Freie Universität Berlin
Deptartment of Mathematics and Computer Science
and
Max Planck Institute for Molecular Genetics
Department of Computational Molecular Biology

Efficient Computation of Probe Qualities

Master's Thesis

by
Christoph Hafemeister
May 21, 2008

Supervising tutors:
Dr. Alexander Schliep
Prof. Dr. Knut Reinert

Contents

Chapter 1

Introduction

Biological Background

Understanding gene regulation is a fundamental problem in molecular biology. DNA microarrays allow us to infer gene expression by measuring the concentrations of thousands of mRNAs in parallel, because of that, they have become one of the most widely used tools in molecular biology. They constitute a high-throughput technology that can also be used for a variety of further applications, for example comparative genomic hybridization, SNP detection and the detection of agents in a sample.

An oligonucleotide DNA microarray consists of a glas or plastic slide with thousands of short (20 - 75 base pairs long) single stranded DNA sequences, called probes, attached to it. During a microarray experiment, a sample containing DNA, or in rare cases RNA, molecules, are given onto the microarray. Due to Watson-Crick base pairing, probes and sample DNA can form duplexes. Due to the fact that the samples have been labeled, the amount of duplexes formed can be measured, and because probe composition and position on the microarray is known, the presence and relative concentration of DNA containing the reverse complement of the probe sequence can be estimated. Abstractly, we can talk about target DNA as specific loci in a genome or set of genomes, or specific regions such as transcripts.

When detecting target hybridization to a probe we want to make sure that we measure the concentration of the target in the genome. This can only be done reliably, when a probe hybridizes to the intended target in the genome exclusively. This leads to the problem of finding unique probes – probes only hybridizing to their intended targets and not to other spots in the genome. A probe which also binds to other than its intended target is said to cross-hybridize. For example, in Figure 1.1 p_5 cross-hybridizes to t_2.

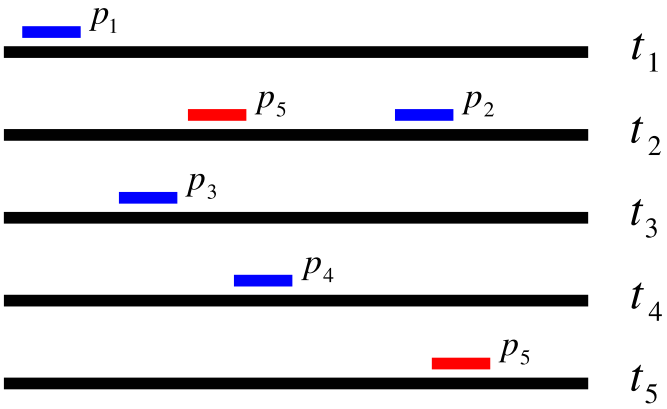

Figure 1.1: The probe selection problem. Probes $p_1, \ldots, p_4$ can be used to measure the relative concentration of targets $t_1, \ldots, t_4$. The probe p_5 is a non-unique probe, t_2 and t_5 will bind to it, and from signal intensity the concentration of t_5 cannot be infered.

The Thesis

When designing a DNA microarray for a given set of targets, it is desirable to pick only unique probes, as to be able to unambiguously decode the results. In order to find a set of unique probes for a set of targets, the hybridization process has to be modeled, and for every probe, hybridization to not intended targets has to be ruled out. Algorithms exist to model free energy of hybridization of probes and targets, computing the most stable of all possible duplex structures, but these are computationally very costly und not feasible for large genomes. Instead, the most widely used approach to detect cross-hybridization is a BLAST search for the probe in the given genome. The number of matched base pairs and the length of the longest matched stretch is then used to infer hybridization.

In this work we present a novel probe selection algorithm, which makes use of the nearest neighbor thermodynamic model for estimating an upper bound of the free energy associated with DNA duplex formation. In order to qickly scan a genome for cross-hybridization for a given probe set, we propose a filtering method to reduce the search space for each probe.

The nearest neighbor model assumes that the stability of a nucleic acid duplex depends on the identity and orientation of neighboring base pairs. It estimates the free energy of a duplex as the sum of free energy contributions by the stacked pairs. We make use of an efficient alignment algorithm which maximizes the nearest neighbor stability, and use this as an upper bound for the

true stability of a duplex. Thus, we take the thermodynamic characteristics of duplex formation more accurately into account, as opposed to simply counting matches and mismatches. Figure 1.2 shows stacking effects for an example sequence.

Scanning through the whole genome to find regions where cross-hybridization occurrs for a given probe is a time consuming procedure. In this work we propose a filter using a seed and extend approach to reduce the search space for each probe. Based on the nearest neighbor model used for our alignment algorithm, we use seeds with a minimum stability contribution. Q-gram indices for probes and targets are used to instantly find matching probes and sequence positions, where the nearest neighbor alignment is then used to compute the lower bound on hybridization free energy.

In our experiments we test the performance of our nearest neighbor alignment and the resulting energy bound, and compare it to simpler heuristics based on the edit distance. We also test the performance of the presented filter for various paramters and datasets.

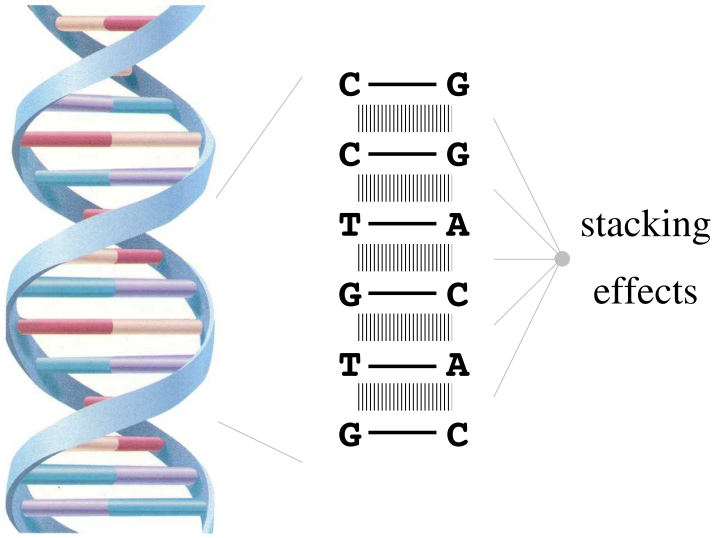

Figure 1.2: Pairing of Watson-Crick bases and base stacking effects between adjacent pairs in a DNA duplex. Adapted from Anatomy & Physiology: The Unity of Form and Function by Kenneth S. Saladin.

1.1 Related Work

The probe selection problem has been studied for more than ten years now. During this time, many approaches have been taken to efficiently calculate probe specificity. Probably the most common way is to use BLAST to do a search for regions in the genome highly similar to the probe, and then use the Hamming distance as specificity measure. For example, this method is used by Reymond et al. (2004) and Rouillard et al. (2002). Other approaches still use the Hamming or edit distance, but do not use BLAST to find regions of interest. Sung & Lee (2003) use gapped hashing and the pigeon hole principle to quickly find regions with low Hamming distance to the probe. Rouillard et al. (2003) do not use the Hamming distance, they compute the best thermodynamic alignment using dynamic programming. As this is a very costly computation, they combine it with a previous BLAST search to reduce the number of computations. Rahmann (2003) uses the length of the longest common factor of probe and target as specificity measure, and finds those quickly with the use of a suffix array. A similar approach is taken by Gräf et al. (2007), they define a uniqueness score based on minimum unique substrings of of probe and target. However, their implementation using suffix arrays needs a large amount of RAM. Another approach by Kaderali & Schliep (2002) uses a heuristic for melting temperature calculation using suffix trees and dynamic programming.

When the target sequences are highly similar, such as different virus subtypes, it is often impossible to find a unique probe for every target. Rash & Gusfield (2002) were among the first ones to address this problem by using suffix trees and integer-linear programming to find a set of probes which uniquely identifies a target. A different approach that also takes cross-hybridization and experimental error into account was presented by Schliep et al. (2003) and later extended to use integer-linear programming and a branch-and-cut algorithm to select a minimal probe set (Klau et al. 2004). When targets are related by a phylogenetic tree and non-unique probes are present, Schliep & Rahmann (2006) present a decoding approach based on a Bayesian framework.

Chapter 2

Basics

2.1 DNA Structure

To understand the hybridization processes during microarray experiments it is necessary to consider basic facts about the structure of DNA.

More than 50 years ago, Franklin, Crick, Watson and Wilkins discovered the structure of DNA, showing that it consists of two antiparallel complementary strands which form a double-helix (Watson & Crick 1953, Wilkins et al. 1953). Each strand arises from the directional polymerization of single nucleotide units. A nucleotide unit consists of one of the four bases, adenine (A), thymine (T), cytosine (C) or guanine (G), a sugar (deoxyribose) and a phosphate.

Esterification reactions between the sugar's C3' hydroxyl group and the phosphate of an incoming nucleoside triphosphate (NTP) form the links between the nucleotides. A phosphodiester group links the 5'C of one deoxyribose to the 3'C of the next sugar, resulting in a phosphodiester-sugar backbone with 5'-3' linkages. As DNA transcription can only proceed in 5'-3' direction it is customary to look at a strand in this direction.

In a double-stranded DNA molecule, the two strands are *Watson-Crick complements*; an A in one strand pairs with a T in the other, in the same way a C pairs with a G, and vice versa.

2.1.1 Higher Order DNA Structure

When two complementary DNA strands form a duplex, they take on the characteristical structure of a double helix. This is an energetically favourable state, because the phosphates and sugars, which carry polarized bonds and are thus

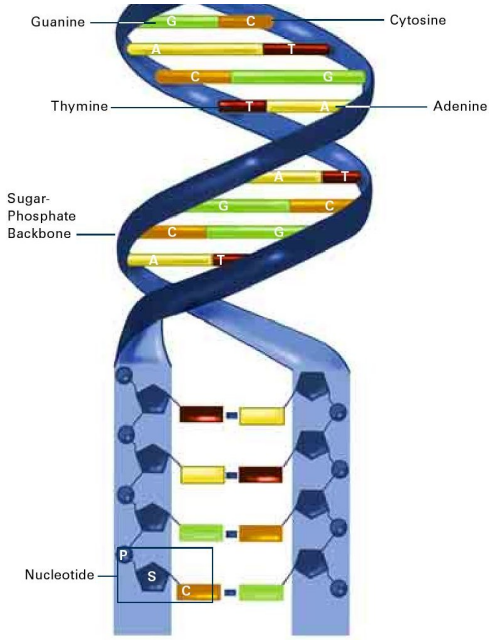

Figure 2.1: A DNA double helix. Image taken from U.S. Department of Health and Human Services, NIH Publication No. 07-662.

hydrophilic in nature, are on the outside. The bases, which are hydrophobic due to the few polarized bonds, are buried within the inside, allowing hydrophobic interactions between them.

2.1.2 Base Pairing and Base Stacking

The highly specific base pairing results from the formation of hydrogen bonds between pairs of nucleotides. These bonds are highly directional, because the interacting atoms must lie in an approximately straight line for strong bonding. The cytosine-guanine coupling is stronger, incorporating three hydrogen bonds, than the adenine-thymine coupling, with only two hydrogen bonds.

The hydrogen bonds between bases are responsible for the specificity of the base interactions, but most of the stability of a DNA duplex arises from interactions that result from base stacking. The interactions of aromatic molecules in a parallel arrangement are called stacking. These aromatic interactions are caused by intermolecular overlapping of p-orbitals in π-conjugated systems. In DNA, the bases of the nucleotides are made from either purine or pyrimidine, containing aromatic rings. The DNA backbone is tilted by an angle of 30°, bringing the planar rings of adjacent base pairs to a position where they lie

vertically one above the other. The non-covalent bonds between the bases, resulting from overlapping p-orbitals, are weaker than covalent bonds, but the sum of all π stacking interactions within the double stranded DNA molecule has the largest contribution to the overall free energy associated with the duplex.

2.2 Microarray Technology

2.2.1 DNA Microarrays

DNA microarrays (also calles DNA chips) are arrays of thousands DNA molecules on a quartz, glass or nylon substrate. The main principle of microarray experiments is the hybridization of two nucleic acid strands. DNA sequences in solution (target sequences, *targets*) are given onto the chip and bind to specific immobilized DNA molecules (probe sequences, *probes*) on the surface of the microarray. Probes of the same type are placed at regularly spaced and well defined locations called *spots* or *features*. The targets, which often are extracted from a tissue or blood sample, are labeled with a fluorescent or radioactive dye and are then allowed to hybridize to the probes on the array. Unhybridized targets are then washed off, and the amount of hybridized targets at a probe spot can be measured by the intensity of the dye or radioactivity. This measurement then reflects the relative concentration of the target sequence in the sample. Because of the high density with which the probes can be attached to the chip surface, this method allows the detection of vast amounts of different target sequences in parallel.

The idea behind this procedure is to quantify the presence and concentration of certain DNA sequences in a sample. Depending on probe and target type, this can have different applications. For gene expression profiling, the mRNA of a cell or tissue is used directly as the target, or is converted to cDNA using reverse transciptase and amplified. Each probe is specifically designed to only bind a target from a certain gene, and thus, the result represents the expression level of that gene. These experiments can be used to compare different state of cells or compare treated and not treated cells. The genomic DNA present in a sample can also be used directly to create the targets. For the application of genetic screening, the DNA is fragmented and used directly as target set. With carefully designed probes, this approach can be used to investigate the presence of microorganisms, viruses and their types, and genetic mutations. Another application is a tiling array. Here, the arrays are composed of a large number of probes from a contiguous region of the genome, selected, so that they are immediately adjacent to, or overlap, one another. This way, transcribed regions

outside any known annotation can be discovered (Mockler et al. 2005).

2.2.2 Probe Design Problem

When designing microarrays, special care has to be taken that the probes satisfy fundamental probe selection criteria: (1) Quantitative criteria; (2) Homogeneity; (3) Sensitivity; (4) Specificity.

Quantitative criteria as described by Lockhart et al. (1996): (1) the content of any single base does not exceed 50% of the probe size; (2) the length of any contiguous `A` and `T` or `C` and `G` region is less than 25% of the probe size; (3) GC content is between 40% and 60%

Homogeneity criterion requires that all probes must react (hybridize) under the same reaction conditions. This is a direct consequence of the paallelization of the hybridization of thousands of probe target pairs. The reaction behaviour is mainly influenced by the salt content, the pH-value and the hybridization temperature.

Sensitivity criterion (or hybridization efficiency) filters out probes with low true positive detection. A low sensitivity can be caused by probes which may fold back on themselves instead of binding its target.

Specificity criterion filters out probes which could possibly hybridize to others than the intended target, which would result in false positives. This is not only the case, when copies of the intended target are present at multiple locations in the genome, but can also be the result of the presence of very similar targets, which, despite having mismatches with the probe sequence, can still hybridize to the probe.

The specificity criterion is the computationally most expensive one to check. In order to validate if a probe could bind to a not intended target, also called cross-hybridize, all targets have to be taken into consideration and binding stability has to be computed. The next section will introduce models that are used to compute the stability of a DNA duplex.

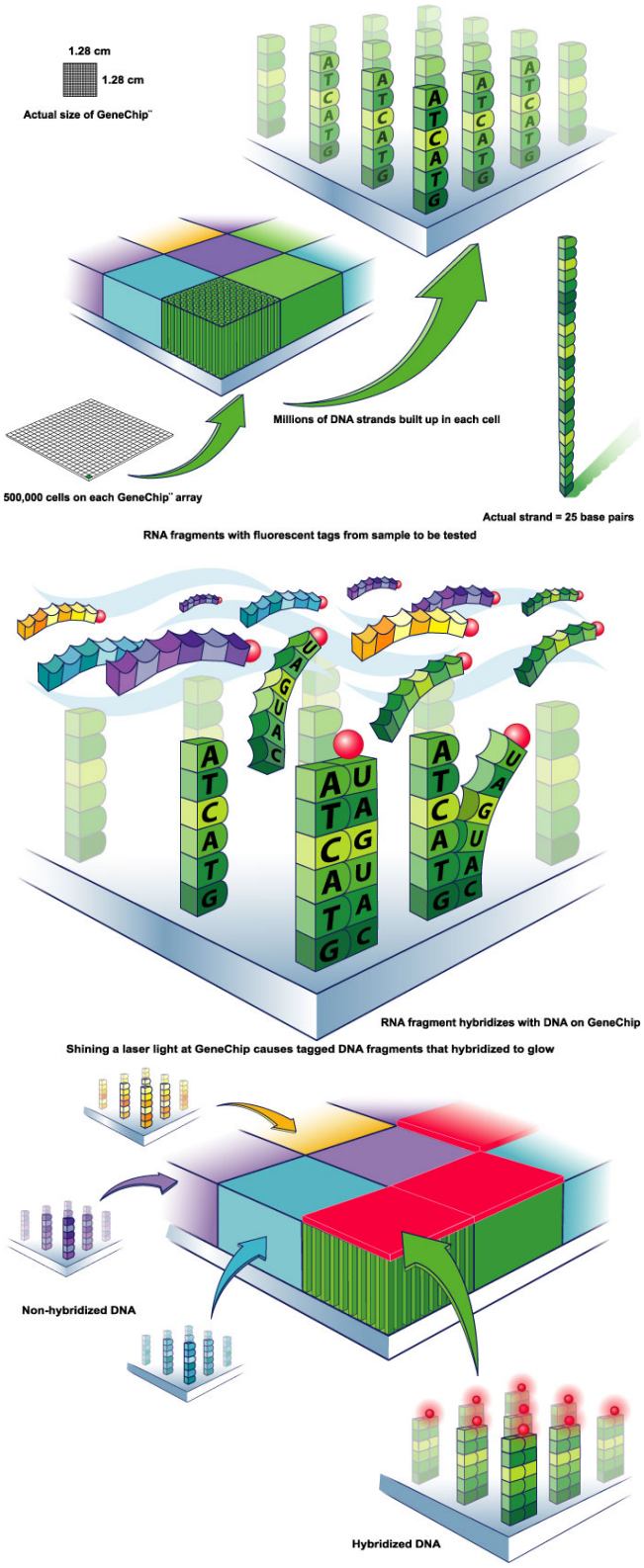

Figure 2.2: Principle of microarray experiments shown on the example of an Affymetrix GeneChip. GeneChip is a registered trademark of Affymetrix Inc., Santa Clara, California. Images taken from www.affymetrix.com.

2.3 Models of DNA Hybridization

The duplex formation of nucleic acids is the foundation of all microarray experiments. The stability of the probe-target hybridization is an important factor for the design and interpretation of the experiments. In this section we give a brief overview over the currently used thermodynamic models being used to predict the stability of nucleic acids duplexes expressed as Gibbs free energy.

2.3.1 Basic Notions of Thermodynamics

The stability of a double-stranded DNA duplex or an DNA-RNA hybrid can be expressed quantitavely in terms of the Gibbs free energy ΔG. Gibbs free energy and fundamental concepts of physical chemistry and heat theory are introduced in the following paragraphs. For a more detailed introduction the reader is referred to the sources of this section (Müller 2001, Alberty & Silbey 1992, Kaderali 2001).

In thermodynamics, the total kinetic, chemical and potential energy of a system's molecules is called the *internal energy* U of the system. Internal energy, heat, and work are all measured in the same unit, the Joule (J), which is defined as $1\ J = 1\ kg\ m^2\ s^{-1}$. It has been found experimentally that the internal energy of a system may be changed either by doing work on the system or by heating it. The *first law of thermodynamics* states that if heat and work are done to a system, then the change in internal energy is given by the formula

$$\Delta U = q + w, \tag{2.3.1}$$

where q is heat transferred to the system and w work done to it. Assuming that only the volume V of a system is changed while the pressure P is constant, w equals $-P\Delta V$. The change in the internal energy of the system is then

$$\Delta U = q_p - P\Delta V, \tag{2.3.2}$$

where q_p is the heat for the constant pressure process. Considering a transition between two states A and B with an internal energy change $\Delta U = U_B - U_A$ and change in volume $\Delta V = V_B - V_A$, it follows from Equation 2.3.2 that the heat absorbed is

$$q_p = (U_B + PV_B) - (U_A + PV_A). \tag{2.3.3}$$

As U, p and V are state functions (i.e. they do not depend on the history of the system), we define a new state function, the *enthalpy*.

Definition 2.3.1 (Enthalpy) Enthalpy H is defined as $H := U + PV$, where U is the internal energy, P is the pressure exerted on the system, and V its volume.

Thus, $q_p = H_B - H_A$, or, in other words, the heat absorbed in a process at constant pressure is equal to the change in enthalpy if the only work done to the system is reversible volume-pressure work (Kaderali 2001). $\Delta H = \Delta U + P\Delta V$ describes the change of the enthalpy of a system for a finite change of temperature.

Enthalpy describes the state of a system, but it does not give any information about the direction of a process (Kaderali 2001). The concept of *entropy* captures this aspect. It is strongly connected with the *second law of thermodynamics*: Chemical reactions proceed spontaneously in the direction that corresponds to an increase in the disorder of the universe. This means that chemical reactions proceed in the direction that converts free energy (energy that is available to do work) into heat. Thus, transitions between two states A and B will proceed in the direction A $\rightarrow$ B, if the change in free energy associated with it is negative.

For infinitesimally small changes in energy, Equation 2.3.2 can be rewritten as $dU = dq + dq$. For reversible transitions this means,

$$dU = dq_{rev} + dw_{rev}, \tag{2.3.4}$$

where

$$dw_{rev} = -PdV, or \frac{dw_{rev}}{-P} = dV. \tag{2.3.5}$$

Therefore, dU can be written in the form

$$dU = -PdV + dq_{rev} \tag{2.3.6}$$

The intensity factor for heat transitions is temperature T, therefore dq_{rev} can be rewritten as

$$\frac{dq_{rev}}{T} = dS, \tag{2.3.7}$$

where S is a state function called entropy. Boltzman examined the probability that gas is in a given state, giving a statistical view of entropy. Amongst others, Moore (1998) describes his work. Without going in to more detail, we use his results to define S.

Definition 2.3.2 (Entropy) Entropy S is defined as $S := R \ln W$, where R is the Boltzmann constant, and W the number of possibilities to realize the state under consideration.

The entropy is a measure of molecular disorder of a system. Systems prefer states with much freedom, thus have the tendency to maximize entropy. Substituting TdS for dq_{rev} in Equation 2.3.6 yields

$$dU = -PdV + TdS, \tag{2.3.8}$$

which combines the first and second law of thermodynamics into one equation (Kaderali 2001).

Here, it is important to note that the statement $\frac{dq_{rev}}{T} = dS$ holds only in the case of a reversible process, i.e. a process that is at chemical equilibrium. For irreversible (spontaneous) processes, dS will be larger than dq/T, while a process with $dS < dq/T$ is impossible. Hence, for irreversible processes

$$dU \leq -PdV + TdS \tag{2.3.9}$$

holds (Kaderali 2001).

Gibbs energy. Gibbs energy G is the Legendre transformation of the enthalpy of a system held at constant temperature T and constant pressure P, defined by $G \equiv H - TS$. Therefore, it follows from Equations 2.3.8 and 2.3.9 and $H = U + PV$, that

$$dG = dH - TdS - SdT \leq VdP - SdT. \tag{2.3.10}$$

Thus at constant temperature and pressure, chemical reactions are spontaneous in the direction of decreasing Gibbs energy $dG \leq 0$.

The *standard Gibbs energy of reaction* is defined in terms of the standard reaction enthalpy and entropy by

$$\Delta_r G^\circ = \Delta_r H^\circ - T\Delta_r S^\circ. \tag{2.3.11}$$

It is the difference in standard molar Gibbs energies G_m° of the products and reactants in their standard states at the temperature of the reaction. From now on we will use the short forms ΔG, ΔH and ΔS in this work.

Intuitevely, ΔG can be interpreted as a difference in stability between the products and the reactants.

2.3.2 Nearest Neighbor Model

The nearest neighbor model estimates the Gibbs energy change for perfect Watson-Crick DNA duplex or RNA/DNA-hybrid formation. It is based on the early work of Zimm (Crothers & Zimm 1964) and Tinoco and Coworkers (Devoe

& Tinoco 1962, Borer et al. 1974). It assumes that the stability of a nucleic acid duplex depends on the identity and orientation of neighboring base pairs. This assumption is justified by the structure of DNA and the base interaction in double strands, see section 2.1.2. The nearest neighbor model expresses ΔG for a duplex of n base pairs as a sum of $n-1$ terms for consecutive overlapping dinucleotides plus additional terms for the ends. In SantaLucia (1998) a unified view of oligonucleotide nearest neighbor thermodynamics is presented. There, the free energy parameters of several laboratories are listed and unified parameters are derived from them, adequately describing polymer and oligomer thermodynamics.

Using the nearest neighbor model, the free energy ΔG for a oligonucleotide duplex of a sequence S_1 and its reverse compliment S_2 is given by:

$$\Delta G = \left(\sum_{i=1}^{n-1} \Delta G(s_i, s_{i+1})\right) + \Delta G(init(s_1)) + \Delta G(init(s_n)) + \Delta G(sym), \quad (2.3.12)$$

where $\Delta G(s_i, s_{i+1})$ are the standard free energy changes for the Watson-Crick nearest neighbor $5's_i\ s_{i+1}3'$. For DNA these parameters are given in table 2.1. $\Delta G(init(s_1))$ and $\Delta G(init(s_n))$ account for the start and terminal base pair. For a `G-C` pair this term is 0.98 and for a `A-T` term 1.03 $\frac{kcal}{mol}$. $\Delta G(sym)$ is an additional penalty for the maintenance of the C2 symmetry of self complementary duplexes; it equals $0.43\frac{kcal}{mol}$ if the duplex is self-complementary and zero if not.

Table 2.1: Standard free energy changes for Watson-Crick nearest neighbors in $\frac{kcal}{mol}$ taken from (SantaLucia 1998)

F	A	C	G	T
A	-1.00	-1.44	-1.28	-0.88
C	-1.45	-1.84	-2.17	-1.28
G	-1.30	-2.24	-1.84	-1.44
T	-0.58	-1.30	-1.45	-1.00

All the parameters are given at a sodium concentration of 1 M NaCl and at a fixed temperature of 37°C or 310.15K. In the original work, the components enthalpy ΔH and entropy ΔS are also given, thus ΔG can be computed for other temperatures by using the Gibbs-Helmholtz equation $\Delta G = \Delta H - T\Delta S$. Different salt concentration can be accomodated by applying a salt correction to each individual dinucleotide entropy term. Similar parameters for RNA/DNA-hybrids are given by Sugimoto et al. (1995).

Example 2.3.3 (Nearest neighbor parameters) For the duplex

$$5'\texttt{CGTTGA}3'$$
$$3'\texttt{GCAACT}5'$$

ΔG predicted by the nearest neighbor model using SantaLucia's unified parameters is

$$\begin{aligned}\Delta G &= \Delta G(init(\texttt{C})) + \Delta G(\texttt{CG}) + \Delta G(\texttt{GT}) + \Delta G(\texttt{TT}) + \Delta G(\texttt{TG}) + \Delta G(\texttt{GA}) + \Delta G(init(\texttt{A})) \\ &= 0.98 - 2.17 - 1.44 - 1.00 - 1.45 - 1.30 + 1.03 \\ &= -5.35 \text{ kcal/mol.}\end{aligned}$$

The duplex is non-self-complementary and thus $\Delta G(sym)$ is zero.

Drawbacks of the model

- The nearest neighbor model is a simplification of reality; all values are approximations and different parameters have been reported by different researchers (Sugimoto et al. 1996).
- Results for temperatures substantially different from 37°C become increasingly inaccurate, because of the different heat capacities of products and reactants.
- The parameters originate from measurements of free short oligonucleotides in solution. Hybridization in microarray experiments occurs at different conditions. The cRNA or cDNA fragments are longer and in case of the probes, attached to the chip surface.
- Non-perfect duplexes, i.e. duplexes with mismatches or loops are hard cases for the nearest neighbor model, as their pairing and stacking interactions are disturbed in complex ways. Allawi und SantaLucia derived parameters for internal mismatches, but also state that the mismatch stability is strongly context dependent (Allawi & SantaLucia 1998*b*,*c*,*a*, 1997).

Chapter 3

Nearest Neighbor Alignment

In this section we will give a brief overview of the concept of sequence alignments and their scoring functions and introduce the *Nearest Neighbor Alignment*. The scoring function used to rate a Nearest Neighbor Alignment takes energy contributions from base stacking effects into account, and can thus be used to compute a lower bound of the free energy of duplex formation of two DNA sequences.

3.1 Idea and Motivation

In general, an alignment of two sequences is a way of arranging them, one sequence on top of the other, so that the bases in one position are thought to have common functional or structural relationships, or evolutionary origin. For example, when aligning protein sequences, scoring functions that use substitution matrices like the ones of the PAM or BLOSUM family are used (for more information see Altschul 1991). The score obtained from an optimal alignment computed with an algorithm like Needleman Wunsch (Needleman & Wunsch 1970) using such a substitution matrix, indicates how likely the sequences are to be derived from a common ancestor. A more complex variant of an alignment algorithm used to fold an RNA molecule was introduced by Zuker & Stiegler (1981). The scoring function used, considers free energy associated with certain structure elements. As a result, the computed alignment and its score represent the most stable conformation and its free energy value.

In our case, we are interested in an alignment of two DNA oligonucleotides that we can interpret as a virtual secondary structure that the two molecules could possibly form. We call this structure virtual, because the two DNA molecules are not expected to form this structure in solution. Instead, the

score of the alignment obtained by the algorithm presented in this work will be used as a lower bound for the free energy associated with the DNA duplex that will form in real life. This is possible, because the algorithm uses a simplified model of the hybridization energy and takes only energetically favorable terms into account disregarding destabilizing structural elements.

As described in chapter 2.1, the nearest neighbor model can be used to approximate hybridization energy of two short DNA sequences. The key point of this model is the idea, that the energy contribution of every base pair depends on the two neighboring base pairs. Every base pair, except for the first and last pairs in the alignment, is part of two stacks and contributes twice to the overall energy. This effect is not considered by some simpler models, which rely on edit distance or other sequence composition independent measures.

3.2 Nearest Neighbor Alignment Algorithm

T-gap insertion-deletion-like metrics, introduced by D'yachkov et al. (2006), can be used for DNA hybridization thermodynamic modeling. In this section we present a slightly modified version of their algorithm and show how it is used to obtain an upper bound for the hybridization energy of two oligonucleotides. The resulting algorithm computes the score for the lowest scoring Nearest Neighbor Alignment (NNA), that can be interpreted as an upper bound for the Gibbs energy $\Delta_r G^\circ$ of DNA duplex formation.

First we the define the nearest neighbor score $NNscore$, which is a score for the thermodynamic stability of a sequence when being aligned to its reverse complement.

Definition 3.2.1 $NNscore$ of a sequence s with length l is the sum of the thermodynamic weights of all pairs, assuming a perfect matching Watson-Crick duplex.

$$NNscore(s) = \sum_{i=1}^{l-1} F(s_i, s_{i+1}) \tag{3.2.1}$$

Here, $F(s_i, s_{i+1})$ denotes the free energy associated with the naturally occuring stacked pair $5's_i, s_{i+1}3'$. For example, $F(\mathtt{G}, \mathtt{A})$ denotes the free energy associated with the stacked pair $\begin{smallmatrix} 5'\mathtt{GA}3' \\ 3'\mathtt{CT}5' \end{smallmatrix}$. Table 3.1 shows the free energy parameters in $\frac{kcal}{mol}$ taken from (SantaLucia 1998).

Scoring Scheme: The total score of the alignment is the sum of $NNscores$ of all matched stretches with a minimum length of two, i.e. it is the sum of all

Table 3.1: Thermodynamic weights of stacked pairs

F	A	C	G	T
A	-1.00	-1.44	-1.28	-0.88
C	-1.45	-1.84	-2.17	-1.28
G	-1.30	-2.24	-1.84	-1.44
T	-0.58	-1.30	-1.45	-1.00

matched dinucleotides scored by the nearest neighbor model. Mismatches and indels do not contribute to the score. They would only lead to destabilizing structures and can be omitted for the computation of an lower bound for the Gibbs energy. An Example for a Nearest Neighbor Alignment and its score can be seen in Figure 3.1.

The use of this scoring scheme is motivated by the following observations

- Indels or mismatches cannot increase the stability of the duplex.
- Edit distance does not take sequence composition into account.
- Position dependence of mismatches, as proposed by Pozhitkov & Tautz (2002), Zhang et al. (2007), is implicitly taken into account. Mismatches at the beginning or end of the sequences will disrupt only one stacked pair, whereas mismatches in the middle disrupt two stacked pairs.
- Many non-contiguous mismatches between two sequences lead to a high number of destabilizing structures. This is reflected by the Nearest Neighbor Alignment approach, as every mismatch disrupts two stacked pairs, increasing the resulting score accordingly.

```
AAGA-TGTC---CCCGAAAGGTCAGTATAC
||||  |||     |||  ||||||||||||
AAGAG-GTCTAT--CGA-AGGTCAGTATAC
```

Figure 3.1: An example of a Nearest Neighbor Alignment of two sequences of length 25. With a score of -22.3 it is the lowest-scoring alignment of the two sequences.

Given the scoring scheme, we need an algorithm which computes the lowest-scoring alignment of two sequences. The used algorithm is based on dynamic

programming and is guaranteed to find an optimal scoring alignment. Similar to other dynamic programming alignment algorithms, the NNA algorithm builds up an optimal alignment using previous solutions of optimal alignments of smaller subsequences.

Given two sequences $x = (x_1, x_2, \ldots, x_m)$ and $y = (y_1, y_2, \ldots, y_n)$, we will compute a matrix $M : \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\} \to \mathbb{R}$, where $M(i, j)$ equals the best score of the alignment of the two prefixes $(x_1, x_2, \ldots, x_i)$ and $(y_1, y_2, \ldots, y_j)$. As a sequence of length one cannot be part of an alignment with stacked pairs, M is initialized with zeros in the first row and column, $M(i, 1) = 0$ for $i = 1, \ldots, m$ and $M(1, j) = 0$ for $j = 1, \ldots, n$. $M(i, j)$ is computed recursively from the values of $M(i-1, j)$, $M(i, j-1)$, or the values on the upper left diagonal from $M(i, j)$, depending on the length of the longest common suffix of $(x_1, \ldots, x_i)$ and $(y_1, \ldots, y_j)$.

There are three different ways an existing alignment can be extended to (i, j). Table 3.2 gives an overview.

1. x_i can be aligned to a gap, the score for $M(i, j)$ is taken from $M(i-1, j)$
2. y_j can be aligned to a gap, the score for $M(i, j)$ is taken from $M(i, j-1)$
3. x_i and y_j can be matched to each other

The third case requires $(x_1, \ldots, x_i)$ and $(y_1, \ldots, y_j)$ to have a longest common suffix (lcs) of at least two, because stretches of matching bases of the length one do not contribute to the nearest neighbor score. For the third case, the number of bases r of the lcs that should be matched to maximize the score, has to be computed for $2 \leq r \leq lcs$. The score for $M(i, j)$ will be taken from $M(i-r, j-r)$. How to find r, and why r is not always the same as lcs will be shown later in this section.

Table 3.2: Three ways of extending an existing alignment to (i, j)

(1)	$(x_1,\ldots,x_{i-1})$ $(y_1,\ldots,y_j)$	x_i –	align x_i to a gap
(2)	$(x_1,\ldots,x_i)$ $(y_1,\ldots,y_{j-1})$	– y_j	align y_j to a gap
(3)	$(x_1,\ldots,x_{i-r})$ $(y_1,\ldots,y_{j-r})$	$(x_{i-r+1},\ldots,x_i)$ $(y_{j-r+1},\ldots,y_j)$	match the suffixes of length r of x and y to each other

This gives rise to the following main recursion for filling the Matrix M

$$M(i,j) = \min \begin{cases} M(i-1,j) \\ M(i,j-1) \\ D(i,j) \end{cases}, \tag{3.2.2}$$

where

$$D(i,j) = \begin{cases} 0 & \text{if } lcf < 2 \\ \min\limits_{2 \le r \le lcf} \left(NNscore(x_{[i-r+1,i]}) + M(i-r, j-r) \right) & \text{else} \end{cases} \tag{3.2.3}$$

with lcf being the length of the longest common suffix of $x_1, \ldots, x_i$ and $y_1, \ldots, y_j$. The notations $x_{[i-r+1,i]}$ and $x_{i-r+1}, \ldots, x_i$ are equivalent and may be used interchangeably throughout this work.

The full algorithm, closer to how it is actually implemented, is shown in Algorithm 1. For two sequences x and y, $nna(x,y)$ denotes the score of the optimal Nearest Neighbor Alignment of x and y.

The algorithm only matches stretches of length two or longer and adds scores as defined in the nearest neighbor model for every matched dinucleotide. Hence the name Nearest Neighbor Alignment (NNA). We are not interested in the actual alignment, but for verification and illustration purposes, we have also implemented a backtracking algorithm using a seperate traceback matrix. For every (i,j) this matrix holds the information, about which case of Equation 3.2.2 holds the minimum. In the case of $D(i,j)$, the value of its minimizer is saved.

Example 3.2.2 Let $x =$ `GAAAGG` and $y =$ `CGAAGG` be two sequences with length $m = n = 6$. After running the NNA score algorithm as described in Algorithm 1, the score matrix looks as depicted in Table 3.3. The gray fields are those of matched bases contributing to the final score. The score of the best NNA of x and y is -4.42, the value in $M(6,6)$. For descriptive purposes we also show the traceback matrix TB for this example, Table 3.4. Here the arrows indicate which value of the three choices in Equation 3.2.2 was the minimum. In the case of the third one, the number behind the $\nwarrow$ indicates wich value for r minimized Equation 3.2.3. The resulting alignment is $\begin{smallmatrix} \texttt{-GAAAGG} \\ \texttt{CGA-AGG} \end{smallmatrix}$ in single-base form, and, to better see the matched stacked pairs, in dinucleotide form $\begin{smallmatrix} \texttt{-- GA AA AA AG GG} \\ \texttt{CG GA A- -A AG GG} \end{smallmatrix}$.

Example 3.2.2 also shows why we have to take the minimum for all $2 \le r \le lcs$ in Equation 3.2.3. When reaching $(i,j) = (5,5)$ it is best to match only the last two bases, even though the lcs has a length of three. In the same way, it is better to match only the last three bases, when at $(i,j) = (6,6)$ instead of the whole lcs of 4. At that point, matching the whole lcs results in $\begin{smallmatrix} \texttt{--GAAAGG} \\ \texttt{CG--AAGG} \end{smallmatrix}$, with a

Algorithm 1: NNA score

input : two sequences x, y with lengths m, n
output: score of best Nearest Neighbor Alignment

initialize first column of M: $M(i,1) = 0$ for $i = 1, \ldots, m$
initialize first row of M: $M(1,j) = 0$ for $j = 2, \ldots, n$
for *i=2* **to** m **do**
  for *j=2* **to** n **do**
    $d = 0.0$
    $d\_tmp = 0.0$
    $score\_suf = 0.0$
    if $x_i == y_j$ **then**
      $r = 1$
      while $i \geq r$ **and** $j \geq r$ **and** $x_{i-r} == y_{j-r}$ **do**
        $++r$
        $score\_suf \mathrel{+}= F(x_{(i-r+1)}, x_{(i-r+2)})$
        if $i < r$ **or** $j < r$ **then**
          $d\_tmp = score\_suf$
        else
          $d\_tmp = score\_suf + M(i-r, j-r)$
        endif
        $d = \min(d, d\_tmp)$
      endwhile
      $M(i,j) = \min(M(i-1,j), M(i,j-1), d)$
    else
      $M(i,j) = \min(M(i-1,j), M(i,j-1))$
    endif
  endfor
endfor
return $M(m,n)$

Table 3.3: M matrix after running the NNA score algorithm with $x =$ `GAAAGG` and $y =$ `CGAAGG`. Gray fields show matched bases.

M	C	G	A	A	G	G
G	0.0	0.0	0.0	0.0	0.0	0.0
A	0.0	0.0	-1.3	-1.3	-1.3	-1.3
A	0.0	0.0	-1.3	-2.3	-2.3	-2.3
A	0.0	0.0	-1.3	-2.3	-2.3	-2.3
G	0.0	0.0	-1.3	-2.3	-2.58	-2.58
G	0.0	0.0	-1.3	-2.3	-2.58	-4.42

Table 3.4: Traceback matrix after running the NNA score algorithm with $x =$ `GAAAGG` and $y =$ `CGAAGG`. Arrows indicate where the value in the corresponding M field derived from. In the case of the $\nwarrow$, the number behind it indicates wich value for r yielded the minimum

TB	C	G	A	A	G	G
G		$\leftarrow$	$\leftarrow$	$\leftarrow$	$\leftarrow$	$\leftarrow$
A	$\uparrow$	$\uparrow$	$\nwarrow$ 2	$\leftarrow$	$\leftarrow$	$\leftarrow$
A	$\uparrow$	$\uparrow$	$\uparrow$	$\nwarrow$ 3	$\leftarrow$	$\leftarrow$
A	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$
G	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\nwarrow$ 2	$\leftarrow$
G	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\nwarrow$ 3

score of $-1.0 - 1.28 - 1.84 = -4.12$. Matching only the last three bases, allows the `A` at y_3 to match with x_2, which results in a lower score of -4.42, as there is a preceding `G` in both sequences. Figure 3.2 shows the two alignments.

```
--GAAAGG        -GAAAGG
    ||||         || |||
CG--AAGG        CGA-AGG
```

Figure 3.2: Two examples of Nearest Neighbor Alignments of $x =$ `GAAAGG` and $y =$ `CGAAGG`. *left:* Matching the whole lcf, score is -4.12 *right:* Matching only the last three bases results in the lowest scoring alignment, score is -4.42

3.2.1 Runtime

For two sequences with the same length of k, the matrix M has k^2 fields. In the worst case, the alphabet size of the concatination of both sequences is one, i.e. both sequences are stretches of the same single nucleotide. In this case, the lcs in Equation 3.2.3 will always be $\min(i, j)$ at every $D(i, j)$. The worst-case running time is thus $O(k^3)$. However, for our application to DNA sequences this is a very unlikely scenario. If we assume two sequences independently generated from the i.i.d. model, the probability of having a lcs of length 0 is $P(lcs = 0) = \frac{3}{4}$, $P(lcs = 1) = \frac{1}{4}$, $P(lcs = 2) = \frac{1}{16}$, $P(lcs = n) = \frac{1}{4^n}$. The expected length of a lcs at field $M(i, j)$ is thus

$$E(lcs) = 1\frac{1}{4} + 2\frac{1}{16} + 3\frac{1}{64} + \ldots + n\frac{1}{4^n} = \sum_{i=1}^{n} \frac{n}{4^n} \tag{3.2.4}$$

where n is the minimum of i and j. As this sum quickly converges to 0.44, it is sufficient to assume this value as $E(lcs)$ in every field of the matrix. As a result, for two independent i.i.d. model sequences the expected run time is of the order of $1.44 \cdot k^2$.

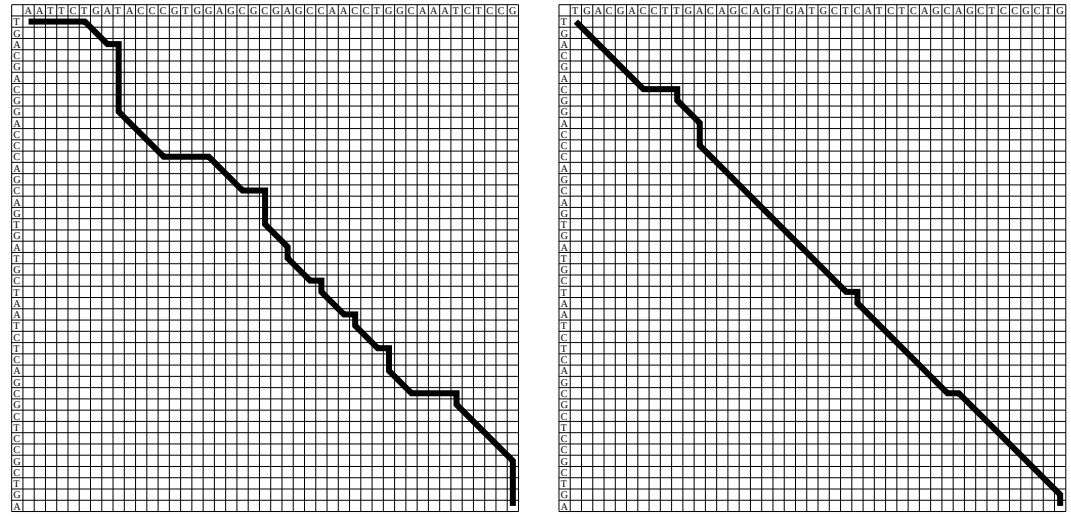

Figure 3.3: Two examples of a NNA traceback and the resulting virtual secondary structure of two sequences x and y. Both examples use sequences of length $k = 44$, the same sequence x is used in both cases. The minimum possible NNA score, $nna(x, x)$, is -65.85. *left:* Sequence y has only little similarity to x, resulting in a NNA score of -26.04 from 16 contributing base pairings. *right:* Sequence y has a higher degree of similarity, resulting in a NNA score of -51.67 from 35 contributing base pairings.

3.3 Thresholded NNA

In the previous section we described an algorithm, which will compute a lower bound on the Gibbs energy of a duplex of two oligonucleotides. When computing probe qualities, the main question is whether a probe and a given sequence exceed a certain hybridization stability. If a score stays above a threshold, it is assumed that no cross-hybridization will occur at that point. The exact score is not of interest, as long as it stays below the threshold. When looking at Figure 3.3, it can be seen that the traceback of lower scoring alignments follows fields on or close to the diagonal. Only matches contribute to the final score. Thus, fields in the matrix that are off-diagonal are less likely to be part of a traceback of a low-scoring alignment.

Other alignment algorithms based on ideas by Needleman & Wunsch (1970), Fickett (1984), Waterman (1989) exploit a similar behaviour using a banded approach, e.g. FASTA (Pearson 1990). These algorithms limit the computed fields of the matrix to the diagonal and the surrounding band of width W. The traceback is forced to stay within the given band, limiting the difference of the numbers of indels of both sequences. This technique reduces the computation time to $O(NW)$, where N is the length of the shorter of the two sequences. For sequences with sufficient similarity, given the choice of W, the same scores and alignments are returned as the rigorous Smith-Waterman calculation (Pearson 1991).

The method presented in this section uses a similar approach to reduce the number of fields of the DP matrix that have to be computed. Given a threshold score t, we do not want to fill fields of the DP matrix, that cannot be part of a traceback with a score $\leq t$. By the nature of the score matrix M, this means, that we can, depending on sequence composition and value of t, drop fields with a certain distance from the diagonal.

Given two sequences $x = (x_1, x_2, \ldots, x_m)$ and $y = (y_1, y_2, \ldots, y_n)$, let $R(i, j)$ be the value by which the score can decrease at most by aligning $x_{[i,m]}$ and $y_{[j,n]}$. Clearly, the score of an alignment of $x_{[i,m]}$ and $y_{[j,n]}$ is higher or equal to the maximum of $NNscore(x_{[i,m]})$ and $NNscore(y_{[j,n]})$, and hence

$$R(i, j) = \max(NNscore(x_{[i,m]}), NNscore(y_{[j,n]})). \tag{3.3.1}$$

In order to estimate the lowest possible score that can be reached from a field (i, j) in the DP matrix, we can add $M(i, j)$ and $R(i, j)$, but we also have to take the decrease of the score into account, that can result from traversing over all possible values of $2 \leq r \leq lcf$ in Equation 3.2.3.

To understand how this effects the score, we look closer at the algorithm. Let $M(i,j)$ be the current field in the matrix. The longest common suffix of $x_{1,i}$ and $y_{1,j}$ is lcs. Let r be the value minimizing Equation 3.2.3 and $r < lcs$. We want to estimate how much is gained by matching only the last r bases instead of lcs. When matching the last lcs bases, the score will be $M(i - lcs, j - lcs) + NNscore(x_{[i-lcs+1,i]})$. When matching the last r bases, the score will be $M(i - r, j - r) + NNscore(x_{[i-r+1,i]})$. Since picking $r < lcs$ resulted in a lower score, $M(i - r, j - r)$ must contain score contributions not found in $M(i - lcs, j - lcs) + NNscore(x_{[i-lcs+1,i-r]})$. This can be only the score of the base pair x_{i-lcs} or y_{j-lcs}. When $x_{[i-lcs,i-r]}$ can be matched to $y_{[1,j]}$, gaps are introduced after x_{i-r}. The same applies to sequence y. Ultimately, the number of stacked pairs stays the same, but one pair of the longest common suffix is exchanged for an energetically more stable pair. Figure 3.4 shows the stacked pairs that differ.

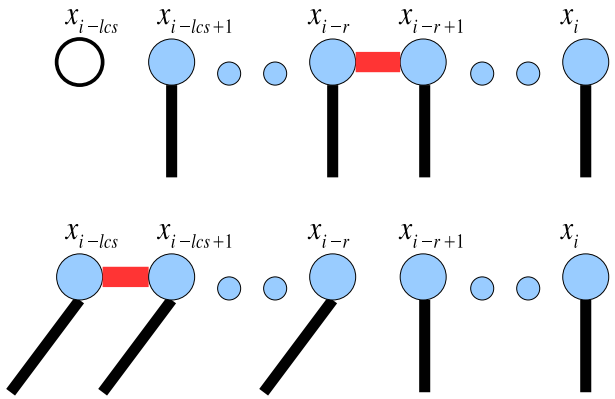

Figure 3.4: Two ways of matching the suffix of sequence x at $M(i,j)$. Sequence y is not shown. *top:* The lowest score resulted from matching the whole of the lcs of x and y. *bottom:* The lowest score resulted from matching only the last r bases of x and y, with $r < lcs$. Stacked pair contributions by which both versions differ are marked by red lines.

As a result, for the estimation of the minimal score that can be reached from a field $M(i,j)$, this means, that in case of a $lcs > 1$ for $x_{[1,i]}$ and $y_{[1,j]}$ the possibility of an additional decrease of the score by the effects described above is given. The additional decrease can be as much as 1.66, as this is the maximum difference of entries in the scoring matrix shown as Table 3.1. We define the lowest possible score that can be reached from an index i,j as $L(i,j) = M(i,j) + R(i,j) - 1.66$. By this definition, we can be sure, that for

$i \leq j$ (upper right corner of the matrix) all values of $L(k,l)$ for $k < i$ OR $l > j$ (to the right OR above) will be greater or equal to $L(i,j)$. Similarly, for $i \geq j$ all values of $L(k,l)$ for $k > i$ OR $l < j$ will be greater or equal to $L(i,j)$.

When filling out the matrix M, we can make use of these observations and ignore fields that cannot be part of the traceback of the best alignment and as such do not contribute to the minimum of Equation 3.2.2. We can also stop the computation, when we know that the lowest possible score will not fall below a given threshold.

Algorithm 2 is an extension of the NNA algorithm, such that less fields of the DP matrix have to be computed, when a score threshold is given. To save computations, we do not compute the whole R matrix containing the lowest possible score contribution from $x_{[i,k]}$ and $y_{[j,k]}$ for a given position i, j. Instead, we compute $MinScoreX$ and $MinScoreY$, which hold the minimum score a suffix can contribute to the overall score from sequence x, and y respectively. After computing the score $M(i,j)$ we can compute a lower bound for the final score an alignment passing through field i, j can have, by adding $\max(MinScoreX(i), MinScoreY(j)) - 1.66$ to it. The variable $start\_row$ holds the index of the first row of the previous column which has a lower bound for the final score below or equal to the given $limit$. This means, that in the current column rows before $start\_row$ cannot be part of the traceback, hence, the index of i starts at $start\_row$. If we have computed one or more fields of a column and reach a field with a lower bound for the final score higher than the threshold, we can ignore the rest of the column for the same reason, and start with the next one. If we do not have a single field in the column from which the threshold could be reached, we can stop the algorithm and return 0.0. Figure 3.5 shows two examples, visualizing which fields of the DP matrix have to be computed, before returning the result.

This method provides a good trade-off between bounding the DP matrix and introducing more computational effort. These simple modifications do not introduce a significant overhead to the computation, but allow to discard parts of the matrix fields. The improvement increases with higher differences between threshold and real score.

Algorithm 2: NNA score - bounded

input : two sequences x, y with length k, and threshold *limit*
output: score of best Nearest Neighbor Alignment or 0.0 if score $>$ *limit*

$MinScoreX(i) = NNscore(x_{[i,k]})$ for $i = 2, \ldots, k$
$MinScoreY(i) = NNscore(y_{[i,k]})$ for $i = 2, \ldots, k$
initialize first column of M: $M(i, 1) = 0$ for $i = 1, \ldots, k$
initialize first row of M: $M(1, j) = 0$ for $j = 2, \ldots, k$

$start\_row = 2$
for *j=2* **to** k **do**
  $start\_row\_set = False$
  for *i=start_row* **to** k **do**
    $d = 0.0$
    $d\_tmp = 0.0$
    $score\_suf = 0.0$
    if $x_i == y_j$ **then**
      $r = 1$
      while $i \geq r$ **and** $j \geq r$ **and** $x_{i-r} == y_{j-r}$ **do**
        $++r$
        $score\_suf \mathrel{+}= F(x_{(i-r+1)}, x_{(i-r+2)})$
        if $i < r$ **or** $j < r$ **then**
          $d\_tmp = score\_suf$
        else
          $d\_tmp = score\_suf + M(i - r, j - r)$
        $d = \min(d, d\_tmp)$
      $M(i, j) = \min(M(i - 1, j), M(i, j - 1), d)$
    else
      $M(i, j) = \min(M(i - 1, j), M(i, j - 1))$
    $best\_score = M(i, j) + \max(MinScoreX(i), MinScoreY(j)) - 1.66$
    if $start\_row\_set == False$ **then**
      if $best\_score <= limit$ **then**
        $start\_row = i$
        $start\_row\_set = True$
    else
      if $best\_score > limit$ **then**
        break
  if $start\_row\_set == False$ **then**
    return 0.0
return $M(k, k)$

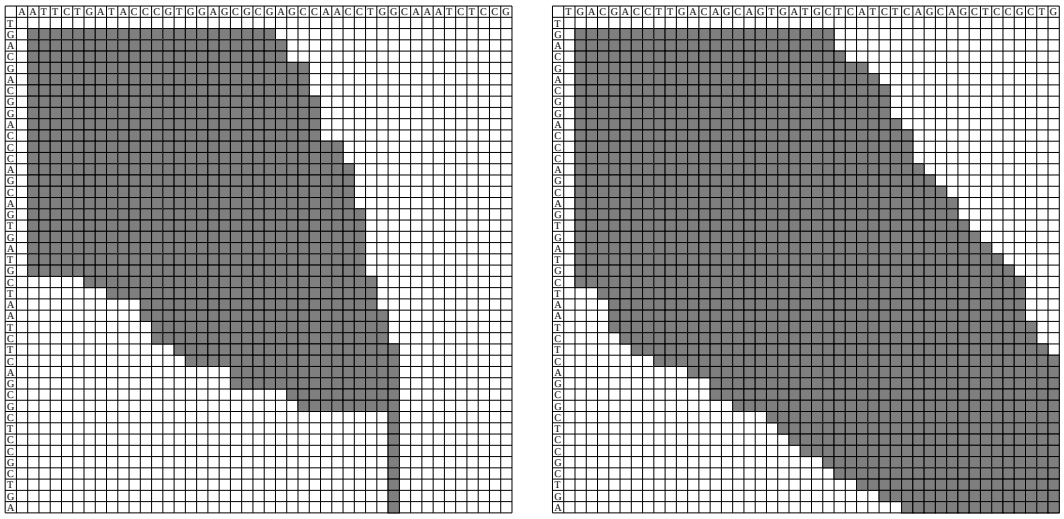

Figure 3.5: Two examples of the M matrix computed with the bounded version of the NNA algorithm. The same sequences as in Figure 3.3 are used and the score threshold is -35.0. Only the dark gray fields of the matrix have to be computed when looking for an alignment with a score less or equal to the given limit of -35.0. *left:* The two sequences cannot be aligned with an NNA score $\leq$ threshold because they are not similar enough; 0.0 is returned. *right:* The NNA score (-51.67) falls below the limit and is returned.

Chapter 4

Filtering Using the Heaviest Common Factor Approach

In the previous section, we have shown an algorithm to quickly estimate a lower bound for the Gibbs free energy of the duplex formation of two DNA sequences. This lower bound is then used when computing the specificity of a probe. The simplest but computationally most expensive way, woud be to align a probe to all non-target positions of the genome and calculate the NNA score. For large genomes and a high number of probes, this approach becomes infeasible, as the computational effort is too large. When looking at the scores, one will observe that in general the vast majority is too high to indicate cross-hybridization. In this section we introduce a filtering method to reduce the number of scores to be computed a priori, by only considering positions in the genome, where chances of obtaining a low NNA score are higher.

Our filtering method is based on the observation that low-scoring alignments have thermodynamic stable contiguous matches, which undercut a certain score threshold. Therefore, we look for stable seeds between query and database and apply the NNA algorithm to only those positions.

4.1 Similar Approaches

In most large scale database search and filtering algorithms, the edit distance is used to rate similarity of a query sequence and a given database. A lot of research has been put into developing efficient filtering algorithms for problems of this type. For example, QUASAR by Burkhardt et al. (1999) is a filtering approach that finds all local approximate matches of a query sequence in a

database. It is based on the following observation: *if two sequences have an edit distance below a certain bound, one can guarantee that they share a certain number of q-grams* (Jokinen & Ukkonen 1991, Ukkonen 1992). The QUASAR implementation uses a suffix array as index structure and shows significant improvements in CPU time, when compared to BLAST. However, it is designed to quickly detect sequences with strong similarity (>= 94% in the experiments) and it is not applicable for searching for occurences with edit distances as high as 75%. Fast programs like BLAST (Altschul et al. 1990, 1997) and FASTA (Pearson 1990) also use filtering techniques. For a given query sequence, BLAST performs a linear scan of the database searching for short substrings that also appear in the query. These hits are then extended to find longer high scoring matches. Similarly, FASTA determines all exact matches of length k between a query and a sequence, these exact matches are called *hot-spots*. *Hot-spots* in close proximity to each other can then be merged to larger alignments by applying a banded Smith-Waterman algorithm. Both BLAST and FASTA are not guaranteed to find all relevant matches of the query in the database, they are heuristics and as such are not lossless. However, sensitivity and specificity can be estimated for a given set of parameters.

4.2 Heaviest Common Factor

Existing algorithms like FASTA and BLAST use a *seed and extent* approach and, in the case of DNA sequences, define a seed to be a substring of query and database with a length exceeding a given threshold. Rahmann (2003) goes as far as introducing the *longest common factor* as a measure of specificity of a probe, but also allowing gaps. Chen et al. (2006) take the same approach. In this section we introduce the *heaviest common factor* (HCF), a weighted substring of two sequences, which we will use in our own *seed and extent* approach.

Our filtering method exploits the correlation between the NNA score $nna(p, t)$ of a probe p and a target t and the weight of the *heaviest common factor* of p and t.

Notation: We write $s \lhd t$ if s is a factor of t; the cases that s is empty or that $s = t$ are allowed.

Definition 4.2.1 (Heaviest common factor) A *common factor* of two strings p and t is a string s with both $s \lhd p$ and $s \lhd t$. A common factor is a *heaviest common factor* if no energetically more stable factor exists. We write

$$hcf(p, t) := \min\{NNscore(s) : s \lhd p \text{ and } s \lhd t\} \tag{4.2.1}$$

for the weight of the heaviest common factor.

Note the minimum in the definition; the weight is the sum of free energy contributions from stacked pairs (which are all negative), and the factor which can contribute most to the overall energy associated with the duplex formation of p and t is called the *heaviest common factor*. The heavier a common factor, the lower its score.

Using the *heaviest common factor* as an indicator for cross-hybridization is motivated by the following observations:

- duplex formation needs a sufficiently stable core to initiate binding.
- low scoring Nearest Neighbor Alignments usually have relatively heavy common factors, and
- depending on sequence composition, the heaviest common factor need not be the longest.

```
A---CGCGTAG-TC-
    |||   ||
-ACTCGC--AGG--C
```

(a) $hcf = -4.41$, factor is `CGC`

```
AC-GCGT-AGTC
||   |||   |||
ACA-CGTC-GTC
```

(b) $hcf = -3.61$, factor is `CGT`

```
A--CGCGTAG--TC
     ||||||
-TG-GCGTAGAG--
```

(c) $hcf = -7.71$, factor is `GCGTAG`

Figure 4.1: Examples of sequence pairs, their heaviest common factor hcf and the resulting alignment.

4.3 Algorithm Outline

Here we want to present the basic outline of our filtering mechanism. We are given a set of genomic sequences D. For a given probe p, with a probe - target NNA score of t, find all heaviest common factors of p and d, with $d \in D$ and a weight below a given threshold t. A heaviest common factor conforming these requirements is called a *seed*. As we are given a large set of probes P with corresponding thresholds T to check against D, we will use q-gram indices for P and D and process all probes simultaneously. The outline is:

1. Generate q-gram index for all probes and for all sequences
2. Generate list of possible seeds

3. Iterate over seeds and use q-gram indices to find occurences in probes and sequences

The maximum length of seeds is given by a variable $qmax$. The minimum length depends on the threshold T_i for a given probe P_i. For example, if T_i is -4.5, there will not be a factor ≤ -4.5 with a length ≤ 4. The lowest scoring 3-grams (`GCG` and `CGC`) have a score of -4.41.

4.4 Q-gram Indices for Probes & Sequences

For a given seed s, which is simply a short string of length l, we need to find a subset P^* of our given probe set P with s being a factor of p for all $p \in P^*$. We also need to know the exact position in p where s starts. In the same way, we want to find all occurences of s in the given sequences D. To accomplish this, we build seperate q-gram indices over all probes in P and all sequences in D. The length l of a seed is not the same for all seeds, it depends on sequence composition. The seeds can vary in length, $qmin \leq l \leq qmax$. To account for that, we could build an index for every possible value of l, but this would require large amounts of memory. For example, for the simple case of D containing only one sequence d with length $|d|$ and $qmin = 2$, this would require

$$\begin{aligned}
\text{space} &= 4^2 + (|d| - 1) + 4^3 + (|d| - 2) + \ldots + 4^{qmax} + (|d| - qmax + 1) \\
&= (qmax - 1)|d| + (4^{qmax}) \sum_{i=0}^{qmax-2} \frac{1}{4^i} \\
&\leq (qmax - 1)|d| + 1.33 \cdot 4^{qmax}.
\end{aligned}$$

This can be interpreted as the number of pointers and positions we need to retain in memory.

To save memory, we can take advantage of the fact that every q-gram of length q will also be indexed in a q-gram index for $q + 1$. For all q-grams of length q but the last one in a sequence, there is a q-gram of length $q + 1$ containing it as a prefix. This observation allows us to only build a q-gram index for $qmax$ and still be able to find shorter q-grams by looking for longer q-grams starting with them. Special care has to be taken to deal with shorter q-grams at the sequence ends. These have to be indexed seperately, because they will not be indexed as a prefix of a longer q-gram. For the above example,

the required space will be reduced to

$$\begin{aligned}\text{space} &= 4^2 + 1 + 4^3 + 1 + \ldots + 4^{qmax} + (|d| - qmax + 1) \\ &= (|d| - qmax + 1) + (qmax - 2) + (4^{qmax}) \sum_{i=0}^{qmax-2} \frac{1}{4^i} \\ &\leq |d| - 1 + 1.33 \cdot 4^{qmax}.\end{aligned}$$

It is important to note that there is no factor in front of the $|d|$, which results in significant space savings for large sequences.

Q-gram index notation: From now on qgi_q^P will denote a q-gram index over all sequences in the sequence set P. Indexed q-grams have the length q. If $q < qmax$, the indexed q-grams will only be the ones at the last q positions of the sequences in P. When the sequence set is not relevant or clear from the context it is ommitted. When no subscript is given, qgi^P indicates the set of all qgram-indices qgi_i^P for $i = qmin, \ldots, qmax$.

When querying the q-gram index for a sequence s with length $|s| < qmax$, we have to consider all indexed q-grams with prefix s. For example, when $|s| = qmax - 1$, all occurrences of s are $qgi_{qmax}(s\mathtt{A}) \cup qgi_{qmax}(s\mathtt{C}) \cup qgi_{qmax}(s\mathtt{G}) \cup qgi_{qmax}(s\mathtt{T}) \cup qgi_{qmax-1}(s)$. Here, $s\mathtt{A}$ denotes a string resulting from appending `A` to s. Recursively, this scheme can be applied to shorter sequences to find their occurrences. In the indices, all q-grams are sorted lexicographically. This provides an effortless access to all q-grams sharing the same prefix. The first q-gram with length q and s as a prefix is expanded by `A`s until it has length $qmax$. Starting with this q-gram, the next $4^{qmax-|s|}$ q-grams all share the same prefix s. All occurrences of string s in a set of sequences P, are given by

$$hit(s) = \begin{cases} qgi_{qmax}(s) & \text{if } |s| = qmax \\ hit(s\mathtt{A}) \cup hit(s\mathtt{C}) \cup hit(s\mathtt{G}) \cup hit(s\mathtt{T}) \cup qgi_{|s|}(s) & \text{if } |s| < qmax \end{cases}$$

Here the superscript P is omitted, as it is clear that the indices are build over the sequence set P.

Table 4.1: Example for q-gram indices for the two sequences $P = \{\texttt{ACGCTCGT}, \texttt{GGTCGCTC}\}$. Hits are tuples of the sequence number and position at which the q-gram occurred, both starting at 0. Here only non-empty rows, i.e. q-grams that actually occur in the sequences, of the indices are shown. In the implementation, all possible q-grams are indexed. The left part of the table shows three seperate q-gram indices. In the right half, only for $q = qmax$ all q-grams are indexed, and for shorter q-grams only those occurrences which are not prefixes of already indexed longer q-grams, are indexed.

full indices for $2 \leq q \leq 4$			full index for $q = 4$ and indices for sequence ends		
index	qgram	hits	index	qgram	hits
qgi_4	ACGC	(0, 0)	qgi_4	ACGC	(0, 0)
	CGCT	(0, 1), (1, 3)		CGCT	(0, 1), (1, 3)
	CTCG	(0, 3)		CTCG	(0, 3)
	GCTC	(0, 2), (1, 4)		GCTC	(0, 2), (1, 4)
	GGTC	(1, 0)		GGTC	(1, 0)
	GTCG	(1, 1)		GTCG	(1, 1)
	TCGC	(1, 2)		TCGC	(1, 2)
	TCGT	(0, 4)		TCGT	(0, 4)
qgi_3	ACG	(0, 0)	qgi_3	CGT	(0, 5)
	CGC	(0, 1), (1, 3)		CTC	(1, 5)
	CGT	(0, 5)	qgi_2	GT	(0, 6)
	CTC	(0, 3), (1, 5)		TC	(1, 6)
	GCT	(0, 2), (1, 4)			
	GGT	(1, 0)			
	GTC	(1, 1)			
	TCG	(0, 4), (1, 2)			
qgi_2	AC	(0, 0)			
	CG	(0, 1), (0, 5), (1, 3)			
	CT	(0, 3), (1, 5)			
	GC	(0, 2), (1, 4)			
	GG	(1, 0)			
	GT	(0, 6), (1, 1)			
	TC	(0, 4), (1, 2), (1, 6)			

4.5 Iterating Over Possible Seeds

In the previous section, we introduced q-gram indices and how they can be used to find all occurrences of a short string in an indexed set of sequences. We will make use of these indices, when looking for common factors between the set of probes and the set of genomic sequences.

Our filtering method follows a *seed and extend* approach, where seeds are substrings of probe and target which can contribute to the NNA score more than a certain threshold. Every seed is given a weight, which corresponds to its free energy contribution. We generate a list of all possible q-grams for $qmin \leq q \leq qmax$ and sort them ascending by its weight, i.e. the q-gram with the highest contribution to duplex stability comes first. Table 4.2 shows a list of weighted q-grams for $2 \leq q \leq 11$.

For every probe P_i we define a weight threshold T_i, which determines the maximum score a seed can have when looking for common seeds of P_i and the database D. This T_i is defined as a fraction of the NNA score of P_i and its intended target $nna(P_i, P_i)$. For example, if a probe and its intended target have a NNA score of -59.20, then the maximum score for a seed will be expressed as $w \cdot (-59.0)$. Here w can be in the range of $0 \leq w \leq 1$. Small w will result in a greater number of seeds that will be considered which in turn leads to more positions in the database that will have to be verified. In the above example, a w of 0.1 will lead to a threshold of -5.92, the last possible seed to be considered is then `GCCC` with a score of -5.92.

The whole filtering process can be summarized by the following steps:

- Read database D and build q-gram index qgi^D
- Read probes P, build q-gram index qgi^P and calculate seed thresholds T
- Generate list of possible seeds and sort by weight
- For every $p \in P$
    - For every occurrence of a factor of p and D with length $> qmax$, compute NNA score of probe p and subsequence of D at that position
- For every seed s
    - For every $p \in P$ containing s and seed threshold $\geq$ seed weight
        - For every occurrence of s in D, compute NNA score of probe p and subsequence of D at that position

Table 4.2: Some of 5,592,400 q-grams with $2 \leq q \leq 11$ ordered descending by score contribution.

number	q-gram index position	sequence	length	weight
1	1677721	CGCGCGCGCGC	11	-22.05
2	2516582	GCGCGCGCGCG	11	-22.05
3	2464153	GCCGCGCGCGC	11	-21.72
4	2513305	GCGCCGCGCGC	11	-21.72
⋮	⋮	⋮	⋮	⋮
2091291	4115044	TTGTAGGCGC	10	-14.24
2091292	55513	AAATCGATCGC	11	-14.24
⋮	⋮	⋮	⋮	⋮
5425215	2095664	CTTTTGGAT	9	-9.75
5425216	1050560	CAAAACTTAA	10	-9.75
⋮	⋮	⋮	⋮	⋮
5451275	2071296	CTTGCGT	7	-9.58
5451276	565760	AGAGGAGA	8	-9.58
⋮	⋮	⋮	⋮	⋮
5589949	4115456	TTGTATA	7	-5.93
5589950	2441216	GCCC	4	-5.92
⋮	⋮	⋮	⋮	⋮
5592399	786432	AT	2	-0.88
5592400	3145728	TA	2	-0.58

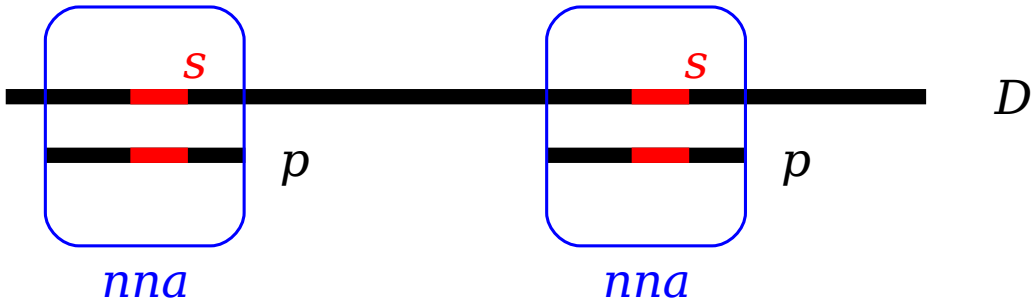

Figure 4.2: One step during filtering. Probe p contains seed s, and NNA scores at positions of s in the sequence set D are computed. Nearest Neighbor Alignments are only computed between regions in the boxes.

4.6 Avoiding Redundant Computations

When iterating over the list of sorted seeds, we will reach seeds that are factors of seeds that had been considered in an earlier iteration. In practice this means, that for a seed s redundant computations will be made if we simply compute NNA scores of all $hit^P(s)$ with $hit^D(s)$ for a probe set P and a target sequence set D. Figure 4.3 illustrates an instance of this problem.

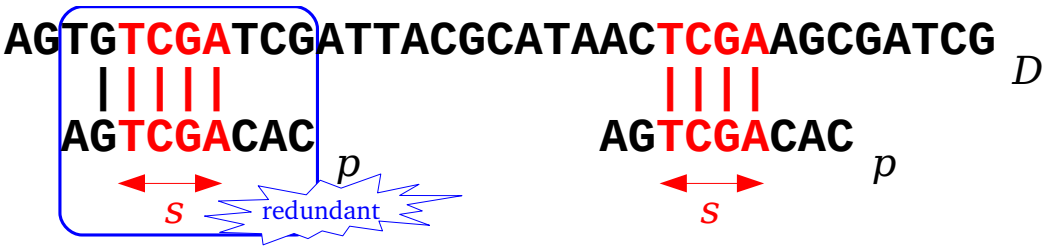

Figure 4.3: Example for redundant computation of NNA score. Current seed $s =$ `TCGA`. Probe p contains s and NNA scores of p and sequence D are computed for those positions where s in p matches s in D. The seed `GTCGA` will have been processed earlier in the course of the algorithm and the NNA score of p and D at the first occurrence of s will have already been computed.

Whenever probe and sequence have the same base before or after the seed, the NNA score will have already been computed. For random sequences from the i.i.d. model we can quantify how often this will happen for a given seed s. The probability that the base coming directly before the seed is different in the probe and the sequence is 3/4, it is also 3/4 for the base just after the seed. As the two positions are independent, the joint probability is 9/16. This means that the probability of having the same base before or after the seed is 7/16. Almost half of the NNA scores need not to be computed again.

In order to avoid these redundant computations, we simply check the positions before and after the seed in the current probe and at the current hit in the sequence set for identity. Is the same base before or after the seed in probe and sequence present, we can skip the NNA computation and proceed with the next hit.

Chapter 5

Computation of Probe Qualities

In the previous two sections we introduced the NNA score, an upper bound for the free energy of probe - target hybridization, and a seed and extend filtering approach to speed up the computation of NNA scores for a large set of probes versus a database. In this section we motivate the introduction of a *cross-hybridization potential*, which we interpret as a specificity measure, and use it to rank the given probes by their quality.

During a microarray experiment, the target DNA sequences are placed on the chip. We assume a high concentration of all targets and an equal distribution over all probes. We also assume, that when a probe - target pair has a NNA score greater than a certain threshold, hybridization can occur. This threshold is a variable and depends on the sequence composition of the probe and a free energy value ΔE given by the user. This ΔE is the minimum difference of NNA scores between probe - intended-target and probe - unintended-target that eliminates the chance of cross-hybridization. The intended target and unintended targets compete for the probe on the chip, with the duplex of probe and intended target having the greatest stability. Thus, it is reasonable to define the threshold as $nna(\text{probe}, \text{probe}) + \Delta E$, i.e. the free energy of probe - intended-target duplex raised by ΔE. For example, if a probe and its intended target have a NNA score of -65.30, and ΔE is 30.0, then all NNA scores of probes and unintended targets smaller than -35.30 are considered to lead to cross-hybridization.

5.1 Cross-Hybridization Potential

For some applications, such as designing tiling arrays, it can be desirable to also use non-perfect probes, i.e. probes that, given certain criteria, might cross-

hybridize. Now, we do not only want to distinguish between good (unique, not cross-hybridizing) and bad (non-unique, cross-hybridizing) probes, but want a measure of badness for probes that might cross-hybridize. In this section we introduce such a measure, the *cross-hybridization potential*, and show how to compute it.

Definition 5.1.1 (Cross-hybridization potential) Given a probe p and its NNA scores $T = T_1, \ldots, T_n$ with the database, and a cross-hybridization threshold $cht = nna(p, p) + \Delta E$, the cross-hybridization potential chp of p is defined as

$$chp(p) = \sum_{i=1}^{n} \begin{cases} -T_i + cht & \text{if } T_i < cht \\ 0 & \text{else.} \end{cases} \tag{5.1.1}$$

Whenever a NNA score indicates cross-hybridization, the amount it falls below the threshold becomes part of the sum.

We make a few assumptions to justify this measure. Sequences with a NNA score $\leq cht$ do not hybridize. The more cht is surpassed, the stronger the affinity of probe and sequence becomes. All sequences, that are given onto the microarray are present in a high number of copies which are distributed evenly over the surface of the chip. We assume the hybridization process to be stochastic. The probability of hybridization increases linearly with the amount tne NNA score surpasses the cht, and the number of sequences present.

For example, a probe p with $nna(p, p) = -70$ and a choice of $\Delta E = 35$ has the chance of cross-hybridizing at seven position in the genome. At every of these seven positions, the NNA score is around -40, surpassing the cross-hybridization threshold $cht = -70 + 35 = -35$ by 5. This leads to a cross-hybridization potential of $chp(p) = 35$. For another probe t, with $nna(t, t) = -70$ and the same choice of $\Delta E = 35$, there exists one perfect match at a not intended position with a NNA score of -70 and no other cross-hybridization is observed. The cross-hybridization potential of t is then $chp(t) = 35$.

The chps of different probes can only be compared if they were computed, using the same target sequence set. The scores itself can then be used to rank the probes by their probability of cross-hybridizing. For the above example, this means, that p and t have the same probability of cross-hybridizing, whereas a probe with a lower chp value is less likely to cross-hybridize.

Chapter 6

Experiments

6.1 Data Used

Generated Data: Artificial DNA sequences generated from the i.i.d. model, if not stated otherwise. For sequences with a desired GC content every base was drawn independently from its neighbors from the distribution $\mathtt{A} : 0.5(1-gc), \mathtt{C} : 0.5gc, \mathtt{G} : 0.5gc, \mathtt{T} : 0.5(1-gc)$, where gc specifies the desired GC content as fraction of $\mathtt{G}+\mathtt{C}$ of all bases of the sequence.

Mycoplasma genitalium: sequence length 580,076 nt; GC content 0.31 (NCBI 2001*b*)

Human chromosome 21: sequence length 35,449,598; GC content 0.41 (NCBI 2008)

Escherichia coli K12: sequence length 4,639,675 nt; GC content 0.50 (NCBI 2001*a*)

Mycobacterium bovis BCG Pasteur 1173P2: sequence length 4,374,522 nt; GC content 0.65
(NCBI 2007)

6.2 Influence of GC Content on Scores

Unlike the edit distance, NNA scores are affected by sequence composition, not only the number of matches or mismatches. In order to assess the influence different GC contents have on the NNA scores, we ran the algorithm on generated and real data. For every GC content level from 0.25 to 0.75 in steps of size 0.01,

and every genomic sequence listed in Section 6.1. we performed the following procedure:

- Pick a random *probe* of length 50 from the sequence
- Do 100,000 times :
  - Compute NNA score for *probe* and its perfect Watson-Crick complement (*target*)
  - Pick a random *non-target* of length 50 from the sequence
  - Compute NNA score for *probe* and *non-target*
  - *probe*:=*non-target*

For the synthetic data, instead of picking a random 50mer from a sequence, a random sequence of length 50 was generated according to the parameters given by the GC content.

The results are shown in Figure 6.1. Here, the mean and standard deviation of the results is plotted versus the GC content. There is a linear increase in the probe - target scores, which is due to the higher chance of energetically stable pairs, like those containing a `G` or `C`. For the probe - non-target scores the slope of the curve increases with higher GC content, also the standard deviation increases. The results for the random selected 50mers from the real data sets, comply with these findings. The higher standard deviation of both scores in *Human chromosome 21* are an artefact of the uneven distribution of `G`s and `C`s in the sequence. They accumulate in CpG islands, and are underrepresented in the rest of the genome (Takai & Jones 2002, Dunham 2005).

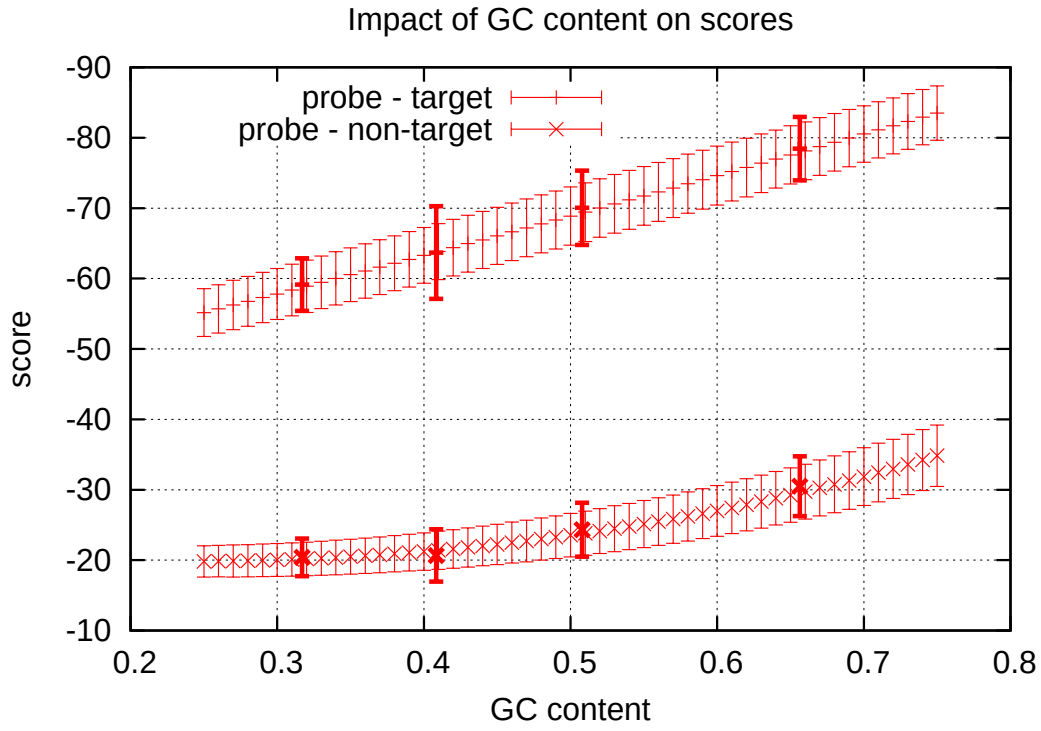

Figure 6.1: Influence of GC content on average scores. Probe-target and probe-non-target NNA scores for random 50mers with different GC content levels. Also shown with thicker lines, the results for *Mycoplasma genitalium* (GC content of 0.32), *Human chromosome 21* (0.41), *Escherichia coli K12* (0.51) and *Mycobacterium bovis BCG Pasteur 1173P2* (0.66). For every GC level, the scores of 100,000 samples are averaged.

6.3 NNA Score Compared to Edit Distance

We introduced the NNA score as an upper bound for the free energy associated with the duplex formation of two DNA strands. This measure can then be used to predict the likelihood of duplex formation between a probe and a target during a microarray experiment. Currently, the edit distance is an often used predictor for hybridization. The experiments in this section show how the edit distance performs versus the NNA score, when compared to a computationally expensive computed free energy value.

To obtain accurate free energy values, we use the program **hybridize**, which is part of **unafold**, which in turn is part of **DINAMelt** by Markham & Zuker (2005). DINAMelt is a software package available on a webserver, which simulates the melting of one or two single-stranded nucleic acids in solution. It predicts the melting temperature for a hybridized pair of nucleic acids and entire equilibrium melting profiles as a function of temperature. For the computation of free energy of a duplex, stacked pairs, interior loops, bulges and dangling bases at the ends are taken into account, and all possible conformational states are recursively tested (Dimitrov & Zuker 2004). We modified the program **hybridize**, so that it would take the input sequences as command line arguments, rather than reading them from files. All free energy values are those of the duplex at a temperature of 37°C and a Na^+ concentration of 1 M and Mg^{++} concentration of 0 M.

We picked a random probe of length 10 and computed the edit distance, NNA score and the free energy to all $4^{10} = 1,048,576$ possible sequences of length 10. Figure 6.2 shows the results. In $1,732$ cases **hybridize** did not return valid free energy values, for the other cases we plotted edit distance versus free energy and NNA score versus free energy. It can be seen, that there is a much more pronounced correlation between NNA score and free energy, than between edit distance and free energy score. Free energy scores for a certain edit distance are spread over a large range, and a low edit distance does not necessarily mean a low free energy value. Similarly, low free energy values appear at high edit distances. The point cloud for the NNA score is more compact and shows a linear behaviour.

To assess the performance of the edit distance and the NNA score as a classifier for probe-target hybridization in microarray experiments, we generate two more datasets. We now use a longer probe sequence of length 50 and therefore have to generate the datasets carefully as to choose sequences with certain properties. For both datasets, we use the same randomly picked sequence of length 50 as probe p, with $nna(p, p) = -76.63$. The two datasets are:

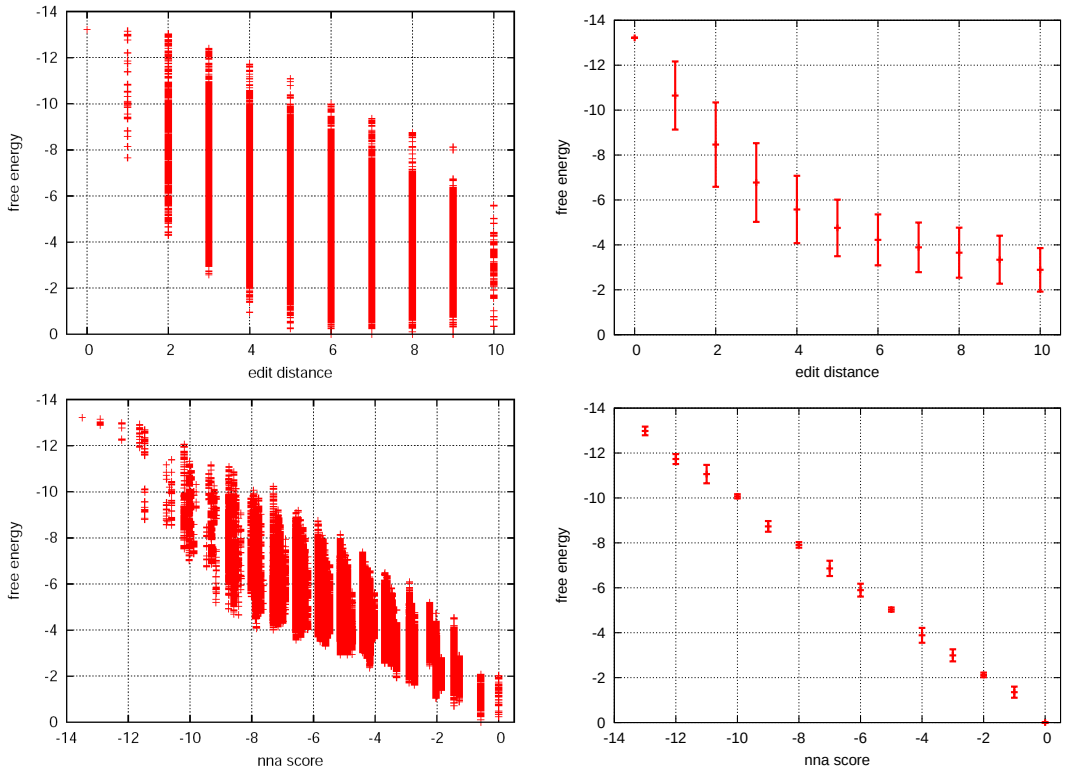

Figure 6.2: Edit distance and NNA scores versus free energy. Scores were computed between 10mer `TACACGTGCT` and all possible 10mers. The top row shows the edit distance vesrus free energy, bottom row nna score versus free energy. The left side shows all points, the right side only the mean and standard deviation.

- **random**: Pick random positions of the probe and substitute with a different random base. This dataset has 526,190 unique sequences with edit distances to the probe between 0 and 15.
- **placed**: Create random mismatches at either the begining, end or middle of the probe sequence, or uniformly distribute the mismatches. This creates sequences that lead to duplexes with dangling ends or matching blocks. This dataset has 184,874 unique sequences with edit distances to the probe between 0 and 15.

For every sequence in both datasets we computed the free energy value, edit distance and NNA score with the probe. The distribution of edit distance and NNA score versus free energy of both datasets is plotted in Figure 6.3 and Figure 6.4. Plots showing the point clouds are given in Figure A.1 in the Appendix. There is a strong linear correlation for both measures for the **random** dataset. For the **placed** dataset, the edit distance shows less correlation and similar free energy values are spread over multiple edit distances. In contrast, NNA score shows the same linear correlation for this dataset as for the **random** dataset.

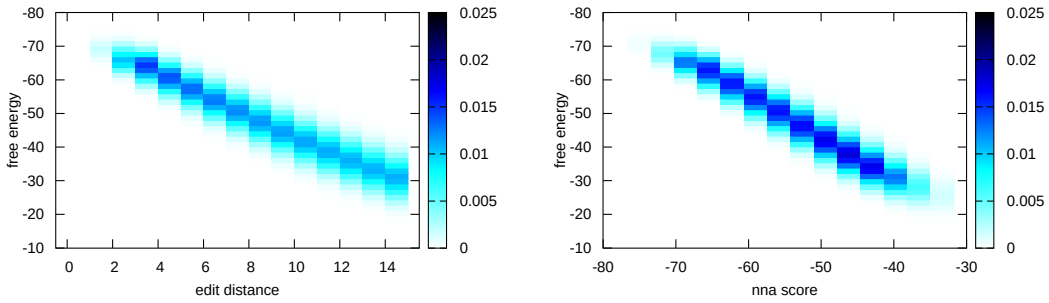

Figure 6.3: Heat map showing the distribution of free energy values for the probe versus dataset **random**. *left*: edit distance versus free energy *right*: NNA score versus free energy.

To evaluate sensitivity and specificity of NNA score and edit distance as a classifier for cross-hybridization, we select a free energy value of -46.6 as the true cross-hybridization threshold, this corresponds to a ΔE of 30. For varying edit distances from 0 to 15 in steps of 1 and nna scores from 0 to -76.63 in 100 steps, we computed the sensitivity, specificity and false positive rate. The resulting ROC plots are shown in Figure 6.5. The corresponding specificity and sensitivity plots are given in the Appendix in Figure A.2. For the **random** dataset, both measures show an equal performance, but for the **placed** dataset,

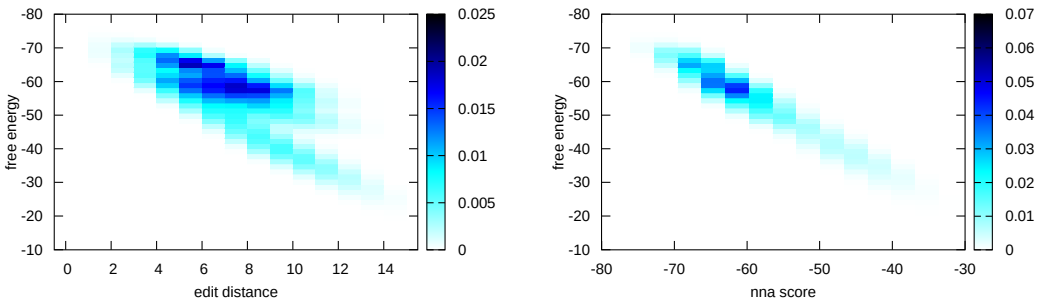

Figure 6.4: Heat map showing the distribution of free energy values for probe versus dataset **placed**. *left*: edit distance versus free energy *right*: NNA score versus free energy.

the performance of the NNA score increases, while the performance of the edit distance decreases. This is a direct result of the lower correlation of edit distance and free energy for this dataset (see Figure 6.4).

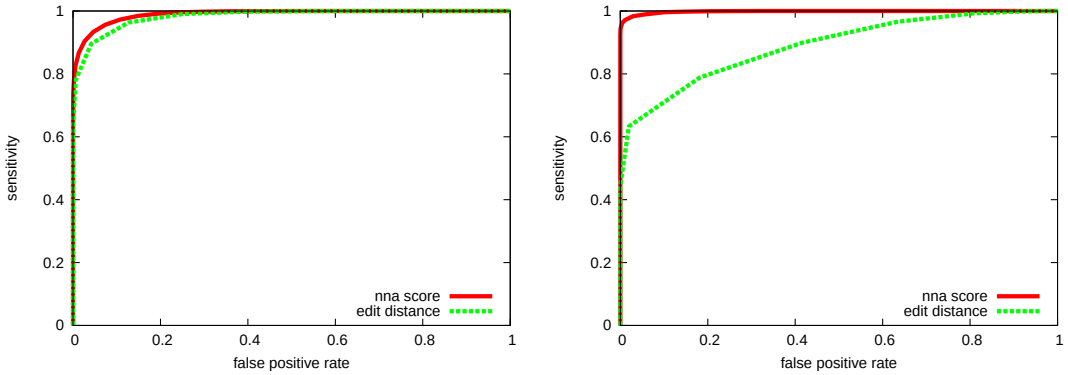

Figure 6.5: ROC curves for edit distance and nna score as classifier for a free energy value below -46.6. *left*: probe versus **random** dataset *right*: probe versus **placed** dataset

6.4 NNA Score Versus Kane's Criteria

The experiments of the previous section have shown that the NNA score is highly correlated with free energy and that the edit distance is only of limited use, when predicting free energy. Thus, Kane et al. (2000) use an additional

criteria to define cross-hybridizing probes. For 50mers they define the following rules: targets with > 75–80% sequence similarity or contiguous stretches of > 15 identical bases will lead to cross-hybridization. With the following experiments we want to evaluate the performance of Kane's criteria and compare it with the NNA score.

Dataset: We combine the two datasets **random** and **placed** of section 6.3 to one dataset **random+placed**.

For every probe-target pair, the free energy, edit distance, length of the longest common substring and NNA score is computed. Free energy values are obtained by the same means as in section 6.3. We want to evaluate the performance of Kane's criteria and the NNA score with various ΔE values as a classifier for cross-hybridization. For varying true free energy cross-hybridization thresholds, we compute specificity, sensitivity and false positive rate of detecting cross-hybridization for both methods. Free energy cross-hybridization thresholds are picked in the range from 0 to -76.36 in 100 steps. Three different NNA score cross-hybridizatiom thresholds are used: $-76.33 + \Delta E$ for $\Delta E \in \{25, 30, 35\}$. Kane's criteria resulted in the following thresholds: an edit distance ≤ 12 (similarity $\geq 76\%$) or a length of the longest common substring ≥ 16 leads to cross-hybridization.

The results are shown as a ROC curve in Figure 6.6. Figure A.3 in the Appendix shows the ROC curves for the seperate datasets. For all ΔE values, the NNA score shows a better performance than Kane's criteria, coming closer to the upper left corner of the plot. For high sensitivity the NNA score yields a smaller false positive rate than Kane's criteria. In practice, a relatively small difference in the false positive rate can lead to more probes being considered as not cross-hybridizing probes. The false positive rate is the fraction of false positives of all real negatives, so wrongly detecting cross-hybridization increases this rate. When testing a probe for cross-hybridization with all targets, the number of real negatives will be much larger than the number of real positives, as cross-hybridization does not take place often. The results of this experiment show, that the NNA score will be able to label more probes as good probes (not cross-hybridizing) and still retain a high sensitivity.

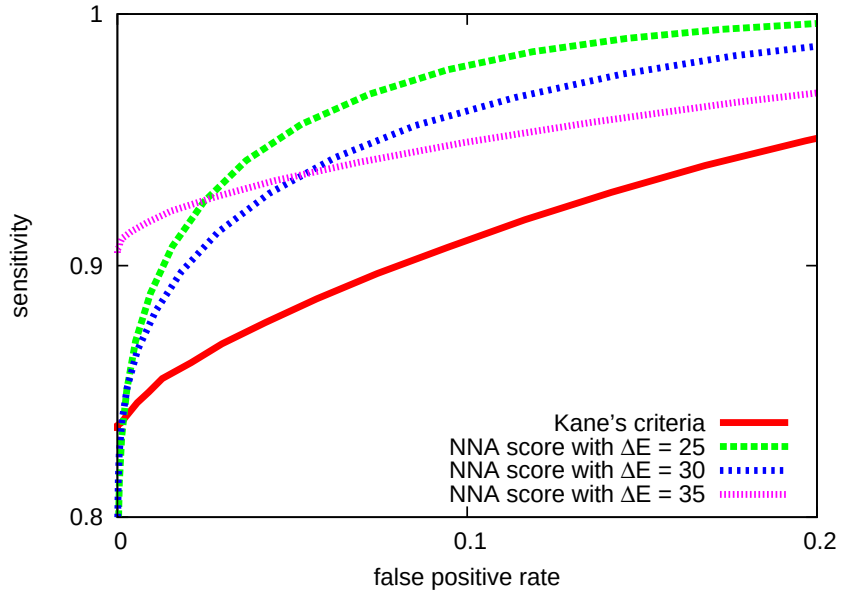

Figure 6.6: ROC curve for predicting cross-hybridization for varying free energy cross-hybridization thresholds. The NNA score for the probe and the inteded target is -76.63 resulting in NNA thresholds of $-51.63, -46.63 and -41.63$ for $\Delta E = 25, 30$ and 35. Note the ranges of the axes, this is only the upper left corner of the ROC plot.

6.5 Filtration Performance

In chapter 4 we introduced a filtering method we use to reduce the amount of sequence we have to scan for cross-hybridization for each probe. With the following experiments we want to evaluate the performance of this filter.

6.5.1 Filtration Ratio

For a filter the filtration ratio is defined as

$$\text{filtration ratio} = \frac{\#\text{ of particles downstream}}{\#\text{ of particles upstream}}.$$

The smaller this number, the less particles passed the filter. In our case, every $k$mer of the sequence, which does not overlap with the probe is an upstream particle, a candidate that has to be checked for cross-hybridization with the given probe using its NNA score. When applying out filtering technique, only a portion of all these candidates is considered. This portion corresponds to the downstream particles.

In this experiment, the following sequences were used: *Mycoplasma genitalium*, *Escherichia coli*, *Mycobacterium bovis* and a random sequence of length 1 Mbp. For every sequence the following procedure is carried out:

- Do 1000 times:
  - Pick a random 50mer as probe
  - For every 50mer in the sequence not overlapping with the probe
    - Compute weight of heaviest matched factor with probe

The heaviest matched factor is similar to the heaviest common factor but with the constraint that the factor starts at the same position in both sequences. For our example this means that if the weight of the heaviest matched factor is below the seed threshold T (compare to section 4.5), this 50mer of the sequence will never be considered, and the NNA score will not be computed. Thus, this 50mer will not appear downstream.

For every probe p the seed threshold T is given as $T = w\ nna(p, p)$. We vary w from 0 to 0.3 in 100 steps and compute the filtration ratios and call this the seed threshold. The mean values for the 1000 probes of all sequences are shown in Figure 6.7. More detailed plots showing the standard deviation of the runs can be found in the Appendix in Figure A.4.

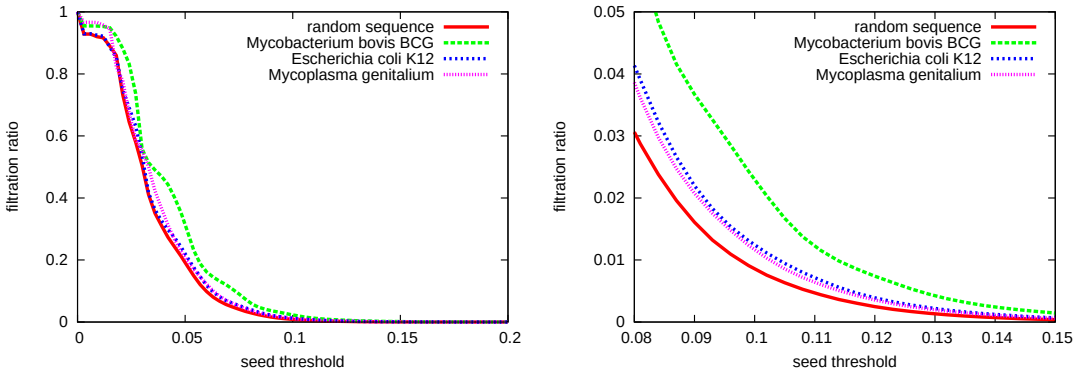

Figure 6.7: Filtration ratios for different seed thresholds and different sequences. The x-axis show the seed threshold as the value of w. The right plot shows only a section of the left one.

For increasing seed thresholds, the filtration ratio decreases exponentially, quickly reaching values below 10%. For example, for *Escherichia coli* the mean filtration ratio when picking $w = 0.1$ for the seed threshold is 1.3%. As expected, filtration ratios for the random sequence are the lowest, as this sequence shows the highest entropy. Biological sequences show reoccurring motifs and repeated regions, but filtration ratios are on the same order. For example, for $w = 0.11$ the filtration ratio for the random sequence is 0.5%, for *Mycobacterium bovis* it is 1.2%.

6.5.2 Filtration Quality

In the previous section we examined the filtration ratio of our filter for different seed thresholds and sequences. This gives us information about the amount of sequence that we have to inspect for cross-hybridization. With the experiment in this section we want to analyze the quality of the filter. The goal of this filter is to let positions of a target sequence set pass, which will lead to cross-hybridization, whereas the other position, which will lead to NNA scores greater than the cross-hybridization threshold are blocked (i.e. not considered). Similar to the above experiment, we take the same four sequences and the same 1000 random probes for each. We then proceed:

- For every 50mer in the sequence not overlapping with the probe
  - Compute weight of heaviest common factor hcf with probe
  - Compute NNA score with probe

- If $hcf >=$ seed threshold:
  - If NNA score > cross-hybridization threshold: count as true positive
  - Else: count as false positive
- Else:
  - If NNA score > cross-hybridization threshold: count as false negative
  - Else: count as true negative

The cross-hybridization threshold is given by ΔE, as described in chapter 5. As in the previous section, the seed threshold is given by the value of w, the fraction of the NNA score of probe and intended target. The experiment is run for $\Delta E \; in 20, 25, 30, 35, 40, 45$ and we vary w from 0 to 0.3 in 100 steps. The results for *Escherichia coli* and $\Delta E = 30$ are shown in Figure 6.8. Additional plots are given in the Appendix in Figure A.5.

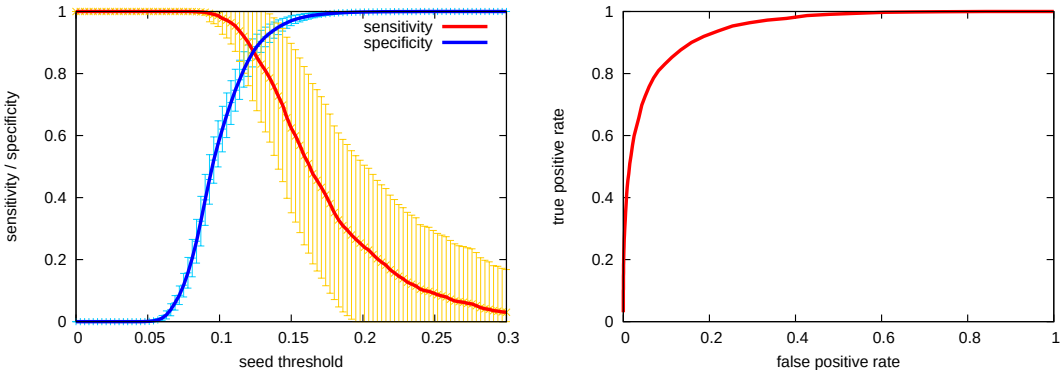

Figure 6.8: Results of the filtration quality experiment for *Escherichia coli* and $\Delta E = 30$. Of the 100 probes 864 showed cross-hybridization at an average of 678 locations. *left:* Sensitivity and specificity versus seed threshold; mean values of the 1000 runs are connected by solid lines; standart deviations are shown in a lighter color. *right:* Corresponding ROC curve.

From the results it can be seen, that for *Escherichia coli* and $\Delta E = 30$ sensitivity does not decline until a seed threshold value of about 0.1. For this value, sensitivity is around 0.98, meaning that 98% of all positions which yield a NNA score below the cross-hybridization threshold have passed the filter. Specificity is around 0.6, which means that 60% of all positions which passed the filter resulted in NNA scores below the cross-hybridization threshold. For lower ΔE sensitivity increases for a constant seed threshold and error bars become larger, as there are less positives. For higher ΔE (an easier to reach

cross-hybridization threshold), performance decreases, as more and more positions lead to a NNA score below the threshold. Similarly, for a constant ΔE, sequences with higher average differences between probe and intendet-target, and probe and not-intended-target scores (compare Figure 6.1) are easier to filter.

6.6 Probe Qualities

In this section we present experiments carried out to compare the cross-hybridization potential with Kane's criteria, and to test our implementation on a large biological example.

6.6.1 CHP and Kane's Criteria

We interpret the NNA scores as lower bound for the free energy of duplex formation and assume that no cross-hybridization occurrs, when the NNA score for a probe p and target stays below T, where $T = nna(p, p) + \Delta E$. If we take this assumption as the truth, we can define false positives and true negatives for probes marked as cross-hybridizing by Kane's criteria. If a probe is a positive according to Kane's criteria (edit distance ≤ 12 (similarity $\geq 76\%$) or a length of the longest common substring ≥ 16 for 50mers), but the NNA score does not fall below the threshold T, this probe is considered a false positive (fp). On the other hand, if both methods agree on not cross-hybridizing, the probe is a true negative (tn). As the NNA score is a lower bound, we cannot realiably say, when a probe is a true positive or false negative accoring to Kane's criteria.

For this experiment we used the same data as in the previous two experiments. For 1000 probes of each genome, we computed cross-hybridization based on Kane's criteria and based on the NNA score. The fp and tn count for the different genomes and ΔE values is given in table 6.1.

Of course, the results strongly depend on the choice of ΔE. A smaller ΔE results in less positives for the NNA criteria, leading to more false positives for Kane's criteria. However, $\Delta E = 30$ can be a reasonable choice for *Mycobacterium bovis*, and in this case, leads to at least 6.4% false positives for Kane's criteria.

Table 6.1: False positives (fp) and true negatives (tn) for Kane's cross-hybridization criteria, given those probes considered negative by the NNA criteria. 1000 random probes versus the whole sequence for *Mycoplasma genitalium* (MG), *Escherichia coli* (EC), *Mycobacterium bovis* (BCG) and a random sequence of length 1 Mbp

ΔE	Sequence							
	MG		EC		BCG		random	
	fp	tn	fp	tn	fp	tn	fp	tn
20	181	781	330	641	530	405	22	978
25	49	401	287	569	492	390	22	962
30	5	59	36	100	64	45	11	426
35	0	0	0	0	0	0	0	7

6.6.2 Probe Qualities for Escherichia coli

To simulate the size of a real world example we computed probe qualities for a large set of probes and a real genome. The probe set consists of 50mers, one starting at every position in the *Escherichia coli* genome. This results in a set of 4,639,626 probes. As target sequence to check for cross-hybridization, we also use the whole 4,639,675 base pair *Escherichia coli* genome, both strands are considered. We use a ΔE of 25 to compute the NNA score cross-hybridization threshold for each probe and a $w = 0.14$ for the computation of the seed threshold for each probe. The task is divided in 100 about equally sized jobs, in every job cross-hybridization potentials for about 46,666 probes are computed. Table 6.2 summarizes the results. Given a ΔE of 25, a total of 527,754 probes show cross hybridization. Of these, a large number has low cross-hybridization potentials and can still be useful depending on the application. Table 6.3 shows the frequency of probes with a low *chp*.

On an Intel Xeon 2.8 GHz CPU the average running time for a job was 4 hours, the fastest job was done in 81 minutes, the slowest in 17 hours. The computation of all probe qualities took 17 CPU days, or 0.318 seconds per probe.

In a real application the user would never compute specificity for all $k$mers of the target sequence. Instead other filters are used beforehand, to reduce the size of the probe candidate set. Nevertheless, this experiment shows, that the algorithm in its implemented form can be applied to large datasets to efficiently compute probe qualities.

Table 6.2: The absolute frequency of cross-hybridization potentials of all 4,639,626 probes used in the experiment.

chp			number of probes
	$chp \leq$	0.00	4111872
0.00	$< chp \leq$	184.76	502356
184.76	$< chp \leq$	369.52	14525
369.52	$< chp \leq$	554.28	3778
554.28	$< chp \leq$	739.04	2390
739.04	$< chp \leq$	923.80	1574
923.80	$< chp \leq$	1108.56	1149
1108.56	$< chp \leq$	1293.33	743
1293.33	$< chp \leq$	1478.09	539
1478.09	$< chp \leq$	1662.85	374
1662.85	$< chp \leq$	1847.61	200
1847.61	$< chp \leq$	2032.37	92
2032.37	$< chp \leq$	2217.13	29
2217.13	$< chp \leq$	2401.89	4
2401.89	$< chp \leq$	2586.65	1

Table 6.3: The absolute frequency of low but non-zero cross-hybridization potentials of all 4,639,626 probes used in the experiment.

chp			number of probes
0.00	$< chp \leq$	5.00	311579
5.00	$< chp \leq$	10.00	36324
10.00	$< chp \leq$	15.00	17252
15.00	$< chp \leq$	20.00	10851
20.00	$< chp \leq$	25.00	17395
25.00	$< chp \leq$	30.00	14020

Chapter 7

Discussion and Outlook

We have presented an efficient method to compute probe specificity for a large set of probes. In contrast to often used simple rules based on edit distance and length of longest common substring, our approach accounts for the more accurate nearest neighbor model without introducing the computational complexity of an optimal thermodynamic alignment. In conjunction with an index-based filter we are able to compute probe specificities quickly without having to scan the whole genome.

The Nearest Neighbor Alignment (NNA) algorithm minimizes the sum of all free energy values of matched pairs. We compute the score using dynamic programming and the well known unified nearest neighbor parameters by SantaLucia (1998) as scoring matrix. Our experiments show that the NNA score has a much higher correlation with free energy of DNA duplex formation than the edit distance. It also shows a higher specificity and sensitivity as a classifier for cross-hybridization than the edit distance and the length of the longest substring combined, as proposed by Kane et al. (2000).

To reduce the search space in the genome for a given probe, we exploit the correlation of short stable matches and low NNA scores between probes and targets, and follow a seed and extend strategy. In order to quickly find these seeds, we use qgram indices for probes and targets. For several sequences, we analyze the filtration ratio as well as the probability of false positives and false negatives in respect to the choice of w as the seed weight threshold. The results show that the filter is able to significantly reduce the number of targets that have to be considered, while maintaining a high sensitivity and reaching a high specificity.

These findings make the NNA algorithm presented here and the weighted seed filter a perfect extension for the tool chain used during probe selection.

When we compare the performance of Kane's simple criteria with the NNA score as a classifier for cross-hybridization, we can see that Kane's criteria lead to more false negatives and false positives, but in general is a good indicator for cross-hybridization. This suggests that it is a good idea to still use these criteria which are fast to compute. For example, for probes of length 50, Kane's constraints could be softened to exclude only probes with matching stretches of length ≥ 19 and an edit distance of ≤ 5. This is a fast method of filtering out probes which obviously cross-hybridize. The remaining probe candidates can then be inspected for cross-hybridization by the computationally more expensiv NNA score, which will have higher accuracy than Kane's original constraints alone.

Future extensions to the NNA algorithm could include a banded version, much like in the FASTA algorithm (Pearson 1990). Using only fields within a narrow band around the diagonal of the scoring matrix should yield to the same or very similar scores for cross-hybridizing sequences. Another future improvement would be the use of gapped seeds for the filter. Allowing a small gap in the seed could improve filtration ratio by retaining sensitivity and specificity, but more research has to be done in this area.

Bibliography

Alberty, R. A. & Silbey, R. J. (1992), *Physical Chemistry*, John Wiley and Sons.

Allawi, H. T. & SantaLucia, J. (1997), 'Thermodynamics and nmr of internal g.t mismatches in dna.', *Biochemistry* **36**(34), 10581–10594.

Allawi, H. T. & SantaLucia, J. (1998*a*), 'Nearest neighbor thermodynamic parameters for internal g.a mismatches in dna.', *Biochemistry* **37**(8), 2170–2179.

Allawi, H. T. & SantaLucia, J. (1998*b*), 'Nearest-neighbor thermodynamics of internal a.c mismatches in dna: sequence dependence and ph effects.', *Biochemistry* **37**(26), 9435–9444.

Allawi, H. T. & SantaLucia, J. (1998*c*), 'Thermodynamics of internal c.t mismatches in dna.', *Nucleic Acids Res* **26**(11), 2694–2701.

Altschul, S. F. (1991), 'Amino acid substitution matrices from an information theoretic perspective.', *J Mol Biol* **219**(3), 555–565.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990), 'Basic local alignment search tool.', *J Mol Biol* **215**(3), 403–410.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997), 'Gapped blast and psi-blast: a new generation of protein database search programs.', *Nucleic Acids Res* **25**(17), 3389–3402.

Borer, P. N., Dengler, B., Tinoco, I. & Uhlenbeck, O. C. (1974), 'Stability of ribonucleic acid double-stranded helices.', *J Mol Biol* **86**(4), 843–853.

Burkhardt, S., Crauser, A., Ferragina, P., Lenhof, H.-P., Rivals, E. & Vingron, M. (1999), q -gram based database searching using a suffix array (QUASAR), *in* 'RECOMB', pp. 77–83.

Chen, Y. A., Chou, C.-C., Lu, X., Slate, E. H., Peck, K., Xu, W., Voit, E. O. & Almeida, J. S. (2006), 'A multivariate prediction model for microarray cross-hybridization.', *BMC Bioinformatics* **7**, 101.

Crothers, D. M. & Zimm, B. H. (1964), 'Theory of the melting transition of synthetic polynucleotides: Evaluation of the stacking free energy.', *J Mol Biol* **9**, 1–9.

Devoe, H. & Tinoco, I. (1962), 'The stability of helical polynucleotides: base contributions.', *J Mol Biol* **4**, 500–517.

Dimitrov, R. A. & Zuker, M. (2004), 'Prediction of hybridization and melting for double-stranded nucleic acids.', *Biophys J* **87**(1), 215–226.

Dunham, I. (2005), Chromosomes 21 and 22: Comparisons, Technical report, The Sanger Institute, Hinxton, UK.

D'yachkov, A. G., Macula, A. J., Pogozelski, W. K., Renz, T. E., Rykov, V. V. & Torney, D. C. (2006), 'New t-gap insertion-deletion-like metrics for dna hybridization thermodynamic modeling.', *J Comput Biol* **13**(4), 866–881.

Fickett, J. W. (1984), 'Fast optimal alignment.', *Nucleic Acids Res* **12**(1 Pt 1), 175–179.

Gräf, S., Nielsen, F. G. G., Kurtz, S., Huynen, M. A., Birney, E., Stunnenberg, H. & Flicek, P. (2007), 'Optimized design and assessment of whole genome tiling arrays.', *Bioinformatics* **23**(13), i195–i204.

He, Z., Wu, L., Li, X., Fields, M. W. & Zhou, J. (2005), 'Empirical establishment of oligonucleotide probe design criteria.', *Appl Environ Microbiol* **71**(7), 3753–3760.

Jokinen, P. & Ukkonen, E. (1991), 'Two algorithms for approximate string matching in static texts.', *Proc. of the 16th Symposium on Mathematical Foundations of Computer Science* **520**, 240–248.

Kaderali, L. (2001), 'Selecting target specific probes for dna arrays.', Diplomathesis, Wirtschafts– und Sozialwissenschaftliche Fakultät, Universität Köln.

Kaderali, L. & Schliep, A. (2002), 'Selecting signature oligonucleotides to identify organisms using dna arrays.', *Bioinformatics* **18**(10), 1340–1349.

Kane, M. D., Jatkoe, T. A., Stumpf, C. R., Lu, J., Thomas, J. D. & Madore, S. J. (2000), 'Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays', *Nucl. Acids Res.* **28**(22), 4552–4557.

Klau, G. W., Rahmann, S., Schliep, A., Vingron, M. & Reinert, K. (2004), 'Optimal robust non-unique probe selection using integer linear programming.', *Bioinformatics* **20 Suppl 1**, i186–i193.

Leber, M., Kaderali, L., Schönhuth, A. & Schrader, R. (2005), 'A fractional programming approach to efficient dna melting temperature calculation.', *Bioinformatics* **21**(10), 2375–2382.

Lee, A. J. T., Lin, C.-W., Lo, W.-H., Chen, C.-C. & Chen, J.-X. (2007), 'A novel filtration method in biological sequence databases', *Pattern Recogn. Lett.* **28**(4), 447–458.

Li, F. & Stormo, G. D. (2001), 'Selection of optimal dna oligos for gene expression arrays.', *Bioinformatics* **17**(11), 1067–1076.

Liebich, J., Schadt, C. W., Chong, S. C., He, Z., Rhee, S.-K. & Zhou, J. (2006), 'Improvement of oligonucleotide probe design criteria for functional gene microarrays in environmental applications.', *Appl Environ Microbiol* **72**(2), 1688–1691.

Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H. & Brown, E. L. (1996), 'Expression monitoring by hybridization to high-density oligonucleotide arrays.', *Nat Biotechnol* **14**(13), 1675–1680.

Markham, N. R. & Zuker, M. (2005), 'Dinamelt web server for nucleic acid melting prediction.', *Nucleic Acids Res* **33**(Web Server issue), W577–W581.

Mockler, T. C., Chan, S., Sundaresan, A., Chen, H., Jacobsen, S. E. & Ecker, J. R. (2005), 'Applications of dna tiling arrays for whole-genome analysis.', *Genomics* **85**(1), 1–15.

Moore, W. J. (1998), *Physical Chemistry*, Longman Publishing Group.

Müller, I. (2001), *Grundzüge der Thermodynamik*, Springer.

NCBI (2001*a*), 'Escherichia coli str. k-12 substr. mg1655, complete genome', GenBank U00096.

NCBI (2001*b*), 'Mycoplasma genitalium g37, complete genome', GenBank L43967.

NCBI (2007), 'Mycobacterium bovis bcg str. pasteur 1173p2, complete genome', GenBank AM408590.

NCBI (2008), 'Homo sapiens chromosome 21 genomic contig, reference assembly', ftp.ncbi.nih.gov/genomes/H_sapiens/CHR_21.

Needleman, S. B. & Wunsch, C. D. (1970), 'A general method applicable to the search for similarities in the amino acid sequence of two proteins.', *J Mol Biol* **48**(3), 443–453.

Pearson, W. R. (1990), 'Rapid and sensitive sequence comparison with fastp and fasta.', *Methods Enzymol* **183**, 63–98.

Pearson, W. R. (1991), 'Searching protein sequence libraries: comparison of the sensitivity and selectivity of the smith-waterman and fasta algorithms.', *Genomics* **11**(3), 635–650.

Pearson, W. R. & Lipman, D. J. (1988), 'Improved tools for biological sequence comparison.', *Proc Natl Acad Sci U S A* **85**(8), 2444–2448.

Pozhitkov, A. E. & Tautz, D. (2002), 'An algorithm and program for finding sequence specific oligonucleotide probes for species identification.', *BMC Bioinformatics* **3**, 9.

Rahmann, S. (2003), 'Fast large scale oligonucleotide selection using the longest common factor approach', *Journal of Bioinformatics and Computational Biology* **1**(2), 343–361.

Rash, S. & Gusfield, D. (2002), String barcoding: Uncovering optimal virus signatures, *in* 'Proceedings of RECOMB 2002', pp. 254–261.

Reymond, N., Charles, H., Duret, L., Calevro, F., Beslon, G. & Fayard, J.-M. (2004), 'Roso: optimizing oligonucleotide probes for microarrays.', *Bioinformatics* **20**(2), 271–273.

Rouillard, J.-M., Herbert, C. J. & Zuker, M. (2002), 'Oligoarray: genome-scale oligonucleotide design for microarrays.', *Bioinformatics* **18**(3), 486–487.

Rouillard, J.-M., Zuker, M. & Gulari, E. (2003), 'Oligoarray 2.0: design of oligonucleotide probes for dna microarrays using a thermodynamic approach.', *Nucleic Acids Res* **31**(12), 3057–3062.

SantaLucia, J. (1998), 'A unified view of polymer, dumbbell, and oligonucleotide dna nearest-neighbor thermodynamics.', *Proc Natl Acad Sci U S A* **95**(4), 1460–1465.

SantaLucia, J. & Hicks, D. (2004), 'The thermodynamics of dna structural motifs.', *Annu Rev Biophys Biomol Struct* **33**, 415–440.

Schliep, A. & Rahmann, S. (2006), 'Decoding non-unique oligonucleotide hybridization experiments of targets related by a phylogenetic tree.', *Bioinformatics* **22**(14), e424–e430.

Schliep, A., Torney, D. C. & Rahmann, S. (2003), Group testing with DNA chips: Generating designs and decoding experiments, *in* 'Proceedings of the 2nd IEEE Computer Society Bioinformatics Conference (CSB 2003)', IEEE, pp. 84–93.

Sugimoto, N., Nakano, S., Katoh, M., Matsumura, A., Nakamuta, H., Ohmichi, T., Yoneyama, M. & Sasaki, M. (1995), 'Thermodynamic parameters to predict stability of rna/dna hybrid duplexes.', *Biochemistry* **34**(35), 11211–11216.

Sugimoto, N., Nakano, S., Yoneyama, M. & Honda, K. (1996), 'Improved thermodynamic parameters and helix initiation factor to predict stability of dna duplexes.', *Nucleic Acids Res* **24**(22), 4501–4505.

Sung, W.-K. & Lee, W.-H. (2003), 'Fast and accurate probe selection algorithm for large genomes.', *Proc IEEE Comput Soc Bioinform Conf* **2**, 65–74.

Takai, D. & Jones, P. A. (2002), 'Comprehensive analysis of cpg islands in human chromosomes 21 and 22.', *Proc Natl Acad Sci U S A* **99**(6), 3740–3745.

Ukkonen, E. (1985), 'Algorithms for approximate string matching', *Inf. Control* **64**(1-3), 100–118.

Ukkonen, E. (1992), 'Approximate string-matching with q-grams and maximal matches', *Theor. Comput. Sci.* **92**(1), 191–211.

Waterman, M. S. (1989), *Mathematical Methods for DNA Sequences*, CRC Press, Inc., Boca Raton, FL, USA.

Watson, J. D. & Crick, F. H. (1953), 'Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid.', *Nature* **171**(4356), 737–738.

Wilkins, M. H. F., Stokes, A. R. & Wilson, H. R. (1953), 'Molecular structure of deoxypentose nucleic acids.', *Nature* **171**(4356), 738–740.

Zhang, L., Wu, C., Carta, R. & Zhao, H. (2007), 'Free energy of dna duplex formation on short oligonucleotide microarrays.', *Nucleic Acids Res* **35**(3), e18.

Zuker, M. & Stiegler, P. (1981), 'Optimal computer folding of large rna sequences using thermodynamics and auxiliary information.', *Nucleic Acids Res* **9**(1), 133–148.

Appendix A

Appendix

A.1 Possible Seeds

Table A.1: Frequency of weighted seeds. Weight count versus length q.

q	weight											
	< 2	< 4	< 6	< 8	< 10	< 12	< 14	< 16	<18	< 20	< 22	< 24
11	0	0	0	236	39724	518955	1528402	1487214	544514	72539	2718	2
10	0	0	0	1503	62007	318853	436273	198907	29797	1236	0	0
9	0	0	10	4620	55850	118672	69904	12550	538	0	0	0
8	0	0	160	7229	28776	23947	5205	219	0	0	0	0
7	0	0	542	5898	7679	2153	112	0	0	0	0	0
6	0	12	862	2284	898	40	0	0	0	0	0	0
5	0	72	570	356	26	0	0	0	0	0	0	0
4	0	104	144	8	0	0	0	0	0	0	0	0
3	10	48	6	0	0	0	0	0	0	0	0	0
2	14	2	0	0	0	0	0	0	0	0	0	0

A.2 NNA Score Compared to Edit Distance

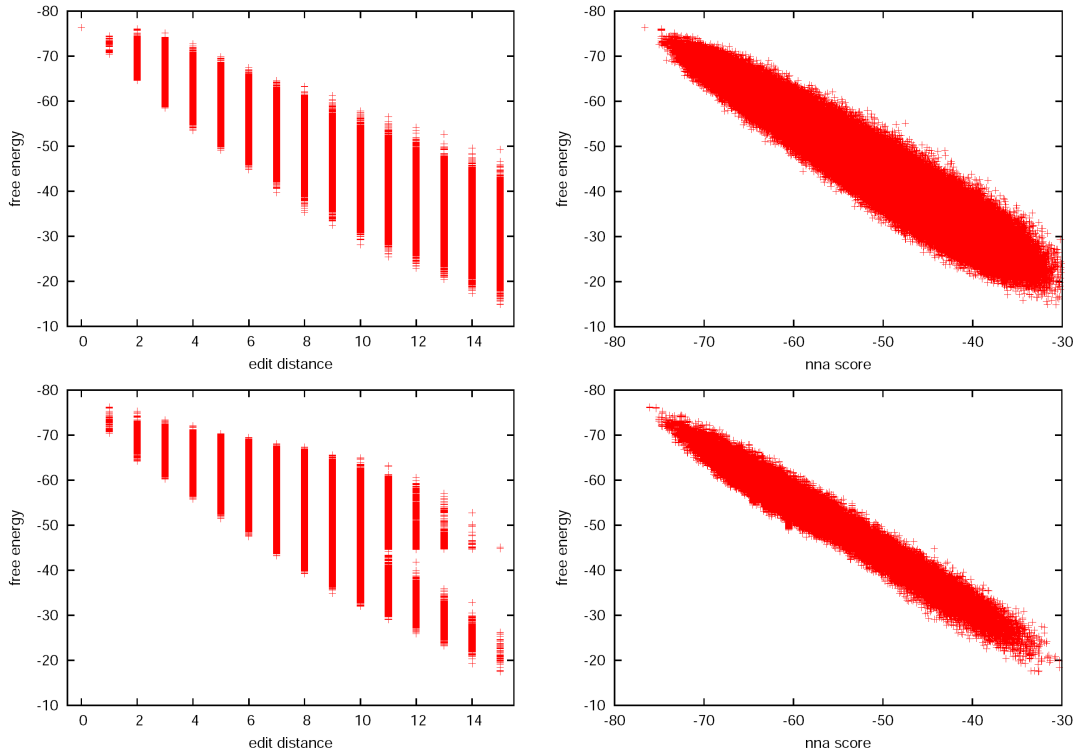

Figure A.1: Edit distance (*left*) and NNA score (*right*) versus free energy for the **random** dataset (*top*) and the **placed** dataset (*bottom*)

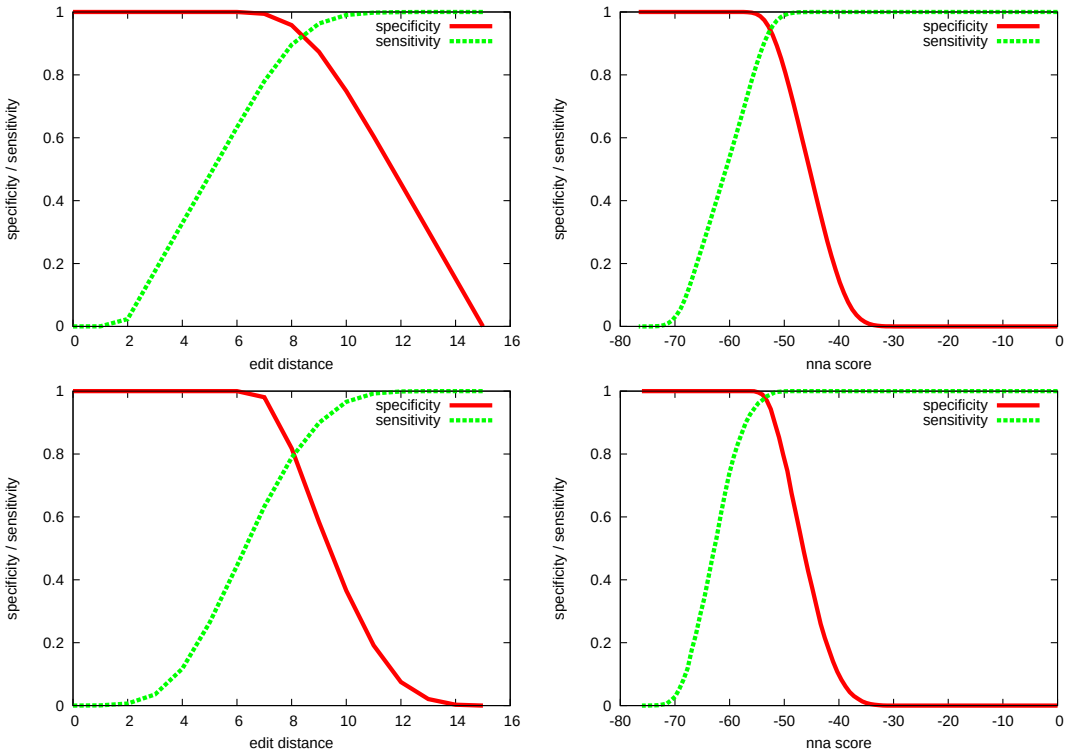

Figure A.2: Specificity and sensitivity of edit distance and NNA score as classifier for cross-hybridization. *top:* **random** dataset *bottom:* **placed** dataset.

A.3 NNA Scores Versus Kane's Criteria

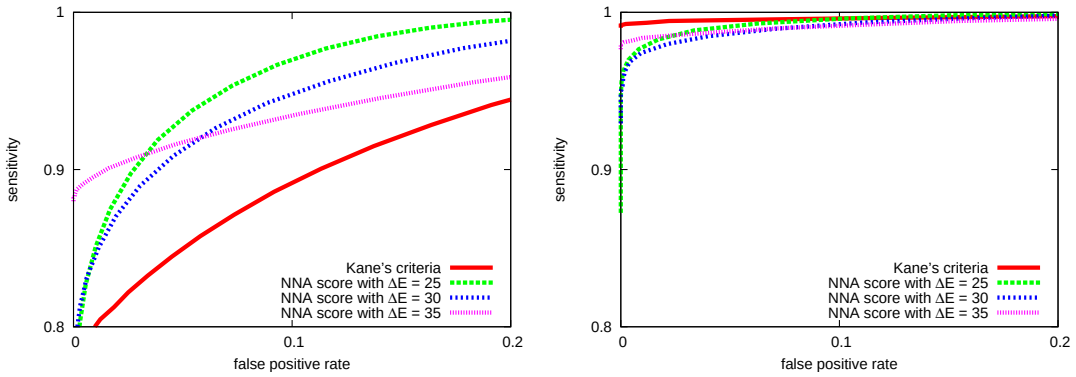

Figure A.3: ROC curves for the **random** dataset (left) and the **placed** dataset.

A.4 Filtration Ratio

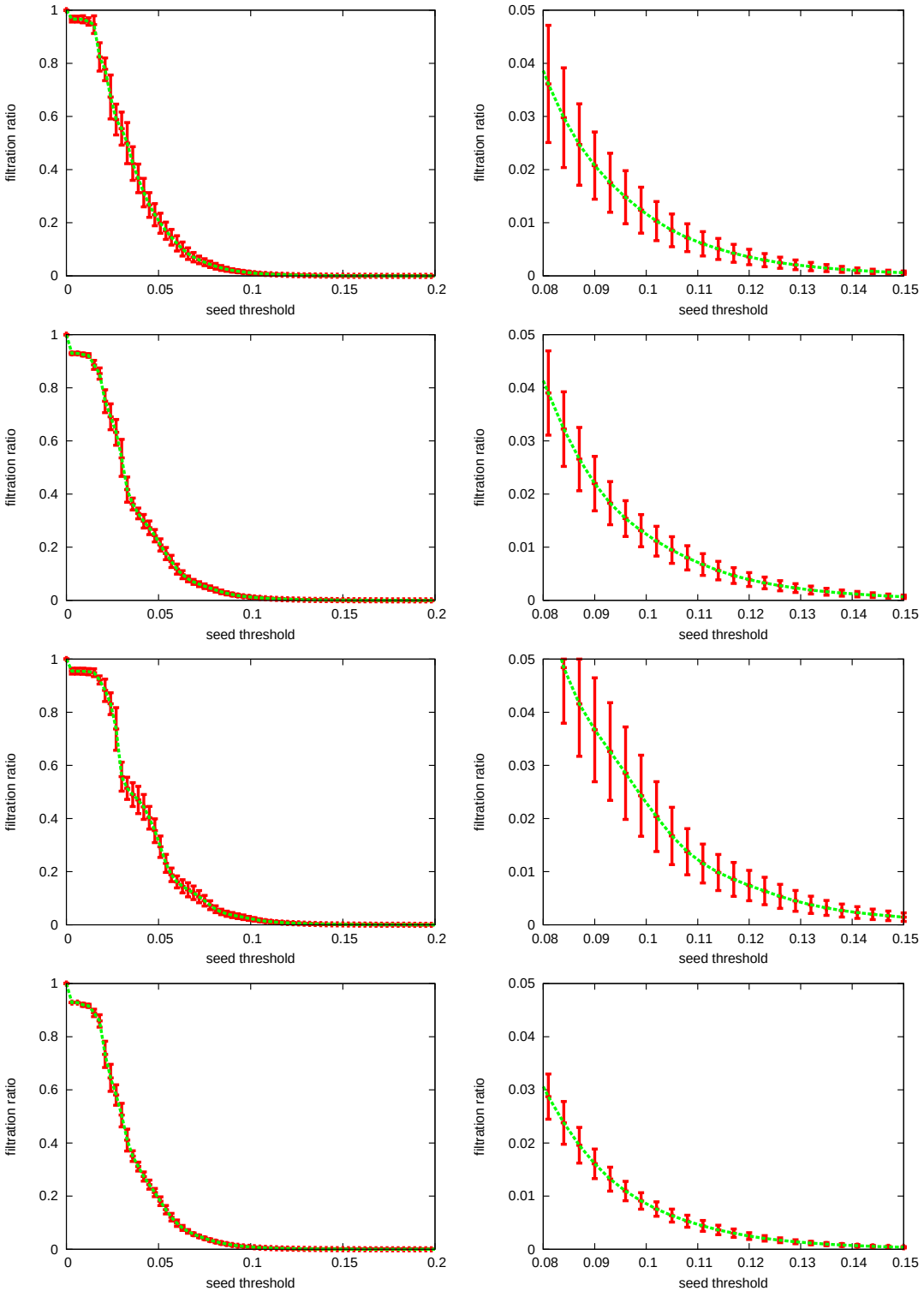

Figure A.4: Filtration ratios for different seed thresholds and different sequences. Mean and standard deviation of 1000 runs. Top to bottom: *Mycoplasma genitalium*, *Escherichia coli*, *Mycobacterium bovis*, a random sequence of length 1 Mbp

A.5 Filtration Quality

Figure A.5 shows the results for additional datasets. The experiment is described in section 6.5.2.

Table A.2: Number of probes showing cross-hybridization; for these probes, in brackets the average number of positions where cross-hybridization occurs is given.

sequence	ΔE									
	20		25		30		35		40	
Mycoplasma genitalium	38	(44)	550	(82)	936	(8,200)	1000	(109,666)	1000	(372,511)
Escherichia coli	29	(101)	144	(50)	864	(678)	1000	(39,838)	1000	(541,862)
Mycobacterium bovis	65	(28)	118	(35)	891	(47)	1000	(2,974)	1000	(119,355)
random sequence	0	(0)	16	(5)	563	(72)	993	(4,825)	1000	(103,361)

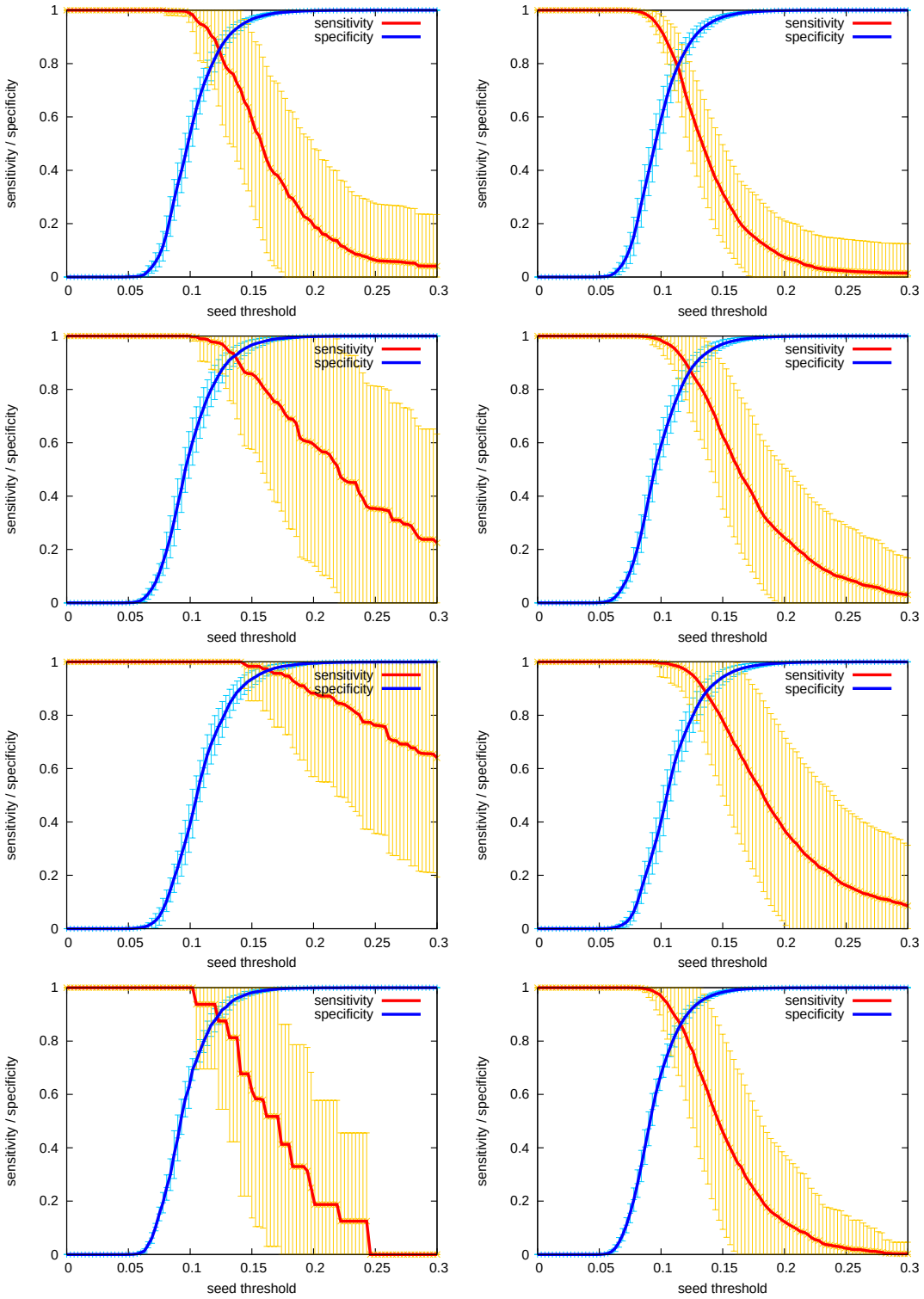

Figure A.5: Results of the filtration quality experiment for top to bottom: *Mycoplasma genitalium*, *Escherichia coli*, *Mycobacterium bovis*, a random sequence of length 1 Mbp and $\Delta E = 25$ (left) and $\Delta E = 30$ (right).

A.6 The Software

The NNA algorithm and the filtering procedure were implemented in ansi C (ISO C90) on a PC running Linux. The source code will be made available by the author.

```
Usage: ./proqual probe_file sequence_file [options]

probe_file: fasta compatible file containing probe candidates
sequence_file: fasta compatible file containing target sequences

Options:
-p, probe range starting with zero (optional, if omitted all probes are used) = [first probe] [last probe+1]
-deltae, delta E for cross hybridization threshold = [deltae] as min difference in binding energy
-minw, minimum weight of seeds = [minimum weight] as fraction of probe target binding energy
-maxq, maximum length of seeds = [maximum length] in bases
-s, strands to check
        [0] = check for cross hybridization to forward strand
        [1] = check for cross hybridization to both strands
        [2] = check for cross hybridization to reverse strand

Example:
./proqual genome_candidates.fa genome.fa -deltae 30.0 -minw 0.10 -s 1 -maxq 10
```

For the usage example above, the output file will be

```
genome_candidates.chp


6702  0.000000
6703  0.000000
6704  3.700008
6705  3.700008
6706  8.130013
6707  11.940048
6708  17.050079
```

with one line for each processed probe giving the number of the probe and its cross-hybridization potential.