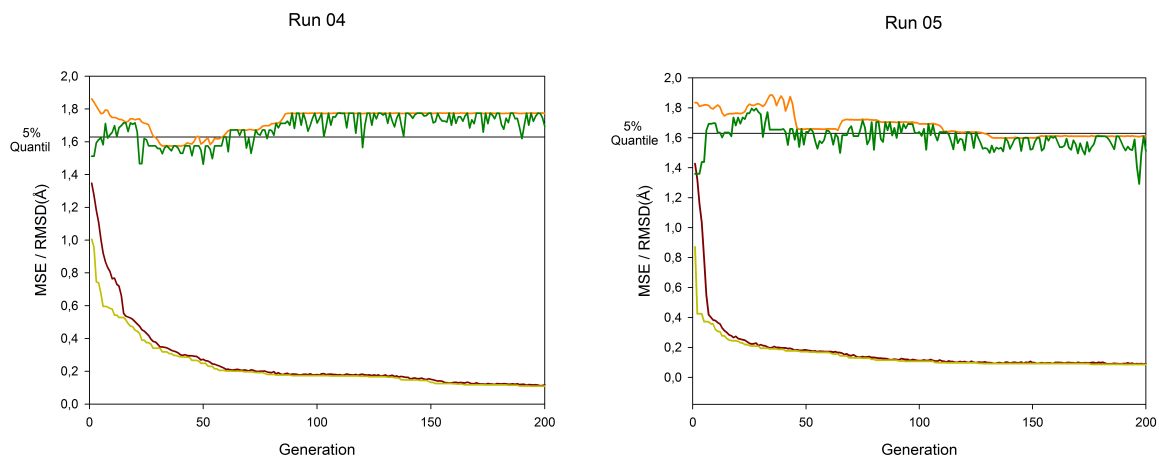
4.1.1 Initial Runs

In the first runs (run 01-03) of the experiment I used the PPK and the RBF Kernel on the RDF of the molecules and the APK to generate the kernel matrix. The parameters were set to their default values to check the overall function of the optimization. (see table 4.1)

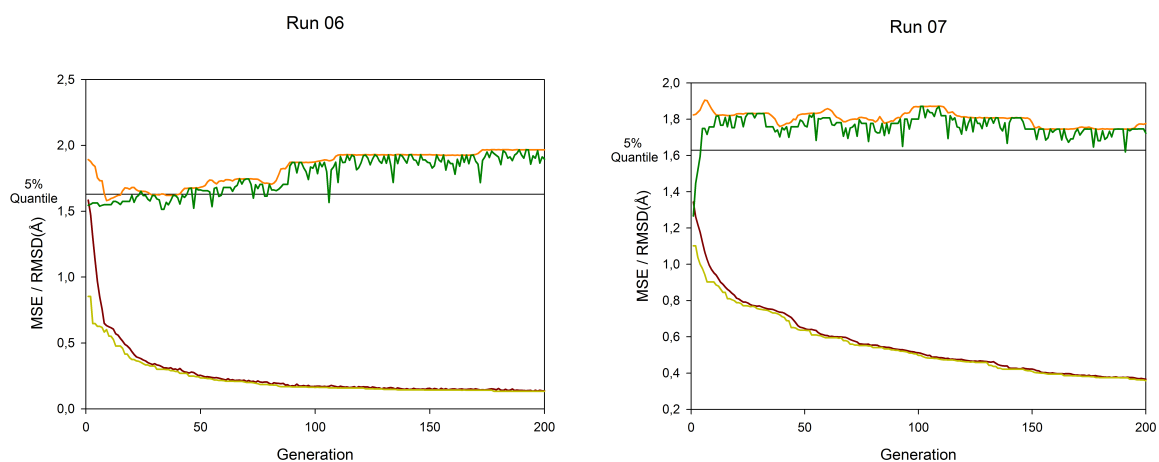
The results of these three runs are depicted on the left side in figure (4.1). One can see that the basic optimization is functional. The average MSE declines from the values of 1.256, 1.191 and 1.039 to values of 0.109, 0.536 and 0.337. But, while the average RMSD declines slightly in Run03 with the use of the APK it remains at the same level with the use of the PPK and RBF. Although some isolated individuals get below the 5% quantil mark they are dismissed in the next generation implying that the individuals with a higher average RMSD result in better models with lower MSEs.

My first consideration on evaluating these results where twofold. Either the use of the large dataset of molecules with unknown active structures impeded the decline of the ones with known active structures because their weight in the model building process was too large, or the parameters used were not fit for this kind of optimization.

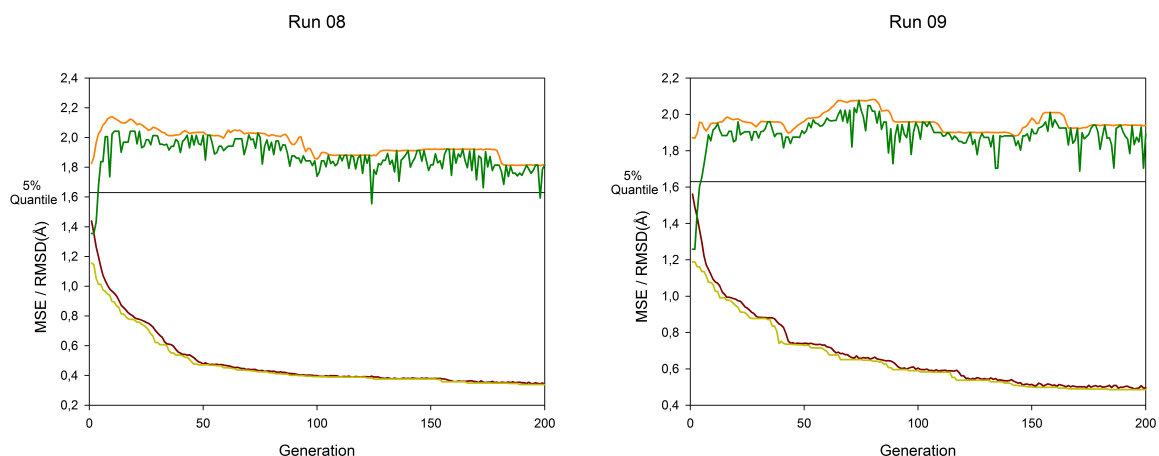
Therefore I consecutively lowered the size of the dataset to 56, 41 and 34 molecules, always including the 12 known active structures, and changed the parameters of the kernels used. Which are the B parameter for the RDF resulting in a smoother RDF function and the smoothing factor for the APK, both in the hope of a better generalization. These changes are shown in the next sections.



(a) This figure shows the results for the PPK with $B = 1000$ on the dataset with 56 molecules (b) This figure shows the results for the PPK with $B = 1000$ on the dataset with 41 molecules



(c) This figure shows the results for the PPK with $B = 1000$ on the dataset with 34 molecules (d) This figure shows the results for the RBF Kernel with $\sigma = 100$ on the dataset with 56 molecules



(e) This figure shows the results for the RBF Kernel with $\sigma = 100$ on the dataset with 41 molecules (f) This figure shows the results for the RBF Kernel with $\sigma = 100$ on the dataset with 34 molecules

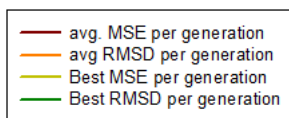


Figure 4.2: These figures show the results of the runs with reduced datasets for the PPK and RBF Kernel.

Parameter / Run Nr.	04	05	06	07	08	09
Kernel method	PPK	PPK	PPK	RBF	RBF	RBF
RDF B factor	1000	1000	1000	1000	1000	1000
RBF Sigma factor	-	-	-	100	100	100
Mutation Probability	0.1	0.1	0.1	0.1	0.1	0.1
Mutate first 12 only	no	no	no	no	no	no
Conformation Sampling	interm.	interm.	interm.	interm.	interm.	interm.
Dataset size	56	41	34	56	41	34
Start avg. MSE	1.3470	1.4283	1.5836	1.3432	1.4391	1.5611
End avg. MSE	0.1184	0.0917	0.1366	0.3663	0.3422	0.4955
Diff. Start/End MSE	1.2286	1.3366	1.447	0.9769	1.0969	1.0656
Best avg. MSE	0.1128	0.0885	0.1343	0.3663	0.3422	0.4897
Best individual MSE	0.1091	0.0856	0.1334	0.3578	0.3393	0.4862
Start avg. RMSD	1.8626	1.8348	1.8924	1.8239	1.8206	1.8717
End avg. RMSD	1.7738	1.6095	1.9664	1.7765	1.8151	1.9408
Diff. Start/End RMSD	0.0888	0.2253	-0.074	0.0474	0.0109	-0.0691
Best avg. RMSD	1.5744	1.5977	1.5805	1.7443	1.8120	1.8688
Best individual RMSD	1.4630	1.2913	1.5148	1.2660	1.3552	1.2588

Table 4.2: This table shows the parameters and the results for Run 04 through Run 09. Parameters denoted by ‘-’ are not available for the chosen kernel method

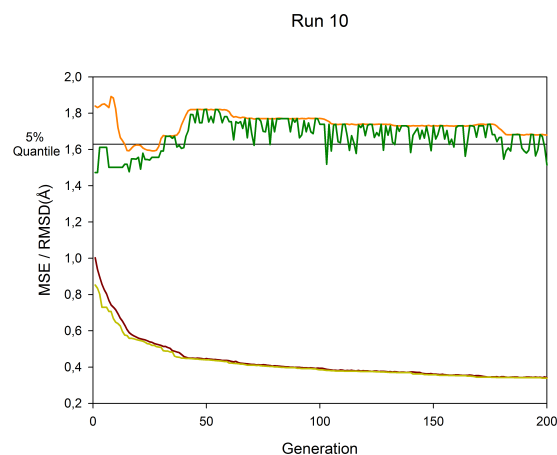
4.1.2 Reduced Dataset with PPK and RBF Kernel

To see if reducing the dataset size would yield models with a lower average RMSD I ran the PPK and the RBF on datasets where the only every 2nd, 3rd and 4th molecule with unknown active structure were included. The hypothesis was that due to the fact that the overall influence of the known active structures on the model is higher and if the general assumption of good models consisting of good data (i.e. the active structure) the RMSD would be lower.

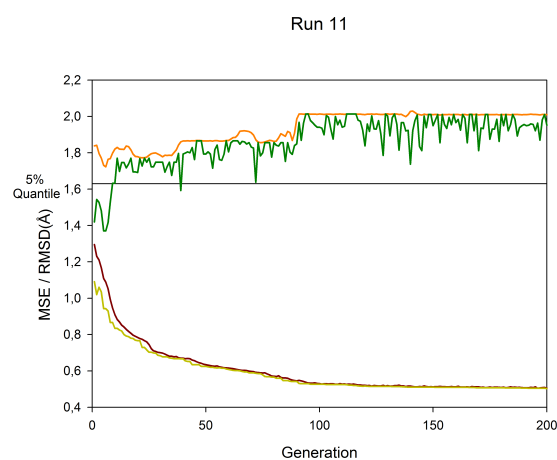
The results of these runs are shown in figure 4.1.1 and table 4.2. With the use of the PPK (runs 04-06) the average RMSD gets below the 5% quantile at some point. In run 04 and run 05 the average RMSD gets below the 5% quantile within the first 50 generations but returns to its starting level shortly after and stagnates. In run 05 however the average RMS stays at a relative high value in comparison to run 04 and 06 but declines to a value below the 5% quantile mark after 50 generations. In addition the MSE of the best model found for run 05 was the lowest of all three runs with final value of 0.0917 in contrast to 0.1184 and 0.1366 for runs 04 and 06.

With the use of RBF kernel (run 07, run 08 and run 09) the average RMSD didn’t get below the 5% quantile in any of the 3 runs. Although run 08 on the dataset with 41 molecules showed a steady decline of the average RMSD which is similar to the results of run 05. It is noticeable that the initial generation of all three runs consisted of at least one individual with a very low average RMSD and considering the low starting RMSD more than one. These individuals however were dismissed in the first 25 generations resulting in a average RMSD. Furthermore the best models found by using the RBF kernel had MSE values of 0.3578, 0.3393 and 0.4852 which is for each more then three times the MSE value of the best model for the PPK with corresponding dataset size where the values are 0.1091, 0.0856 and 0.1334.

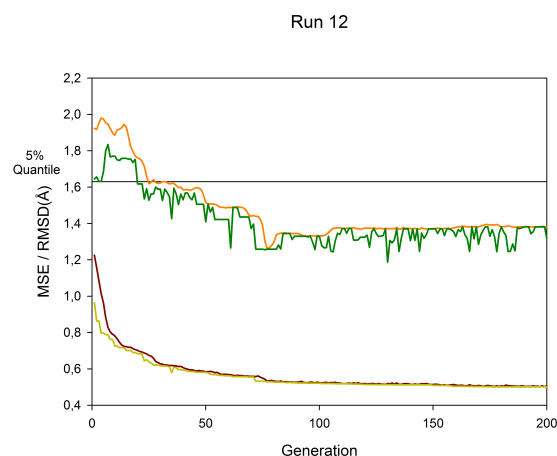
Considering this direct comparison of the PPK and the RBF kernel the PPK shows better results, both for the modeling and for the use of the actual active structure.



(a) This figure shows the results for the APK with $smoothingfactor = 0.1$ on the dataset with 56 molecule



(b) This figure shows the results for the APK with $smoothingfactor = 0.1$ on the dataset with 41 molecule



(c) This figure shows the results for the APK with $smoothingfactor = 0.1$ on the dataset with 34 moleculel

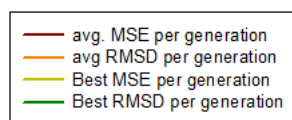


Figure 4.3: These figures show the results of the runs with reduced datasets for the APK. Run 12 is the model with lowest final average RMSD of all runs.

Parameter / Run Nr.	10	11	12
Kernel method	APK	APK	APK
RDF B factor	-	-	-
RBF Sigma factor	-	-	-
Smoothing factor	0.1	0.1	0.1
Mutation Probability	0.1	0.1	0.1
Mutate first 12 only	no	no	no
Conformation Sampling	intermediate	intermediate	intermediate
Dataset size	56	41	34
Start avg. MSE	1.0026	1.2949	1.2255
End avg. MSE	0.3420	0.5053	0.5067
Diff. Start/End MSE	0.6606	0.7896	0.7188
Best avg. MSE	0.3420	0.5051	0.5023
Best individual MSE	0.3390	0.5028	0.5012
Start avg. RMSD	1.8399	1.8381	1.9241
End avg. RMSD	1.6799	2.0098	1.3813
Diff. Start/End RMSD	0.16	-0.1717	0.5428
Best avg. RMSD	1.5922	1.7233	1.2618
Best individual RMSD	1.4729	1.3699	1.1870

Table 4.3: This table shows the parameters and the results for Ru10, Run11 and Run 12. Parameters denoted by ‘-’ are not available for the chosen kernel method

4.1.3 Reduced Dataset with Atom Pair Kernel

The reduction of the dataset had a similar effect on the use of the APK as it had on the RPK. The starting average RMSD was 1.8399, 1.8381 and 1.9241 and while run 10 and 12 had a considerably lower end RMSD with 1.699 and 1.3913 the final RMSD of run 11 was 2.0098. Which is 0.1717 higher than the start RMSD.

The first and third run, 10 and 12 show a similar development as the earlier runs 04, 05 and 06 with the average RMSD dropping by several percent around generation 50. But in contrast to all other previous runs the RMSD of run 12 declines further giving an indication that the optimization reaches a point where it can drop into several minima one of them being a model that included structures more likely to be near the conformation of the active structure.

Further one can see that for all three kernel methods the overall end MSE value rises with descending dataset size. This can be lead back to loss of information with decreased data set size. But both the APK and especially the PPK mostly lead to better models than the RBF does with the full dataset of 100 molecules. Where the APK and PPK differ in the way that using the APK leads to models which have a lower RMSD to the active structures but a higher MSE while the use of the PPK leads to very good models with the lowest MSE of all models created but with higher RMSD values.

Because most of the resulting models using the APK and PPK with the reduced datasets were as good as the ones using the full dataset due to their equal or lower MSE, I decided to use the reduced dataset for future runs. Since the SVR is contained in $O(n^3)$ and the GA is contained in $O(n)$ this measure cut the computation time for a complete run by at least 50%.

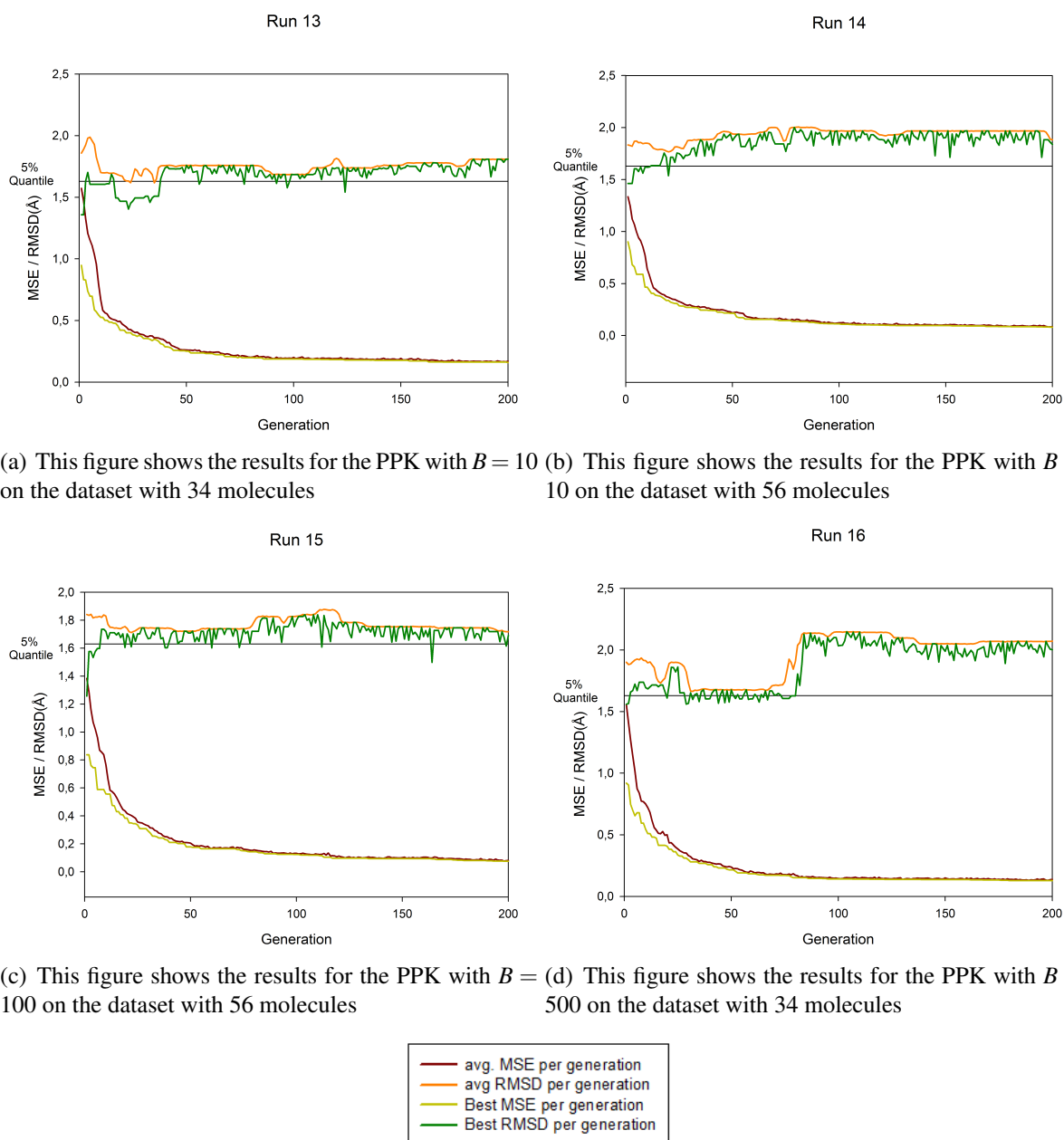


Figure 4.4: These figures show the results of the runs with reduced datasets and altered parameters for the PPK.

Parameter / Run Nr.	13	14	15	16
Kernel method	PPK	PPK	PPK	PPK
RDF B factor	10	10	100	500
RBF Sigma factor	-	-	-	-
Smoothing factor	-	-	-	-
Mutation Probability	0.1	0.1	0.1	0.1
Mutate first 12 only	no	no	no	no
Conformation Sampling	intermediate	intermediate	intermediate	intermediate
Dataset size	34	56	56	34
Start avg. MSE	1.5727	1.3344	1.3834	1.5559
End avg. MSE	0.1701	0.0853	0.0813	0.1387
Diff. Start/End MSE	1.4026	1.2491	1.3021	1.4172
Best avg. MSE	0.1645	0.0838	0.0777	0.1306
Best individual MSE	0.1595	0.0814	0.0763	0.1285
Start avg. RMSD	1.8585	1.8323	1.8402	1.9009
End avg. RMSD	1.8107	1.8826	1.7181	2.0711
Diff. Start/End RMSD	0.0478	-0.0503	0.1221	-0.1702
Best avg. RMSD	1.6172	1.7662	1.7118	1.6644
Best individual RMSD	1.4992	1.3194	1.0926	1.6265

Table 4.4: This table shows the parameters and the results for run 13, run 14, run 15 and run 16. Parameters denoted by ‘-’ are not available for the chosen kernel method

4.1.4 Alternative Parameters for the Product Probability Kernel

In addition to reducing the dataset I changed the parameters of the PPK and APK. The results for the PPK with the B parameter of the RBF set to 10, 100 and 500 in relation to 1000 at the previous runs are shown in figure 4.4 and table 4.4. One can see that run 16 still shows the low RMSD values around generation 50 with a strong increase and stagnation afterwards. The runs 13, 14 and 15 also show the decrease of the RMSD around generation 50 but not as strong as runs with a higher parameter.

The average RMSD values of runs 13 and 15 only decrease slightly by 0.0478 and 0.1221 from 1.8585 and 1.8107 to 1.8107 and 1.7181. While the average RMSD values of runs 14 and 16 even increase by 0.0503 and 0.1702 from 1.8323 and 1.9009 to 1.8826 and 2.0711. The MSE though reaches the lowest values of all runs with run 15 at a value of 0.777 and the second lowest at run 14 with 0.0838.

The fact that a run with parameter $B = 10$ renders the best resulting model can be lead back to the fact that the B parameter describes the ‘smoothness’ and distinctness of a RDF. With declining B the RDF becomes more of a general description of the respective molecule and its conformation instead of an exact characterization. In this case the presence of distinct chemical groups or pharmacophores and their arrangement to each other is more important than their individual orientation. This leads to a better generalization of the model at the cost of a better discrimination of the conformations for each molecule.

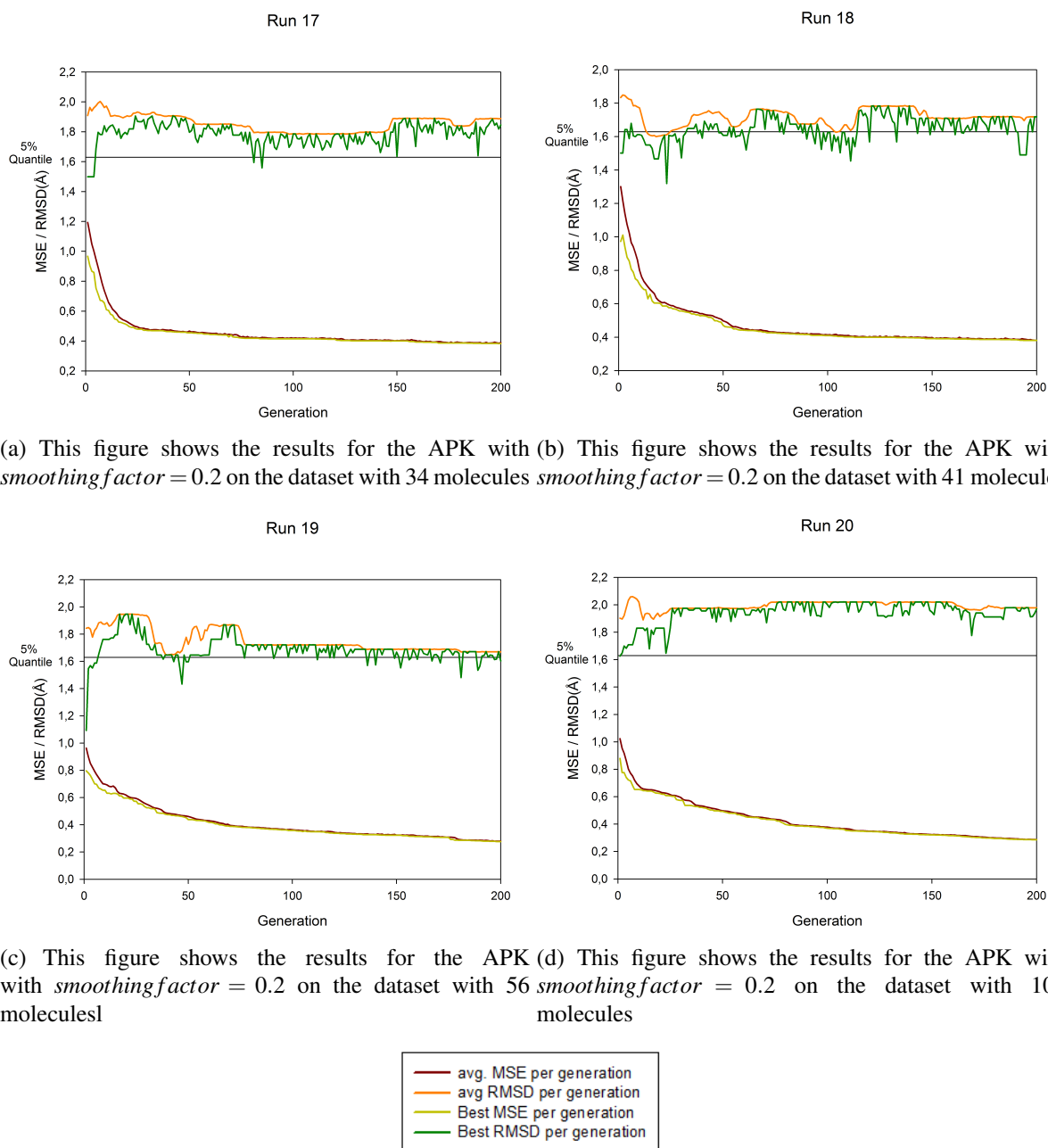


Figure 4.5: These figures show the results of the runs with reduced datasets and altered parameters for the APK.

Parameter / Run Nr.	17	18	19	20
Kernel method	APK	APK	APK	APK
RDF B factor	-	-	-	-
RBF Sigma factor	-	-	-	-
Smoothing factor	0.2	0.2	0.2	0.2
Mutation Probability	0.1	0.1	0.1	0.1
Mutate first 12 only	no	no	no	no
Conformation Sampling	intermediate	intermediate	intermediate	intermediate
Dataset size	34	41	56	100
Start avg. MSE	1.1934	1.3013	0.9636	1.0234
End avg. MSE	0.3901	0.3813	0.2805	0.2878
Diff. Start/End RMSD	0.8033	0.92	0.6831	0.7356
Best avg. MSE	0.3857	0.3813	0.2796	0.2878
Best individual MSE	0.3830	0.3807	0.2774	0.2864
Start avg. RMSD	1.9084	1.8337	1.8434	1.9025
End avg. RMSD	1.8879	1.7185	1.6696	1.9787
Diff. Start/End RMSD	0.0205	0.1152	0.1738	-0.0762
Best avg. RMSD	1.7840	1.5997	1.6470	1.8900
Best individual RMSD	1.4992	1.3194	1.0926	1.6265

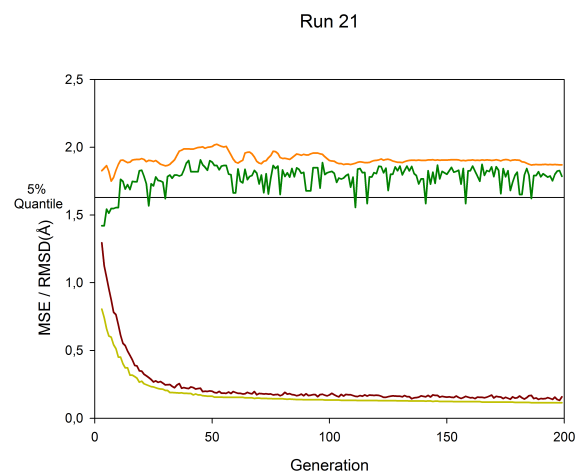
Table 4.5: This table shows the parameters and the results for run 17, run 18 and run 19 and run 20. Parameters denoted by ‘-’ are not available for the chosen kernel method

4.1.5 Alternative Parameters for APK

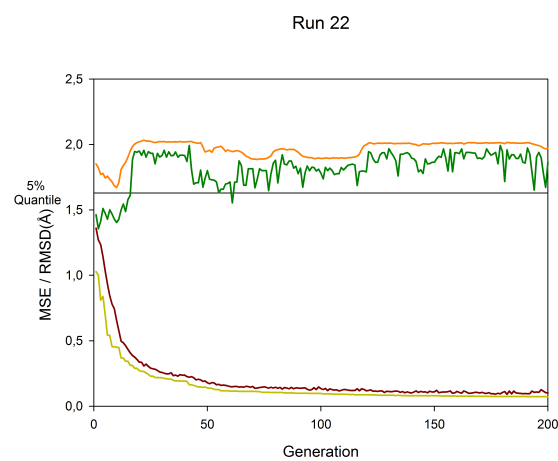
Figure 4.5 and table 4.5 show the results for the runs 17, 18, 19 and 20 using the APK with the full and the reduced datasets and a smoothing factor of 0.2. Runs 18 and 19 show the first deep decline of the RMSD at 50 generations to a global minimum of an average RMSD of 1.5997 and 1.6470 while run 17 shows a steady decline and run 20 an overall stagnation at a RMSD of approximately 2.0.

As with the alternation of the B parameter of the PPK, setting the smoothing factor to a value of 0.2 for the APK changes the generalization of the model resulting in lower MSE values than previous runs with the use of the APK for all four runs. While the APK only encodes atom types, distances and binding modes, doubling the smoothing factor still holds enough information to fit the model. It allows further for the GA to hold more individuals with a wider RMSD range. This can be seen in figure 4.5 with the best individual RMSD values being distinctively low than the average RMSD values over several generations in all four runs.

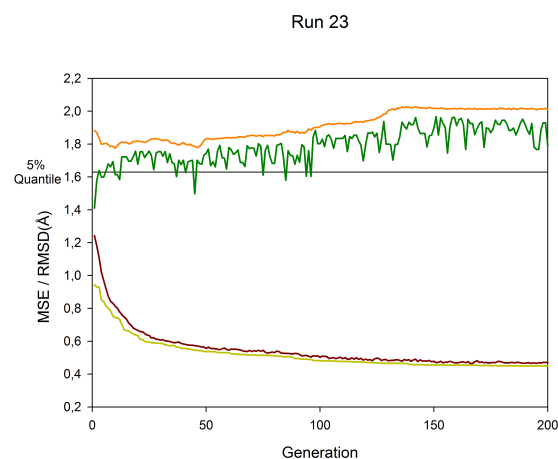
The average MSE values of the final models was 0.3901, 0.3813, 0.2805 and 0.2878, which is approximately 0.15 below previous runs. But in change for the better generalization and lower MSE values the overall RMSD stagnated with only run 19 showing a decline to a finale value of 1.7185 which is still above the 5% quantile.



(a) This figure shows the results for the PPK with $B = 10$ on 56 molecules and a mutation probability of 0.2



(b) This figure shows the results for the PPK with $B = 10$ on 56 molecules and a mutation probability of 0.2



(c) This figure shows the results for the PPK with $B = 10$ on 56 molecules and a mutation probability of 0.2

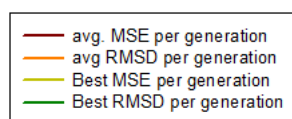


Figure 4.6: These figures show the results of the runs with reduced datasets and increased mutation probability for the PPK and APK.

Parameter / Run Nr.	21	22	23
Kernel method	PPK	PPK	APK
RDF B factor	10	10	-
RBF Sigma factor	-	-	-
Smoothing factor	-	-	0.1
Mutation Probability	0.5	0.5	0.5
Mutate first 12 only	no	no	no
Conformation Sampling	intermediate	intermediate	intermediate
Dataset size	34	56	34
Start avg. MSE	1.5839	1.3608	1.2421
End avg. MSE	0.1584	0.1012	0.4708
Diff. Start/End MSE	1.4255	1.2596	0.7713
Best avg. MSE	0.132	0.0909	0.4651
Best individual MSE	0.1148	0.0736	0.4497
Start avg. RMSD	1.8889	1.8520	1.8823
End avg. RMSD	1.8699	1.9711	2.0111
Diff. Start/End RMSD	0.0190	-0.1191	-0.1288
Best avg. RMSD	1.7506	1.6712	1.7752
Best individual RMSD	1.4206	1.3559	1.4107

Table 4.6: This table shows the parameters and the results for Run 21, Run 22 and Run 23. Parameters denoted by ‘-’ are not available for the chosen kernel method

4.1.6 Increased Mutation Rate

Studying the results of the changes in decreasing the dataset and altering the kernel parameter the next step was to change the parameters and overall process of the GA. The easier of both was to set the mutation probability to 0.5 instead of the standard 0.1 value. The mutation probability defines the rate at which mutations occur during the mating process from one generation to the next. Increasing the mutation probability allows the GA to search in a broader range and increases the chance of the optimization to jump out of a local minimum, but it also decreases the optimization rate and may lead to more diverse results.

As one can see in all three runs depicted in figure 4.6 the increased mutation probability leads to at least one individual in each generation with a significantly lower RMSD as the average. Further noticeable is the fact the the progression of the average and especially the best individual RMSD include more peaks.

While changing the mutation probability still leads to good models with an average MSE of 0.1584, 0.1012 and 0.4708, in two of the runs the average end RMSD was even 0.1191 and 0.1288 higher then their start RMSD with 1.8520 and 1.8823.

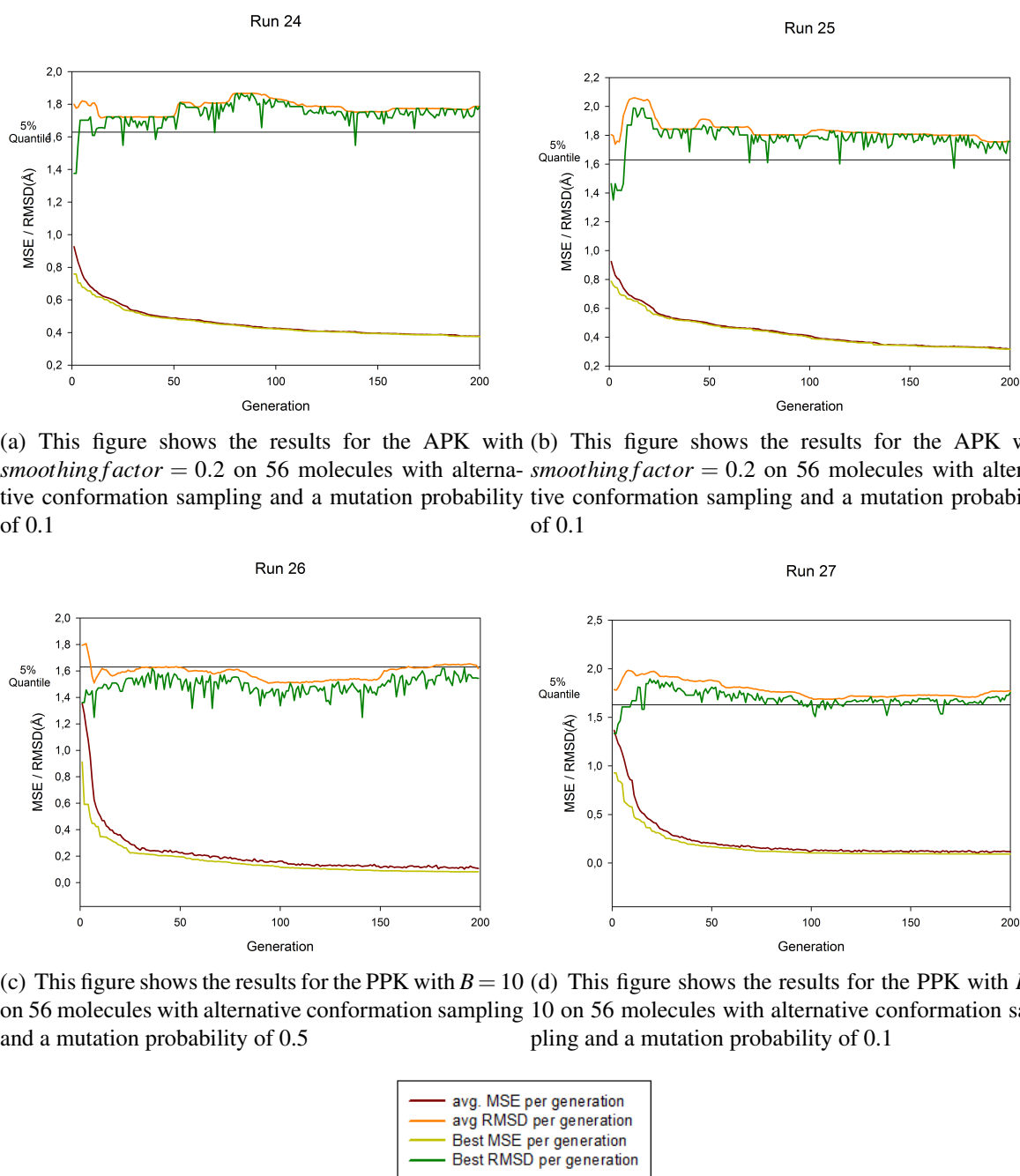


Figure 4.7: These figures show the results of the runs with reduced datasets of the alternative conformation sampling and increased mutation probability for the PPK.

4.1.7 Alternative Conformation Sampling

A second way of increasing the chance of the optimization to jump out of a local minimum was to change the conformation sampling of the dataset. While the intermediate parameters for the ConfGen algorithm only allows local minima of relative molecular energy the comprehensive parameters allowed GonfGen to output molecules with dihedral angles and flexible ring energies not being in a local minimum.

Parameter / Run Nr.	24	25	26	27
Kernel method	APK	APK	PPK	PPK
RDF B factor	-	-	10	10
RBF Sigma factor	-	-	-	-
Smoothing factor	0.2	0.2	-	-
Mutation Probability	0.1	0.1	0.5	0.5
Mutate first 12 only	no	no	no	no
Dataset size	56	56	56	56
Conformation Sampling	comprehensive	comprehensive	comprehensive	comprehensive
Start avg. MSE	0.9265	0.9252	1.3531	1.3671
End avg. MSE	0.3775	0.3192	0.1084	0.1191
Diff. Start/End MSE	0.549	0.606	1.2447	1.248
Best avg. MSE	0.3775	0.3192	0.1025	0.1059
Best individual MSE	0.3756	0.3169	0.0816	0.0934
Start avg. RMSD	1.7999	1.8047	1.794	1.7862
End avg. RMSD	1.7873	1.7608	1.6181	1.7728
Diff. Start/End RMSD	0.0126	0.0439	0.1759	0.0134
Best avg. RMSD	1.7175	1.7385	1.5084	1.6867
Best individual RMSD	1.376	1.3512	1.2488	1.3333

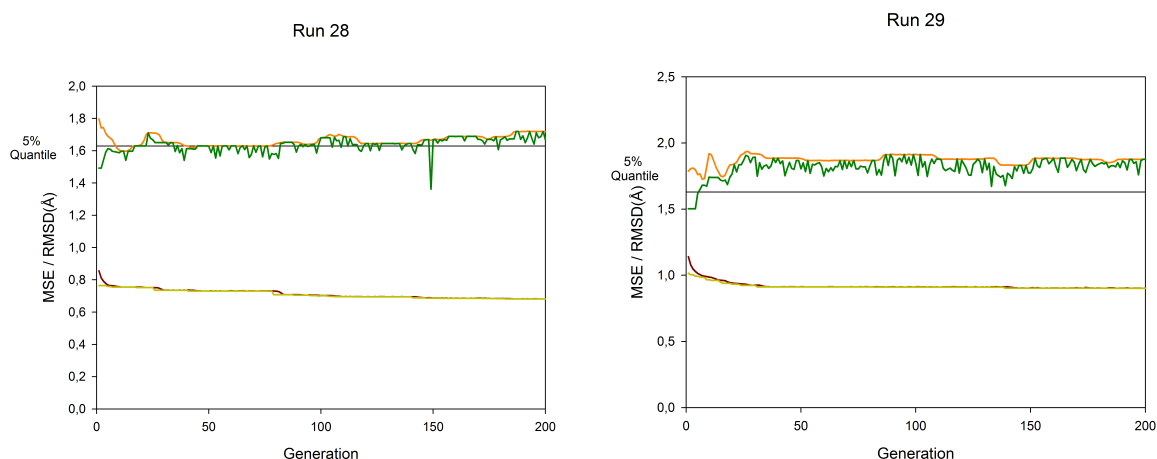
Table 4.7: This table shows the parameters and the results for Run 24, Run 25, Run 26 and Run 27. Parameters denoted by ‘-’ are not available for the chosen kernel method

The thought was to allow the GA to successively get out of local minima due to the differences in relative molecular energies not being as great as with the intermediate conformation sampling. Therefore the error of a change from one conformation to the one with the nearest relative energy would not be as great for the comprehensive conformation sampling as for the intermediate.

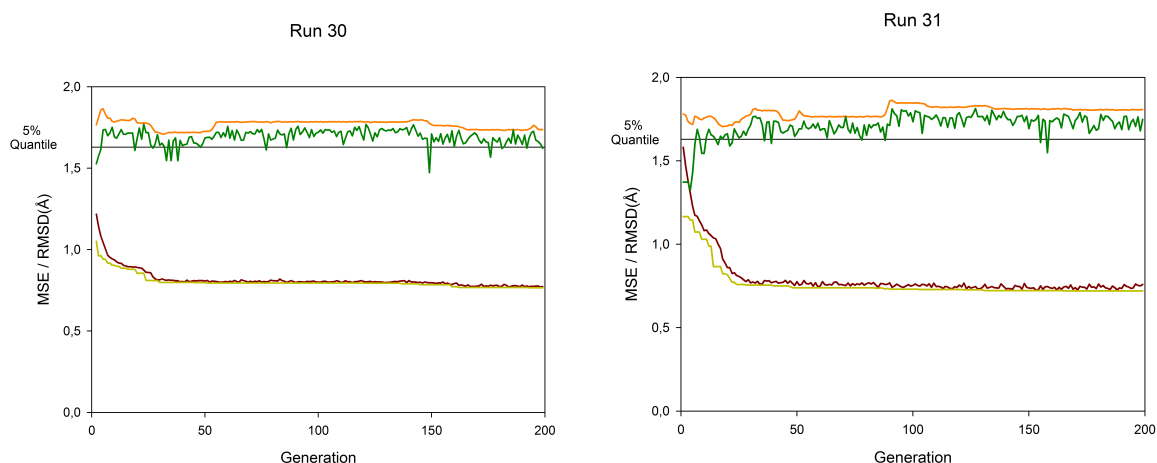
Another reason for using the comprehensive conformation sampling was that the relative energy of an active structure is not necessarily a local minimum due to the interaction of the molecule with its target and the solvent. So by allowing non-minima structures in the dataset I reduced the minimal RMSD between the conformers in the dataset and the active structures.

The results for the experiments with the dataset produced by conformation sampling with comprehensive parameters are shown in figure 4.7 and table 4.7. As one can see in run 24 and run 25 which used the APK and a smoothing factor of 0.2 the first decline of the average RMSD with its concurrent increase between generation 35 and 50 still occurs. But instead of stagnating at the same average level as in most previous runs the average RMSD declines again in later generations. The final average RMSD, however, was only 0.0126 and 0.0439 lower than the starting average RMSD with 1,783 and 1.7608 while the final average MSE with 0.3775 and 0.3192 was better than most of the previous runs with the APK. Therefore the final models were more precise but still did not include conformations near the active structure.

The results for run 26 and run 27 are also shown in figure 4.1.6 and table 4.7. Both runs used the PPK and a mutation probability of 0.5. In run 26 the average RMSD almost always lies within the 5% quantile. In run 27 the average RMSD declines to a value of approximately 1.7 and stagnates for the second half of the optimization. The runs have final average RMSD values of 1.6181 and 1.7728 and final average MSE values of 0.1084 and 0.1191.



(a) This figure shows the results for the APK with $smoothingfactor = 0.1$ on 56 molecules with alternative conformation sampling and a the mutation only allowed on the 12 known active structures (b) This figure shows the results for the APK with $smoothingfactor = 0.1$ on 34 molecules with alternative conformation sampling and a the mutation only allowed on the 12 known active structures



(c) This figure shows the results for the PPK with $B = 10$ on 56 molecules with alternative conformation sampling and a the mutation only allowed on the 12 known active structures (d) This figure shows the results for the PPK with $B = 10$ on 34 molecules with alternative conformation sampling and a the mutation only allowed on the 12 known active structures

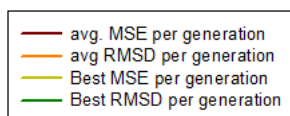


Figure 4.8: These figures show the results of the runs with reduced datasets of the alternative conformation sampling and altered mutation operator to allow mutation only on the conformers of the molecules with known active structure.

Parameter / Run Nr.	28	29	30	31
Kernel method	APK	APK	PPK	PPK
RDF B factor	-	-	10	10
RBF Sigma factor	-	-	-	-
Smoothing factor	0.1	0.1	-	-
Mutation Probability	0.1	0.1	0.1	0.1
Mutate first 12 only	yes	yes	yes	yes
Dataset size	56	34	56	34
Conformation Sampling	comprehensive	comprehensive	comprehensive	comprehensive
Start avg. MSE	0.856	1.1405	1.311	1.5813
End avg. MSE	0.682	0.9029	0.7738	0.7589
Diff. Start/End MSE	0.174	0.2376	0.5372	0.8224
Best avg. MSE	0.6815	0.9022	0.7706	0.7255
Best individual MSE	0.6815	0.9022	0.7656	0.7207
Start avg. RMSD	1.7972	1.7881	1.7808	1.7808
End avg. RMSD	1.7187	1.8774	1.7369	1.8078
Diff. Start/End RMSD	0.0785	-0.0893	0.0439	-0.027
Best avg. RMSD	1.5975	1.727	1.7098	1.7052
Best individual RMSD	1.3621	1.5028	1.3915	1.3247

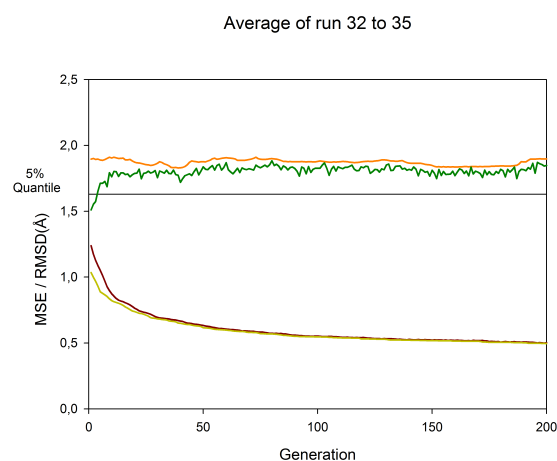
Table 4.8: This table shows the parameters and the results for Run 28, Run 29, Run 30 and Run 31. Parameters denoted by ‘-’ are not available for the chosen kernel method

4.1.8 Alternative Mutation Operator

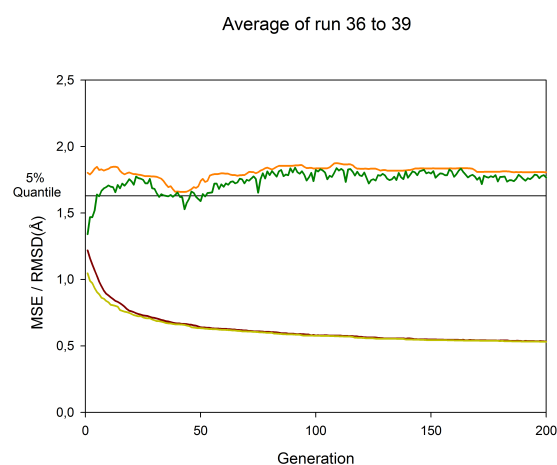
The final change to the mutation was to change the mutation operator in that way that it only allowed the conformers of the molecules with known active structures to be mutated during the mating process at the end of a generation. For the rest of the molecules, which are the ones with unknown active structures, the conformation with the minimal relative energy was fixed. The reason for this change was to reduce the search space for the optimization to the conformations of the molecules with known active structure. Therefore increasing the chance of finding a model with low MSE which included conformations similar to the active structures resulting in a lower RMSD.

The results of the four runs, 28, 29, 30 and 31 with altered mutation operator are shown in figure 4.8 and table 4.8. One can see that, while the average MSE rapidly declines in the first 25 generations the average RMSD remains at the same level throughout the whole run for all four runs. Further the average MSE only reaches values of 0.684 to 0.9029 which is significantly higher than in previous runs due to the fact that the remaining fixed molecules do not allow a better model.

This means that, while only optimizing over the generated conformers of the known active structures, the best models found still do not include conformations similar (i.e. with a low RMSD) to those structures. Possible reasons for that are manifold and will be discussed in the next chapter.



(a) This figure shows the average results for four runs with the APK with *smoothing factor* = 0.1 on conformers of 34 molecules created with the intermediate parameter set



(b) This figure shows the average results for four runs with the APK with *smoothing factor* = 0.1 on conformers of 34 molecules created with the comprehensive parameter set

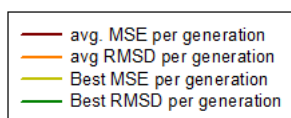


Figure 4.9: These figures show the average results of four runs with the APK and reduced datasets of both conformation sampling parameter sets.

Parameter / Run Nr.	avg. of run 32-35	avg. of run 36-39
Kernel method	APK	APK
RDF B factor	-	-
RBF Sigma factor	-	-
Smoothing factor	0.1	0.1
Mutation Probability	0.1	0.1
Mutate first 12 only	no	no
Dataset size	34	34
Conformation Sampling	intermediate	comprehensive
Start avg. MSE	1.2388	1.2199
End avg. MSE	0.5018	0.5338
Diff. Start/End MSE	0.7370	0.6862
Best avg. MSE	0.4999	0.5338
Best individual MSE	0.4965	0.5316
Start avg. RMSD	1.8967	1.8022
End avg. RMSD	1.8979	1.8070
Diff. Start/End RMSD	-0.0012	-0.0048
Best avg. RMSD	1.8294	1.6579
Best individual RMSD	1.5099	1.3405

Table 4.9: This table shows the parameters and the results for the average of runs 32-35 and 36-39. Parameters denoted by ‘-’ are not available for the chosen kernel method

4.1.9 Reruns

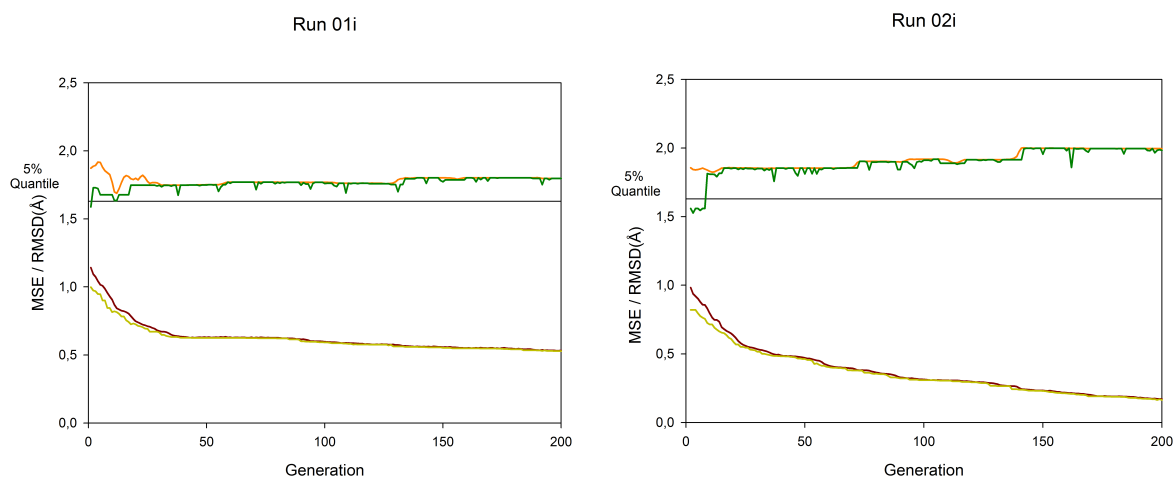
The only run resulting in an considerably lower average RMSD then all other runs was run 12 with a final average RMSD of 1.3813 (see figure 4.3 and table 4.3). To check if this was a random result or if the constellation of kernel, dataset and parameters lead to models using conformations with a low RMSD to the active structure I reran the specific parameter set of run 12 four times with either of both conformation sampling parameters intermediate and comprehensive. The averaged results of these runs are shown in figure 4.9 and table 4.9

As one can see the average RMSD stagnates at 1.8 which is also the mean value for the RMSD of all possible combinations of conformers. The decline and immediate return to the mean RMSD between generation 25 and 50 is also visible for both results.

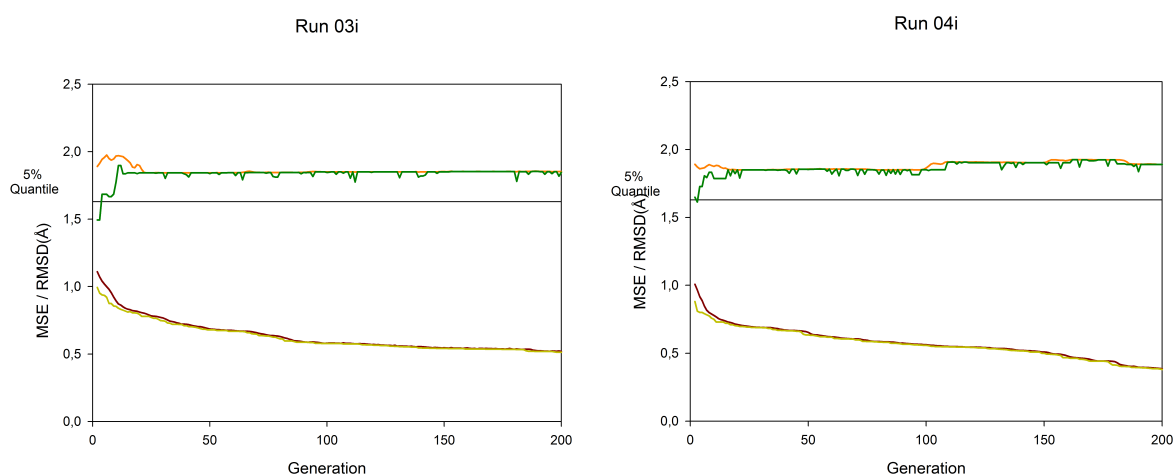
This proves that run 12 was a random result with the GA finding a local minimum. With a value of 0.567 the MSE of run 12 is even higher then the average MSE for both of the 4 runs with 0.5018 and 0.5338.

4.2 Implicit Conformation Sampling

The runs of the optimization with the implicit conformation sampling were done parallel to the runs with precomputed conformation sampling. Therefore the results of the runs with precomputed conformation sampling influenced the decisions made for the parameters and dataset size for the runs with implicit conformation sampling. One run with implicit conformation sampling on the full dataset took up to two weeks on a Xeon quadcore server. This is why there are fewer results for the implicit conformation sampling.



(a) This figure shows the results for the RBF kernel with $B = 1000$ and $\sigma = 100$ on the full dataset with implicit conformation sampling (b) This figure shows the results for the RBF with $B = 1000$ on the full dataset with implicit conformation sampling



(c) This figure shows the results for the PPK with $B = 10$ on the full dataset with implicit conformation sampling (d) This figure shows the results for the PPK with $B = 10$ on the full dataset with implicit conformation sampling

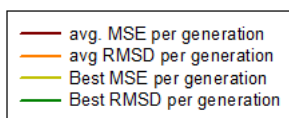


Figure 4.10: These figures show the results of the runs with implicit conformation sampling on the full dataset and the use of the PPK and RBF kernel.

Parameter / Run Nr.	01i	02i	03i	04i
Kernel method	RBF	PPK	PPK	PPK
RDF B factor	1000	10	10	10
RBF Sigma factor	100	-	-	-
Smoothing factor	-	-	-	-
Mutation Probability	0.1	0.1	0.1	0.1
Mutate first 12 only	no	no	no	no
Conformation Sampling	implicit	implicit	implicit	implicit
Dataset size	100	100	100	100
Start avg. MSE	1.1411	1.0375	1.1491	1.0389
End avg. MSE	0.5331	0.171	0.519	0.3874
Diff. Start/End MSE	0.608	0.8665	0.6301	0.6515
Best avg. MSE	0.5311	0.171	0.519	0.3874
Best individual MSE	0.5285	0.1638	0.5142	0.3808
Start avg. RMSD	1.8724	1.8826	1.9039	1.8894
End avg. RMSD	1.7978	1.9955	1.8523	1.8909
Diff Start/End RMSD	0.0746	-0.1129	0.0516	-0.0015
Best avg. RMSD	1.6873	1.8246	1.8425	1.8498
Best individual RMSD	1.5876	1.5268	1.4942	1.6142

Table 4.10: This table shows the parameters and the results for run 01i, run 02i, run 03i and run 04i. Parameters denoted by ‘-’ are not available for the chosen kernel method

4.2.1 Initial Runs

The results for the initial runs, 01i and 02i, are shown in figure 4.10 and table 4.10. Run 03i and run 04i are later runs with the same parameters as run 02i. As one can see the average MSE is decreasing. This shows that the optimization is functional. But in comparison to the runs with precomputed conformation sampling shown in the preceding section the average MSE decreases more slowly and has not reached a minimum at the end of the run. This can be assumed due to the average MSE still decreasing in generation 150 to 200 and not reaching an even level. The final average MSE of the runs was 0.5331, 0.1710, 0.5190 and 0.3874. This is the range of the results for the average MSE from the runs with precomputed conformation sampling.

Further noticeable is the fact that the average RMSD shows almost no change after generation 50 in all four runs stagnating for many generations. The mean RMSD over all generations of all four runs is 1.859, which is near the overall mean of 1.81 of all possible conformations. In addition to the best individual RMSD this can be explained by the small chance of a mutation occurring at the rotatable bonds of the molecules with known active structure. This low chance of a mutation is a result of the molecules with known active structure having fewer rotational bonds than the molecules without known active structure. Therefore the chance of a mutation occurring on a molecule without known active structure is increased in relation to the runs with precomputed conformation sampling where the mutation chances were equally distributed.

In addition all four runs lack the initial decrease of the average RMSD between generation 25 and 50 seen in the runs with precomputed conformation sampling.

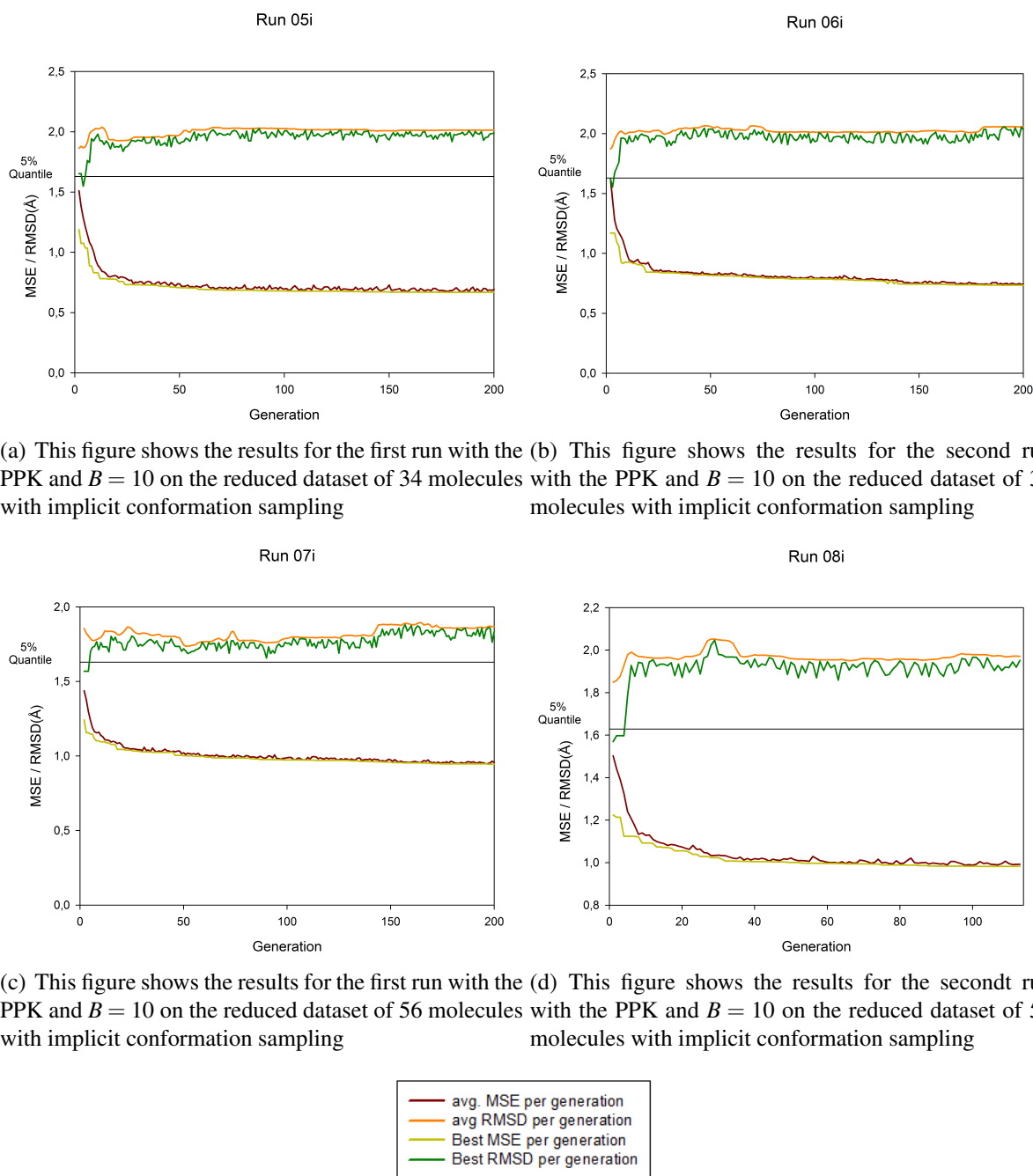


Figure 4.11: These figures show the results of the runs with reduced datasets, implicit conformation sampling and the use of the PPK.

Parameter / Run Nr.	05i	06i	07i	08i
Kernel method	PPK	PPK	PPK	PPK
RDF B factor	10	10	10	10
RBF Sigma factor	-	-	-	-
Smoothing factor	-	-	-	-
Mutation Probability	0.1	0.1	0.1	0.1
Mutate first 12 only	yes	yes	yes	yes
Conformation Sampling	implicit	implicit	implicit	implicit
Dataset size	100	100	100	100
Start avg. MSE	1,687	1,73	1,5033	1,5079
End avg. MSE	0,6932	0,7425	0,9929	0,9555
Diff. Start/End MSE	0,9938	0,9875	0,5104	0,5524
Best avg. MSE	0,673	0,7398	0,989	0,9475
Best individual MSE	0,6693	0,7352	0,9827	0,9459
Start avg. RMSD	1,8592	1,8605	1,8487	1,8546
End avg. RMSD	2,0149	2,0571	1,9715	1,8685
Diff Start/End RMSD	-0,1557	-0,1966	-0,1228	-0,0139
Best avg. RMSD	1,8592	1,8605	1,8487	1,7359
Best individual RMSD	1,5489	1,5543	1,5698	1,5687

Table 4.11: This table shows the parameters and the results for Run 04 through Run 09. Parameters denoted by ‘-’ are not available for the chosen kernel method

4.2.2 Reduced Dataset and Fixed Conformation

In run 05 to run 08 I combined the several changes. First I reduced the dataset to 56 and 34 molecules including the 12 with known active structure. The second change was to fix the conformation of the molecules with unknown active structure to the conformation with the lowest relative energy and allowing mutation only at rotational bonds of the molecules with known active conformation. Run 07 was interrupted at generation 116 due to a server crash and could not be resumed.

The results for the four runs with this configuration are shown in figure 4.11 and table 4.11. One can see that the average MSE declines faster (within the first 50 generation) than in the previous runs with implicit conformation sampling to a final value of 0.6932, 0.7425, 0.9938 and 0.9555. These higher average MSE values can be explained by the fixed conformations and the resulting missing possibility for optimization.

One can see the effect of allowing mutation only on the 12 molecules with known conformation. The best individual RMSD is lower than the average RMSD for almost every generation in all four runs. This can be explained by the higher mutation rate resulting in individuals with conformations with a lower RMSD to the active structures.

5 Discussion

The hypothesis that the best achievable models to predict activity include the active structures of the training molecules, can not be confirmed in this work. The average RMSD over all runs is almost exactly the average RMSD over all possible sets of conformations to the active structure (See figure 5.1). While some runs show a RMSD below the 5% quantile they still are within the normal distribution and are countered by the runs with a RMSD above average. Though models were found with a low average RMSD the optimization in most cases returned to models with an RMSD near the average value. The reasons for these results can come from two directions. They can be either chemically or mathematically qualified.

One possible reason is, that too many factors determining the active structure are missing from the models I created. For example the solvent of the molecules, in this case water, is not included in the model at all. But as studies have shown the solvent often has a great influence on the activity and the active structure of a molecule. It can change the molecule's conformation to a more fitting one or even be part of the active site itself by filling the space not occupied by the ligand. Therefore disregarding the solvent may lead to a model with a *feature space* not having enough information about the active complex.

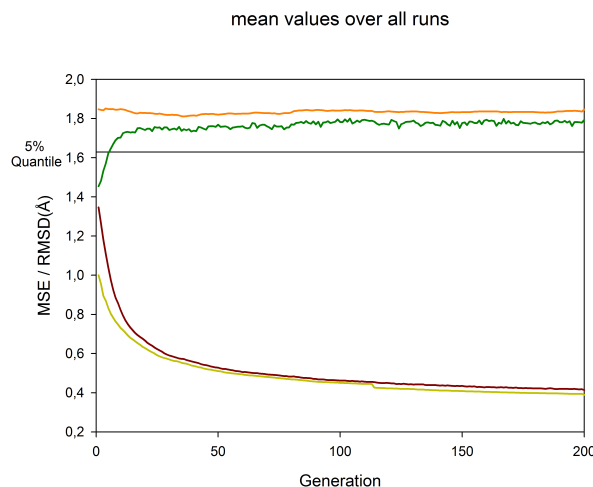


Figure 5.1: This figure shows the diagram for the mean values over all runs. One can clearly see the average RMSD stagnating at 1.81 over the whole run while the MSE declines to a value of about 0.4

A second reason may be that only part of the molecule's conformations are critical for the activity while another, possibly larger part can take up a random, probably energetically minimal, conformation. However this would only account for a part of the normal distribution of the RMSD values. The 'active' part of the molecule's conformations chosen for the model would have a distinctly lower average RMSD to the active conformation resulting in the overall RMSD being lower than mean of the normal distribution. For the dataset used in this work, this can be ruled out because all 12 molecules with known active structure are entirely integrated in the active process and have no parts which can take on free conformations.

Another reason for the model consisting of conformations with an average RMSD to the active structure may come from the used kernel methods in combination with the molecules in the dataset. Regardless of the chemical properties the molecules in the dataset often consisted of several ring systems. These ring systems contribute the same partial results to all kernel values for all different combinations of conformations due to the fact that they are rigid and do not change partial results between different conformations. To rule this out one would have to repeat the experiments with a dataset of molecules with less rigid parts or with a kernel method that prioritizes longer distances within the conformations.

The most important and apparent reason though is based on the principle of the SVR. On optimizing over the activity prediction with the activity value being the same for each conformer of a given molecule, it is clear that to achieve a maximal generalization the process will pick that conformation with the best representation of the whole conformational space of the molecule. In the conformational hypersphere this will be a conformation near the center of the sphere. Which in this case means a conformation with a maximal ‘similarity’ to all other conformations of that molecule. Or in other words, a conformation with a minimal average RMSD to all other conformers. In most cases this would not be the active conformation. For the 12 molecules with known active structure I used in this work, the average distance (i.e. RMSD) from the active conformation to the center of the conformational hypersphere was 1.81Å.

Not regarding the resulting average RMSD values, the PPK kernel yielded the best MSE with values as low to 0.01 in contrast to the RDF kernel with MSE values only in range of 0.5 and the APK with MSE values in the range of 0.3. Furthermore the PPK had the steepest decent of the MSE values reaching an almost even level at generation 25-50, whereas the APK needed more generations. The resulting models of the implicit conformation sampling were less significant because in most runs the optimization process was not finished. This can be explained by the mutation probability only being set to 0.1 and the number of points being about 10 times as much as with the precomputed conformation sampling. One can see that by reducing the possible mutations as in the final runs with implicit conformation sampling and fixed conformers the MSE also decreases more rapidly.

6 Prospects

Following the results of this work models for activity prediction provide the best results not by using the active structures but a conformation with minimal distance to all possible conformations of the respective molecules. To confirm these results one would have to run further experiments with other kernels and data sets. In these experiments one would have to calculate the distances of the best resulting conformations for activity prediction not only to the active conformation but to all possible conformation, or at least an equally distributed set over the conformational space. These new model would be expected to provide the best results if they were based on these ‘average’ conformations and not on the active structures.

If proven correct one would have to rethink the use of active conformations in 3D QSAR models in favor of more generalized conformations. Further this would suggest a method of finding a conformation near the center of the conformational hypersphere without calculating all pairwise distances of all possible conformations.

If proven wrong one would have to revise the results of this work and further investigate the reasons for the average RMSD of the best achievable models constantly being the exact RMSD of the active conformations to the middle of the proposed conformational hypersphere. Therefore one would have to compile a new set of molecules with known active structures. Where the set would include more and diverse active conformations to cover a wider range of the chemspace.

Bibliography

- [Bau09] L.; Heine-A.; Smolinski M.; Hangauer D.; Klebe G. Baum, B.; Muley. Think twice: understanding the high potency of bis(phenyl)methane inhibitors of thrombin. *J.Biol.Mol*, 391:552–564, 2009.
- [Ber77] T.F.; Williams-G.J. Meyer E.E. Jr.; Brice M.D.; Rodgers J.R.; Kennard O.; Shimanouchi T.; Tasumi M. Bernstein, F.C.; Koetzle. The protein data bank: A computer-based archival file for macromolecular structures. *J. of Mol. Biol.*, 112:535, 1977.
- [BGV92] B.E. Boser, I.M. Guyon, and Vapnik V.N. *Annual Workshop on Computational Learning Theory*, chapter Proceedings of the fifth annual workshop on Computational learning theory, pages 144–152. ACM, 1992.
- [Boe99] J.; Klebe G Boehm, M.;Stuerzebecher. Three-dimensional qantitive structure activity relationship analyses using comparative molecular file analysis and comparative molecular similarity indices analysis to elucidate selectivity differences of inhibitors binding to trypsin, thrombin, and factor xa. *Journal of Medical Chemistry*, 42:458–477, 1999.
- [Cha05] Q Chang. Scaling gaussian rbf kernel width to improve svm classification. *Neural Networks and Brain, 2005. ICNN&B '05. International Conference on*, pages 19–22, 2005.
- [CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [Cou04] Chaok; Dill Ken A. Coutsiias, Evangelos A.; Seok. Using quaternions usings rmsd. *J.Comput. Chem*, 25:1849–1857, 2004.
- [Cou05] Chaok; Dill Ken A. Coutsiias, Evangelos A.; Seok. Rotational superposition and least sequares: the svd and quaternions approach yield identical results. reply to the preceeding comment by g. kneller. *J. Comput. Chem.*, 26:1663–1665, 2005.
- [CV95] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
- [Dia76] R. Diamond. On the comparison of conformations using linear and quadratic transformations. *Acta Cryst*, 32:1–10, 1976.
- [Dix06] A.; Knoll E.; Rao S.; Schaw D. Friesner R.A. Dixon, S.; Smondryrev. Phase: a new engine for pharmacophore perception, 3d qsar model developement and 3d database screening. *J. Comput.-Aided Mol. Design*, 20(10):647–671, 2006.

- [Eki04] S. Ekins. Predicting undesirable drug interactions with promiscuous proteins in silico. *Drug Discovery Today*, 9:276–285, 2004.
- [Fis94] E. Fischer. Einfluss der konfiguration auf die wirkung der enzyme. *Berichte der deutschen chemischen Gesellschaft*, 27:2985–2993, 1894.
- [Fle89] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, New York, 1989.
- [Fri04] J.L.; Murphy R.B.; Halgren T.A.; Klicic J.J.; Mainz D.T.; Repasky M.P.; Knoll E.H. Shelley M.; Perry J.K.; shaw D.E.; Francis P.; Shenkin P.S. Friesner, R.A.; Banks. Glide: a new approach for rapid, accurate docking and scoring. 1- method and assessment of docking accuracy. *J.Med.Chem*, 47(7):1739–49, 2004.
- [Gas96] J.; Schuur J.; Selzer P.; Steinhauer L.; Steinhauer V. Gasteiger, J.; Sadowski. Chemical information in 3d space. *J. Chem. Inf. Comput. Sci.*, 36:1030–1037, 1996.
- [Gas97] J.; Selzer P.; Steinhauer L.; Steinhauer V. Gasteiger, J.; Schuur. Finding the 3d structure of a molecule in its ir spectrum. *Fresenius J. Anal. Chem.*, 359:50–55, 1997.
- [Guy93] B.; Vapnik V.N. Guyon, I.; Boser. *Advances in Neural Information Processing Systems*, chapter Automatic capacity tuning of very large VC-dimension classifiers, pages 147–155. Morgan Kaufmann, San Mateo, CA, 1993.
- [Ham66] Sir. Hamilton, William Rowan. *Elements of Quaternions*. Longmans, Green & Co., London, 1866.
- [Han69] C. Hansch. A quantitative approach to biochemical structure-activity relationships. *Acc. Chem. Res.*, 2:232–239, 1969.
- [Har94] George K.; Kauffman Louis H. Hart, John C.; Francis. Visualizing quaternion rotation. *Transactions on Graphics*, 13:256–276, 1994.
- [Hem99] Markus C.; Steinhauer V.; Gasteiger J. Hemmer. Deriving the 3d structure of organic molecules from their infrared spectra. *Vibrational Spectroscopy*, 19:151–164, 1999.
- [HG04] H.Z. Hao and M. Genton. Compactly supported radial basis function kernels. 2004.
- [Hol75] John H. Holland. *Adaptation in Natural and Artificial Systems*. Univ. Michigan Press., 1975.
- [Jah09] G.; Fechner N.; Zell A. Jahn, A.; Hinselmann. Optimal assignment methods for ligand-based virtual screening. *Journal of Chemoinformatics*, 1:14, 2009.
- [Jeb04] R.; Howard A. Jebara, T.; Kondor. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.
- [Jor88] J.T. Jorgensen, T.L.; Tirado-Rives. The opls potential functions for proteins. energy minimization for crystals of cyclic peptides and crambin. *J.Am.Chem.Soc.*, 110:165, 1988.

- [Jor96] D.S.; Tirado-Rives J. Jorgensen, W.L.; Maxwell. Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. *J.Am.Chem.Soc.*, 118:11225–11235, 1996.
- [JZ10] G.; Fechner-N.; Hennekes C. Jahn, A.; Hinselmann and A. Zell. Probabilistic modeling of conformational space for 3d machine learning approaches. *Mol. Inf.*, 29:441–455, 2010.
- [Kab76] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, 32(5)A:922–923, 1976.
- [Kat00] R.; Luong-C.; Radika K.; Martelli A.; Sprengeler P.A.; Wang J.; Chan H.; Wong L. Katz, B.A.; Mackman. Structural basis for selectivity of a small molecule, s1-binding, submicromolar inhibitor of urokinase-type plasminogen activator. *Chem.Biol.*, 7:299–312, 2000.
- [Kat01a] K.; Luong-C.; Rice M.J.; Mackman R.L.; Sprengeler P.A.; Spencer J.; Hataye J.; Janc J.; Link J.; Litvak J.; Rai R.; Rice K.; Sideris S.; Verner E.; Young W. Katz, B.A.; Elrod. A novel serine protease inhibition motif involving a multi-centered short hydrogen bonding network at the active site. *J.Biol.Mol*, 307:1451–1486, 2001.
- [Kat01b] P.A.; Luong-C.; Verner E.; Elrod K.; Kirtley M.; Janc J.; Spencer J.R.; Breitenbucher J.G.; Hui H.; McGee D.; Allen D.; Martelli A.; Mackman R.L. Katz, B.A.; Sprengeler. Engineering inhibitors highly selective for the s1 sites of ser190 trypsin-like serine protease drug targets. *Chem.Biol.*, 8:1107–1121, 2001.
- [Kat03] K.; Verner-E.; Mackman R.L.; Luong C.; Shrader W.D.; Sendzik M.; Spencer J.R.; Sprengeler P.A.; Kolesnikov A.; Tai V.W.-F.; Hui H.C.; Breitenbucher J.G.; Allen D.; Janc J.W. Katz, B.A.; Elrod. Elaborate manifold of short hydrogen bond arrays mediating binding of active site-directed serine protease inhibitors. *J.Biol.Mol*, 329:93–120, 2003.
- [Kat04] C.; Ho-J.D.; Somoza J.R.; Gjerstad E.; Tang J.; Williams S.R.; Verner E.; Mackman R.L.; Young W.B.; Sprengeler P.A.; Chan H.; Mortara K.; Janc J.W.; McGrath M.E. Katz, B.A.; Luong. Dissecting and designing inhibitor selectivity determinants at the s1 site using an artificial ala190 protease (ala190 upa). *J.Biol.Mol*, 344:527–547, 2004.
- [Kea89] Simon K. Kearsley. On the orthogonal transformation used for structural comparison. *Acta Crystallographica*, 45(2)A:208–210, 1989.
- [Kel06] P.; Schalon-C.; Bret G.; Foata N.; Rognan D. Kellenberg, E.; Muller. sc-pdb: an annotated database of druggable binding sites from the protein data bank. *Journal of Chemical Information and Modeling*, 46(2):717–727, 2006.
- [Kos58] Jr. Koshland, D. E. Application of a theory of enzyme specificity to protein synthesis. *Proc. Natl. Acad. Sci. U.S.A.*, 44:98–104, 1958.
- [Kos94] Jr. Koshland, D. E. The key and lock theory and the induced fit theory. *Angew.Chem.Int.Ed.Engl*, 33:2375–2378, 1994.

- [Mac84] A. L. Mackay. Quaternion transformation of molecular orientation. *Acta Crystallographica Section A*, 40(2):165–166, Mar 1984.
- [Mat96] R.; Costanzo-M.J.; Maryanoff B.E.; Tulinsky A Matthews, J.H.; Krishnan. Crystal structures of thrombin with thiazole-containing inhibitors: probes of the sl' binding site. *Biophys.J.*, 71:2830–2839, 1996.
- [McL72] A.D. McLachlan. A mathematical procedure for superimposing atomic coordinates of proteins. *ActaCryst*, 28:656–657, 1972.
- [OW91] T.I. Oprea and C.L. Walter. *Reviews in Computational Chemistry*, chapter Theoretical and practical aspects of three-dimensional quantitative structure-activity relationships, pages 127–182. Wiley-VCH: New York, 1991.
- [Sad94] J. Sadowski, J.; Wagener M.; Gasteiger. Corina: Automatic generation of high-quality 3d-molecular models for application in qsar. In *10th European Symposium on Structure-Activity Relationships: QSAR and Molecular Modelling*, 1994.
- [Sch96] P.; Gasteiger J Schuur, J.H.; Selzer. The coding of the three-dimensional structure of molecules by molecular transforms and its application to structure - spectra correlations and studies of biological activity. *J. Chem. Inf. Comput. Sci.*, 36:334–344, 1996.
- [Sel97] J.H.; Gasteiger Selzer, P.; Schuur. *Software Development in Chemistry 10*, volume 10, chapter Simulation of IR Spectra with Neural Networks Using the 3D-MORSE Code, page 293. Gesellschaft Deutscher Chemiker: Frankfurt am Main, 1997.
- [Sew07] Martin Sewell. Kernel methods. Technical report, Department of Computer Science University College London, 2007.
- [Sho85] K. Shoemaker. Animating rotation with quaternion curves. *Comput. Graph.*, 19:245–254, 1985.
- [SI08a] New York Schroedinger Inc. *LigPrep*, V2.1. 2008.
- [SI08b] New York Schroedinger Inc. *MacroModel*, V9.6. 2008.
- [SJ93] M. Stone and P. Jonathan. Statistical thinking and techniques for qsar related studies. 1 general theory. *J. Chemom.*, 7:455–475, 1993.
- [SOW04] Jeffrey J. Sutherland, Lee A. O'Brian, and Donald F. Weaver. A comparison of methods for modeling quantitative structure-activity relationship. *J. Med. Chem.*, 47:5541–5554, 2004.
- [Ste03] Y.; Kuhn S.; Horlacher O.; Luttmann E.; Willighagen E. Steinbeck, C.; Han. The chemistry development kit (cdk): an open source java library for chemo- and bioinformatics. *J Chem Inf Comput Sci*, 43(2):493–500, 2003.
- [Ste06] C.; Kuhn S.; Floris M.; Guha R. Steinbeck, C.; Hoppe. Recent development of the chemistry development kit (cdk) - an open source library for chemo- and bioinformatics. *Curr Pharm Des*, 12(17):2111–2120, 2006.

-
- [Vap82] V. Vapnik. *Estimation of dependencies bade on empirical data*. Springer Verlag, 1982.
- [Vap95] V. Vapnik. *The Nature of Statistical Learning Theroy*. Springer Verlag, 1995.
- [VC64] V. Vapnik and A. Chervenonkis. A note on one class perceptrons. *Automation and Remote Control*, 25, 1964.
- [VC74] V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition*. Nauka (Russia), 1974.
- [VL63] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, 1963.
- [WS10] P.; Murphy R.B.; Sherman W.; Friesner R.A. Watts, K.S.; Dalal and J.C. Shelley. Confgen: A conformational search method for efficient gerneration of bioactive conformers. *J.Chem.Inf.Model.*, 50:534–546, 2010.
- [XG02] Y.; Ming L. Xi, C.;Lin and K. Gilson. The binding database: data management and interface design. *Bioinformatics*, 18(1):130–139, 2002.
- [Zam76] A. Zamora. An algorithm for finding the smallest set of smalles rings. *J.Chem.Inf.Comput.Sci.*, 16(1):40–43, 1976.

List of Figures

1.1	Overlay of Thrombin inhibitors	2
1.2	QSAR process	2
2.1	SVR	6
2.2	GA individuals	10
2.3	GA mutation operators	11
2.4	Thrombin-Hirudin complex	12
3.1	flowchart of the overall process	14
3.2	examples for RDF	15
3.3	overlay of RDF functions	17
3.4	curve approximation	18
3.5	Atom Pair Kernel	20
3.6	12 structures with known active conformation	22
3.7	example for implicit conformation sampling encoding	25
3.8	exapmle of rotatable bonds	26
4.1	results of the initial runs	30
4.2	results of reduced dataset with ppk and rbf	32
4.3	results of reduced dataset with APK	34
4.4	results for alternative parameters for the PPK	36
4.5	results for alternative parameters for the APK	38
4.6	results of increased mutation rate	40
4.7	results of alternative conformation sampling	42
4.8	results of alternative mutation operator	44
4.9	results for the reruns	46
4.10	results for initial runs with implicit conformation sampling	48
4.11	results for runs with implicit conformation sampling, reduced dataset and fixed conformation	50
5.1	mean over all runs	52

List of Tables

3.1	table of compiled structures	23
3.2	table of parameters for conformer generation	24
3.3	table of parameters for the SVR	27
4.1	results of the initial runs	31
4.2	results of reduced dataset with ppk and rbf	33
4.3	results of reduced dataset with APK	35
4.4	results for alternative parameters for the PPK	37
4.5	results for alternative parameters for the APK	39
4.6	results of increased mutation rate	41
4.7	results of alternative conformation sampling	43
4.8	results of alternative mutation operator	45
4.9	results for the reruns	47
4.10	results for initial runs with implicit conformation sampling	49
4.11	results for runs with implicit conformation sampling, reduced dataset and fixed conformation	51