Efficient algorithms for the computational design of optimal tiling arrays

Alexander Schliep and Roland Krause

(Invited Paper)

***Abstract*— The representation of a genome by oligonucleotide probes is a prerequisite for the analysis of many of its basic properties, such as transcription factor binding sites, chromosomal breakpoints, gene expression of known genes and detection of novel genes, in particular those coding for small RNAs. An ideal representation would consist of a high density set of oligonucleotides with similar melting temperatures that do not cross-hybridize with other regions of the genome and are equidistantly spaced. The implementation of such a design is typically called a tiling array or genome array.**

We formulate the minimal cost tiling path problem for the selection of oligonucleotides from a set of candidates. Computing the selection of probes requires multi-criterion optimization, which we cast as a shortest path problem. Standard algorithms running in linear time allow us to compute globally optimal tiling paths from millions of candidate oligonucleotides on a standard desktop computer for most problem variants. The solutions to this multi-criterion optimization are spatially adaptive to the problem instance. Our formulation incorporates experimental constraints with respect to specific regions of interest and trade-offs between hybridization parameters, probe quality and tiling density easily. A webapplication is available at `http://tileomatic.org`.

***Index Terms*— tiling arrays, micro arrays, genomics, bioinformatics**

I. Introduction

GENOME sequences of all major model organisms and many other relevant species have been decoded and enable detailed studies using many different laboratory and computational approaches. Microarrays have become a crucial tool for the study of dynamic properties of the genome, and with recent technological advances, can represent the complete genome of an organism with high coverage for thorough investigation.

Initial designs focused on the representation using a single probe for each predicted gene. For the study of gene expression of protein coding genes, in particular in prokaryotes or lower eukaryotes such as yeast, this is a feasible approach that has become a convenient standard in many laboratories.

For studies of, for example, transcription factor binding sites or chromosomal breakpoints, this representation is very limiting and ideally, one would like to have the complete view of the genome.

Microarrays that contain the required dense representation of the genome or larger blocks thereof are typically referred to as tiling arrays. Early arrays were constructed from PCR fragments, which is laborious and was not extended to complete eukaryotic genomes with high coverage. Several synthesis technologies such as utilizing maskless photolithography or inkjet printing are becoming accessible at low costs from commercial vendors; some companies offer custom designs that allow different probe sets for every array manufactured.

Manuscript received January 20, 2002; revised August 13, 2002.
Both authors are with the Max Planck Institute for Molecular Genetics.

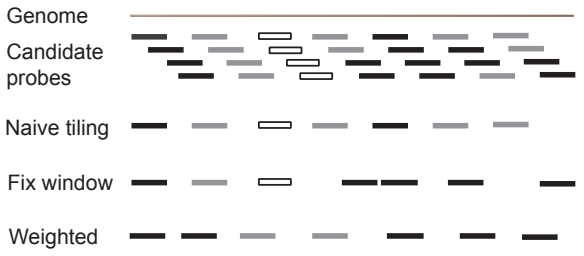


Fig. 1. Strategies for the design of genome or tiling arrays. Probe quality is denoted in color with black bars representing suitable oligonucleotides and white ones those with adverse properties of their sequence. In the naive approach, every 4th oligonucleotide is selected, the windowed approach selects the best oligonucleotides within a window of 4. The weighted approach, which is described in this work, has an optimal distance $d^\star$ of 4. The selection balances oligonucleotide quality and spacing and can avoid stretches of oligonucleotides of low quality while preserving a more homogeneous selection than the fixed window approach.

The primary applications for tiling arrays are found in DNA-protein immunoprecipitation experiments [1] and the analysis of chromosomal breakpoints [2]. Notably, discovery of the many small RNAs in recent years would have not been feasible without the access to tiling arrays [3]. Recently, the ENCODE project [4] studied 1% of the human genomes in depth and showed that many areas of the genome are active that were considered to be irrelevant prior to the study because they lie outside the regions of, typically, protein coding genes and are not conserved. Other relevant applications include the analysis of nucleosome positioning [5], DNAse I sensitivity [6] and, very recently, selection of genomic regions for high-throughput resequencing [7], [8]. It seems very likely that these devices can be used for a multitude of additional applications [9].

A. Design of tiling arrays

Depending on the biological questions and access to technologies, different designs for tiling arrays can be considered. High density arrays often neglect considerations of probe suitability and select probes by simple criteria, which do require only trivial algorithms.

In the following, we are mainly concerned with spaced tiling arrays, which select a tiling path amongst previously generated candidate probes. See Figure 1 for an overview of strategies for oligonucleotide selection. Note that probes can in principle be selected to overlap in the method we describe later; however, most prior approaches utilized a fixed window size in which to place a single oligonucleotide probe without overlaps.

Beyond considering the selection of probes by desired probe distance d given in Figure 1, important parameter for spaced

Digital Object Indentifier 10.1109/TCBB.2008.50

oligonucleotide tiling arrays also include its melting temperature T_m, and the potential for cross-hybridization, binding to unintended loci, indicated by the probe quality q.

The melting temperature T_m of a given probe can be estimated reasonably well using the nearest neighbor model [10]. The selection of an iso-thermal design of probes with homogeneous T_m would be desirable.

Cross-hybridization of probes is a problem for all applications that rely on pairing of DNA strands, not only microarrays, as it compromises experimental results by either false positives or, in the case of replications, increased variance. Many studies propose easily computable criteria under which a probe binds only to its reverse complement and not to other loci. For oligonucleotide probes of length 50, Kane *et al*. [11] measured the level of cross-hybridization and summarized the results into two criteria for unique probe selection. First, a probe shall not have more than 75% sequence identity with any fifty contiguous bases outside its intended loci. Second, for all such 50mers outside its intended loci with a sequence identity between 50% to 75% there may not be a contiguous exact match of 15 bases or more.

Previous approaches for the design of tiling arrays have typically focused on one individual criterion. A flexible formulation of this multi-criterion optimization problem is the major advancement that we report in this work after a review of previously described methods for the design of optimal tiling arrays.

B. Prior work

As one of the first, Selinger *et al*. described a tiling array for the analysis of the transcriptome of *E. coli* that was designed by selecting oligonucleotides at every 6th base pair for intergenic regions and every 60th base pair in coding regions [12]. This approach sacrifices probe quality with respect to the potential for cross hybridization and uniformity of melting temperatures for high density and simplicity. Similar approaches were used for the small human chromosomes in 2002 [13] and recently for the whole human genome [14].

The design of these naive arrays is simple. If wider spacings of individual probes are employed, more care can be taken to select unique probes with homogeneous melting temperatures and other recognized properties of quality. To obtain arrays suitable for comparative genomic hybridization using microarrays (aCGH), Lipson *et al*. [15] propose to choose a whenever possible (WP) ϵ-cover after the computation of candidate probes. A WP ϵ-cover is a subset of candidates so that for any chromosomal position x the following holds. Either there are probes i, j, $i \leq x \leq j$ (probes are identified with their chromosomal location in contrast to our notation) in the cover with $j - i \leq \epsilon$ or there is no candidate between i and j. They provide a greedy algorithm for computation of WP ϵ-covers and minimizing ϵ for a given array size with a log-linear complexity of their entire approach. For a problem instance with candidates at positions $1, 2, \ldots, \epsilon, \epsilon+1, 2\epsilon+1, 2\epsilon+2, \ldots, 3\epsilon+1, 3\epsilon+2, 4\epsilon+2$ the greedy algorithm will arrive at the WP ϵ-cover $\epsilon, \epsilon+1, 2\epsilon+1, 3\epsilon+1, 3\epsilon+2, 4\epsilon+2, \ldots$. This is undesirable as one third of the probes are essentially uninformative and for example $\epsilon+1, 2\epsilon+1, 3\epsilon+2, 4\epsilon+2$ uses fewer probes to cover the same segment with the same resolution. Furthermore, their approach cannot optimize with respect to individual probe quality.

Eukaryotic genomes contain a large number of repetitive elements, which pose a significant complication to microarray design. The individual types of repeats, their origins and properties are an interesting and challenging aspect of genome organization, which we cannot elaborate here in detail. Please see [16] for a introductory review. Initial design strategies circumvented the problem by tiling only the area of the genome outside of known repetitive elements, which are usually obtained by tools such as RepeatMasker [17]. If one is interested in a small region of a genome, this strategy can yield meaningful results for tiling array designs [18]. For a whole genome tiling array, the matter cannot be circumvented this easily as repetitive elements comprise more than 50% of the human genome [19], are actively transcribed and play a role in many processes linked to diseases. One pays a high price to the coverage of the genome by ignoring these features. Even in prokaryotes large duplicated regions exist. For instance the area around the origin of replication in *Mycobacterium bovis* BCG is present in a second copy [20]. Simply ignoring the fact would lead to unnecessary gaps in the coverage if only unique elements in the genome are considered as candidates.

To this end, Bertone *et al*. introduced an approach using dynamic programming that builds a tiling path in linear time [21]. Repeated regions are explicitly addressed by creating tiles of genomic sequences by joining segments of non-repetitive DNA if they are separated by short segments of repetitive DNA only. However, this approach ignores larger repeat-regions by design and does not address individual quality of a given probe.

To further refine the cross-hybridization potential of a given probe, Gräf *et al* [22] devised a scheme that identifies unique substrings and selects maximally unique probes within a given window of the length h. The uniqueness score $U(s, r)$ for a probe p in position r is the number of minimum unique substrings s of a length $< k$ ending in a particular position within h. The score can be efficiently computed without relying on computing individual approximate matches of candidates with the genome.

This can be used to assess the relative similarity of a probe and select an optimal probe within a window of a given size. If none of the candidate probes passes the quality criteria set, no probe will be placed in the window. This potentially leads to handling of large regions in the genome without representation.

As we abandon a fixed window size for the computation of the optimal tiling path this problem should not occur in our approach.

C. Our contribution

We formalize the design of tiling arrays for complete genomes of prokaryotes and higher eukaryotes as the Minimal Cost Tiling Path Problem (MCTPP) and suggest efficient algorithms for its solution. This work focuses on how to overcome several of the technical challenges in the generation of the candidate sequences for larger genomes and the linear time tiling path calculation using Monge theory.

Unlike other approaches with fixed windows of a given length, we let users specify a desired distance $d^\star$, which allows for more flexibility in the treatment of repeats and quality parameters for the design. The spacing will deviate from $d^\star$ if a greater homogeneity of hybridization conditions can be realized or higher quality probes can be chosen.

The variable spacing in our approach is in contrast to most of the prior work, which predominantly used equidistant probes as this simplifies design *and* analysis. Spatial correlations between hybridization intensities of adjacent probes can be ignored, if those correlations are all constant, as is in the case of equidistant probes. Effectively, hybridization intensities can be modeled as

(conditionally) independent random variables. The variable spatial correlations due to the variable-spaced probes in our approach need to be accounted for in the analysis. This can be achieved with extensions to Hidden Markov Models which are beyond the scope of this manuscript. Also, the non-unique oligonucleotides [23] in repeat regions need some care in the analysis.

The two major steps of our method are generation of a set of candidate oligonucleotides and selection of probes by computation of the minimal cost tiling path from the candidates. The candidate generation consists of preprocessing the genome with respect to repetitive elements, which are masked in the process. An enumeration of all possible probes that pass certain criteria and calculate properties such as T_m, and cross-hybridization potential as quality q constitutes the candidate set, which may contain probes that need to be filtered for additional adverse properties, for instance palindromic sequence.

The major focus of this work is on the creation of a minimal cost tiling path over the candidates. We present two approaches to the problem, a linear time variant that utilizes properties of Monge matrices and a more general approach build around shortest path algorithms.

Our goal is to facilitate the design of tiling arrays with probes of the length l = 40 to 70, which are typically manufactured from *in situ* DNA synthesis or piezoelectric printing. Our approach was first applied to the design of a 44,000 spot tiling array for the 6.9 Mbp genome of *Mycobacterium smegmatis*, which was produced and used in the laboratory. Initial experiments gave positive results and will be followed up; they already showed the applicability of our approach.

We start with the formulation of the tiling path and the algorithms to its calculation and discuss the necessary candidate generation subsequently.

II. The minimal cost tiling path problem

The three fundamental criteria in the design of tiling arrays are the distance between adjacent oligonucleotide probes, their potential for cross-hybridization and their hybridization conditions. Obviously, the criteria are conflicting. Selecting oligonucleotide probes at every kth positions on a chromosome will lead to wide variations in quality and hybridization conditions. Recall that for example GC-content correlates with melting temperature. Conversely, selecting only unique probes will necessarily leave large gaps in the tiling. First, we will introduce design parameters which reflect the user's choice for the tiling array neglecting the reality of the genome's particularities. We concentrate on three design parameters in the following, even though the formulation extends to further quantities naturally. Also, we present the formulation for a single chromosome as tiling paths for organisms with multiple chromosomes can be computed individually per chromosome. Given a probe i we denote by $p(i)$ its chromosomal position, by $q(i)$ its quality which is computed in the candidate generation and for which higher values indicate a lower potential for cross-hybridization—note that effective, efficiently computable measures of cross-hybridization potential are still under discussion—and, as one relevant hybridization condition, the melting temperature $T_m(i)$ of probe i in a perfect duplex; that is to its reverse complement. Now we can specify what a tiling should look like: $d^\star$ specifies the desired distance of adjacent oligonucleotide probes in the tiling, $q^\star$ the desired probe quality with respect to its potential for cross-hybridization and by $T_m^\star$ the desired melting temperature. Clearly, $d^\star$, $q^\star$ and $T_m^\star$ will be positive; we also assume that probes are sorted according to their position so that $i < j$ implies $p(i) < p(j)$.

It is unlikely that we can obtain a tiling of a *real* genome exactly matching our design parameters, but we can easily quantify how much a given tiling deviates from the specified parameters. We define penalties for deviations in such a way that we obtain unit-free quantities which are on the same scale for the three parameters: the distance penalty for $i < j$

$$d(i,j) := \frac{|d^\star - (p(j) - p(i))|}{d^\star}, \tag{1}$$

the quality penalty, which only penalizes quality values below $q^\star$,

$$pq(j) := \begin{cases} \dfrac{q^\star - q(j)}{q^\star} & \text{if } q(j) < q^\star, \text{ and} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and the melting temperature penalty

$$pt(j) := \frac{|T_m^\star - T_m(j)|}{T_m^\star}. \tag{3}$$

For example $d(i,j)$ will contribute unit penalty if i and j are of distance $2d^\star$ and a penalty of almost one, $\dfrac{d^\star - 1}{d^\star}$, if the two probes have distance one; $d(i,j)$ equals zero if and only if probes i and j are exactly $d^\star$ nucleotides apart.

Definition 1: A *tiling path* T is a subsequence of the candidate probe sequence $1, \ldots, n$.

We define $d(0,i)$ and $d(i,\infty)$ to penalize excessive distance between the chromosomal start and end and the first, respectively last, oligonucleotide probe on the tiling path; that is, $d(0,i) = \dfrac{p(i) - d^\star}{d^\star}$ for $p(i) > d^\star$ and 0 else and similarly for $d(i,\infty)$. If we introduce the convention that $T_0 = 0$ and $T_i = \infty$ for $i > |T|$ we arrive at a cost function quantifying the deviation from the specified design parameters

$$C(T) = \sum_{i=0}^{|T|} d(T_i, T_{i+1}) + \sum_{i=1}^{|T|} pt(T_i) + \sum_{i=1}^{|T|} pq(T_i). \tag{4}$$

If the design parameters can be exactly realized for a given chromosome, the tiling path will have cost zero. Deviations of a few nucleotides in distance will occur if higher quality oligonucleotides can be selected. As both maximal quality and melting temperature penalties are comparatively small, for example the range of melting temperature for all candidates in an organism might be 45° C to 105° C, they are dominated by the distance penalty if there are no candidate probes in a region.

A. Constrained solutions and inhomogeneous costs

The cost function accommodates constraints for optimal solutions. This is of high interest for applications as custom tiling arrays in contrast to ready-made high density arrays are used for focused studies. A weighted version of the cost function allows us to globally change the trade-off between the different penalties, yielding the following variant of the right hand side of (4):

$$\sum_{i=0}^{|T|} d(T_i, T_{i+1}) + \sum_{i=1}^{|T|} w_t \cdot pt(T_i) + \sum_{i=1}^{|T|} w_q \cdot pq(T_i). \tag{5}$$

Here w_q and w_t denote the weights for the quality and melting temperature penalties in relation to the distance penalty, for which we can assume unit weight without loss of generality.

Further fine-grained control over the solution is given by *inhomogeneous* weights $w_d(x) > 0$, $w_q(x) > 0$ and $w_t(x) > 0$ which depend on the chromosomal location x and a location-dependent distance design parameter $d^\star(x) > 0$ which yields a location-dependent penalty

$$\widetilde{d}(i,j) := \frac{|d^\star(p(i)) - (p(j) - p(i))|}{d^\star(p(i))}, \tag{6}$$

and the cost function

$$C_w(T) = \sum_{i=0}^{|T|} w_d(p(T_i)) \cdot \widetilde{d}(T_i, T_{i+1}) + \sum_{i=1}^{|T|} w_t(p(T_i)) \cdot pt(T_i) + \sum_{i=1}^{|T|} w_q(p(T_i)) \cdot pq(T_i). \tag{7}$$

Lastly, for comparison with previous experimental designs and standardization or to target specific positions such as exon boundaries [24] the inclusion of *obligatory oligonucleotides* is another constraint which is important in applications. Let S be a subsequence of $1, \ldots, n$ specifying obligatory oligonucleotides which must be used. The cost of a tiling path T for which $S \subset T \subset 1, \ldots, n$ holds is

$$C_w(T;S) = \sum_{\substack{i=0 \\ T_i \notin S \vee T_{i+1} \notin S}}^{|T|} w_d(p(T_i)) \cdot \widetilde{d}(T_i, T_{i+1}) + \sum_{\substack{i=1 \\ T_i \notin S}}^{|T|} w_t(p(T_i)) \cdot pt(T_i) + \sum_{\substack{i=1 \\ T_i \notin S}}^{|T|} w_q(p(T_i)) \cdot pq(T_i). \tag{8}$$

Note, that $C_w(T) = C_w(T;\{\})$. We can now formulate the problem we consider.

Problem 1 (Minimal cost tiling path problem (MCTPP)): Find a tiling path T of minimal cost $C_w(T)$ given candidate probes $1, \ldots, n$, a possibly empty set of obligatory oligonucleotides S, probe parameters $p(i), T_m(i)$ and $q(i)$ and design parameters $d^\star(x), T_m^\star$ and $q^\star$ with criteria weights $w_q(x)$, $w_t(x)$ and $w_q(x)$ dependent on chromosomal position x.

If all the weights are homogeneous, that is $w_d(x) \equiv w_d = 1$ without loss of generality, $w_q(x) \equiv w_q$, $w_t(x) \equiv w_t$ and $d^\star(x) \equiv d^\star$ independent of position x, we speak of the *homogeneous MCTPP*. If additionally $w_t = w_q = 1$ we speak of the *unweighted MCTPP*. Similarly we speak of the *distance-homogeneous MCTPP*.

Note that the penalties defined for $d(0,i)$ and $d(i,\infty)$ preclude trivial solutions. As a matter of fact, the *expected* length of the path, that is the expected length of the subsequence and the number of oligonucleotide probes in the tiling, is $\frac{p(n)}{d^\star}$ in the homogeneous case.

III. Algorithmic solutions

The Minimal cost tiling path problem (MCTPP) is a multi-criterion optimization problem in which the user specifies the trade-offs between criteria. This makes the problem tractable and efficient algorithms can be found by exploiting the relationship to shortest paths algorithms which can be solved with standard algorithms and, for a particular case, with Monge theory. For expository reasons we will use the shortest path formulation even though Monge theory does consider dynamic programming problems in general; furthermore, the MCTPP is equivalent to a minimal weight sub-sequence problem.

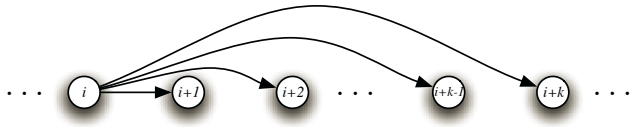


Fig. 2. We show the neighborhood structure for node i. It is adjacent to nodes $i+1, i+2, \ldots, i+k$. Other edges are not shown.

We solve the minimal cost tiling path problem (MCTPP) through a reformulation as a shortest path problem on a specific class of digraphs. Let $V = \{0, 1, \ldots, n, \infty\}$ be the set of vertices corresponding to the n probe candidates and the chromosome start and end, denoted 0 and ∞ respectively. The set of edges is $E = \{(i,j) | i < j\} \setminus \{(i,j) | \exists s \in S : i < s < j\}$ where S denotes the set of obligatory oligonucleotides. We assign a weight to an edge (i,j) corresponding to the terms contributed to (9) by choosing probe j following probe i on the tiling path, that is

$$w(i,j) := w_q(p(i)) \cdot \widetilde{d}(i,j) + w_t(p(j)) \cdot pt(j) + w_q(p(j)) \cdot pq(j). \tag{9}$$

We follow the usual convention that $w(i,j) = \infty$ for $(i,j) \notin E$. We call $G = (V, E)$ an MCTPP instance graph.

Proposition 1: A shortest $0, \infty$-path in G gives an optimal solution of the minimal cost tiling path problem.

The order of G is $n+2$ and its size is at most, for empty S, $\frac{(n+2)(n+1)}{2}$. Note that the *relevant* input size is the number of candidate oligonucleotide probes, n, and that the graph is only a conceptual construct to facilitate computation and its size should not appear in the complexity analysis. That is, a linear time solution to the MCTPP should be linear in n alone.

A. Monge theory

The structure of the cost function, in particular the dominating and monotonous contribution of the distance penalty suggests that efficient algorithms should exist. In fact, improvements for dynamic algorithms for special qualitative cases such as convexity or concavity of cost functions have been widely studied [25] and we can obtain a linear-time algorithm from the theory of Monge matrices; see for example Burkard *et al.* [26] for a review. Monge matrices are also called totally monotone matrices [26].

Definition 2: A matrix C is called an upper triangular Monge matrix, if for all integers i, r, j, s with $1 \le i < r \le j < s \le n$ the following condition holds:

$$c_{ij} + c_{rs} \le c_{is} + c_{rj}. \tag{10}$$

Note, that $c_{ij} \in \mathbb{R} \cup \{\infty\}$.

If we insert the definition of (9) in the Monge condition (10), we obtain

$$\begin{aligned} & w_d(p(i)) \cdot \widetilde{d}(i,j) + w_t(p(j)) \cdot pt(j) + w_q(p(j)) \cdot pq(j) \\ + \; & w_d(p(r)) \cdot \widetilde{d}(r,s) + w_t(p(s)) \cdot pt(s) + w_q(p(s)) \cdot pq(s) \\ \le \; & w_d(p(i)) \cdot \widetilde{d}(i,s) + w_t(p(s)) \cdot pt(s) + w_q(p(s)) \cdot pq(s) \\ + \; & w_d(p(r) \cdot \widetilde{d}(r,j) + w_t(p(j)) \cdot pt(j) + w_q(p(j)) \cdot pq(j). \end{aligned} \tag{11}$$

* Note that the quality and temperature penalties are identical on both sides and hence cancel, leaving us with

$$\begin{aligned} & w_d(p(i)) \cdot \widetilde{d}(i,j) + w_d(p(r)) \cdot \widetilde{d}(r,s) \\ \leq \quad & w_d(p(i)) \cdot \widetilde{d}(i,s) + w_d(p(r \cdot \widetilde{d}(r,j). \end{aligned} \tag{12}$$

If we assume that the MCTPP is distance-homogeneous and distance-unweighted, so $d^\star(x) = d^\star$ and $w_d(x) \equiv 1$, we can show the following lemma.

Lemma 1: The matrix $W = \{w(i,j)\}_{0 \leq i,j \leq n+1}$ of edge weights of a distance-unweighted and distance-homogeneous MCTPP instance graph, where $w(i,j)$ is defined as in (9), is an upper triangular Monge matrix.

Proof: First observe that the Monge condition is satisfied if any of the $w(\cdot,\cdot)$ equals ∞. In this case there must be at least one obligatory oligonucleotide and due to the definition of the graph, there actually must be an infinite weight of both sides of the inequality.

Now let i, r, j, s be integers such that $1 \leq i < r \leq j < s \leq n$ holds. After multiplying both sides of (**??**) by $d^\star$ we obtain

$$\begin{aligned} & \left|d^\star - (p(j) - p(i))\right| + \left|d^\star - (p(s) - p(r))\right| \\ \leq \quad & \left|d^\star - (p(s) - p(i))\right| + \left|d^\star - (p(j) - p(r))\right|. \end{aligned} \tag{13}$$

To check that the inequality holds we have to consider cases depending on the signs of the individual terms.

Case I: If $p(j) - p(r) \geq d^\star$, it follows that also $p(j) - p(i) \geq d^\star$, $p(s) - p(r) \geq d^\star$ and $p(s) - p(i) \geq d^\star$. Then the inequality (13) is equivalent to $-d^\star + p(j) - p(i) - d^\star + p(s) - p(r) \leq -d^\star + p(s) - p(i) - d^\star + p(j) - p(r)$ where all terms cancel.

Case II: In the following three sub-cases we assume that $p(j) - p(r) < d^\star$ always. If $p(j) - p(i) \geq d^\star$ or $p(s) - p(r) \geq d^\star$ then $p(s) - p(i) \geq d^\star$.

Case IIa: If $p(j) - p(i) \geq d^\star$ and $p(s) - p(r) \geq d^\star$ then the inequality (13) is equivalent to $-d^\star + p(j) - p(i) - d^\star + p(s) - p(r) \leq -d^\star + p(s) - p(i) + d^\star - (p(j) - p(r))$ which simplifies to $2p(j) - 2p(r) \leq 2d^\star$.

Case IIb: If $p(j) - p(i) \geq d^\star$ and $p(s) - p(r) < d^\star$ we obtain from (13) $-d^\star + p(j) - p(i) + d^\star - (p(s) - p(r)) \leq -d^\star + p(s) - p(i) + d^\star - (p(j) - p(r))$ which simplifies to $2p(j) \leq 2p(s)$, which is true as the positions are strictly monotone and $i < r$ by assumption.

Case IIc: Now $p(j) - p(i) < d^\star$ and $p(s) - p(r) \geq d^\star$, so (13) reduces to $+d^\star - (p(j) - p(i)) - d^\star + p(s) - p(r) \leq -d^\star + p(s) - p(i) + d^\star - (p(j) - p(r))$ which simplifies to $2p(i) \leq 2p(r)$, which is analogous to IIb.

Case III: Now $p(j) - p(r) < d^\star$, $p(j) - p(i) < d^\star$ and $p(s) - p(r) < d^\star$. If also $p(s) - p(i) < d^\star$ then we obtain from (13) $d^\star - (p(j) - p(i)) + d^\star - (p(s) - p(r)) \leq d^\star - (p(s) - p(i)) + d^\star - (p(j) - p(r))$ where all terms cancel. Otherwise, for $p(s) - p(i) \geq d^\star$, we arrive at $d^\star - (p(j) - p(i)) + d^\star - (p(s) - p(r)) \leq -d^\star + p(s) - p(i) + d^\star - (p(j) - p(r))$ which simplifies to $2p(i) - 2p(s) \leq -2d^\star$, which is true as the positions are strictly monotone and $i < s$ by assumption.

The cases $i = 0$ and $s = n+1$ follow directly from the definition of the $d(0,i)$ respectively $d(i,\infty)$ for $1 \leq i \leq n$.

■

The question about the distance-weighted and the distance-inhomogeneous case remains. Unfortunately, as simple counter examples show, the Monge condition is not satisfied in these settings. Consider the weighted, distance-homogeneous case. Then the terms in (12) have to be multiplied by the respective weights $w_d(p(i))$ and $w_d(p(r))$, which leads to the following inequality for Case I in the proof of the previous lemma

$$\begin{aligned} & w_d(p(i)) \cdot \left(-d^\star + p(j) - p(i)\right) \\ + \quad & w_d(p(r)) \cdot \left(-d^\star + p(s) - p(r)\right) \\ \leq \quad & w_d(p(i)) \cdot \left(-d^\star + p(s) - p(i)\right) \\ + \quad & w_d(p(r)) \cdot \left(-d^\star + p(j) - p(r)\right), \end{aligned}$$

which simplifies to $(w_d(p(i)) - w_d(p(r))) \cdot (p(j) - p(s)) \leq 0$. As $p(j) - p(s) < 0$ by assumption, one would need that also $w_d(p(i)) - w_d(p(r) < 0$, or that the weights are monotonously decreasing. Similarly, for inhomogeneous $d^\star$, for simplicity assuming $w_d(x) \equiv 1$, we obtain in the case that the sums we take the absolute values off are all negative,

$$\begin{aligned} & d^\star(p(r)) \cdot \left(-d^\star(p(i)) + p(j) - p(i)\right) \\ + \quad & d^\star(p(i)) \cdot \left(-d^\star(p(r)) + p(s) - p(r)\right) \\ \leq \quad & d^\star(p(r)) \cdot \left(-d^\star(p(i)) + p(s) - p(i)\right) \\ + \quad & d^\star(p(i)) \cdot \left(-d^\star(p(r)) + p(j) - p(r)\right), \end{aligned}$$

which simplifies to $(d^\star(p(i)) - d^\star(p(r))) \cdot (p(s) - p(j)) \leq 0$; this only holds for monotonously increasing $d^\star(\cdot)$.

The algorithmic improvement coming from the Monge condition translates to the shortest path problem as follows. Instead of having to consider all neighbors we can ignore edges to traverse and nodes to visit because we know a priori that they cannot be part of a shortest path.

This is relevant, as linear time, $O(n)$, on-line algorithms [27]–[29] for the shortest path problem for upper triangular Monge matrices exist, which use the SMAWK-algorithm of Aggarwal *et al.* [30]. Moreover, also efficient algorithms to compute shortest paths with a prescribed number of edges have been proposed; e.g. [31].

B. General shortest path algorithms

We can obtain solutions for the MCTPP in the general case by using standard shortest path algorithms. However, even *linear* graph algorithms, that is algorithms running in $O(|V| + |E|)$ where V and E are the set of vertices and edges respectively, are *quadratic* in the number of candidate oligonucleotides, as the size of the MCTPP instance graphs is $O(n^2)$.

Clearly one can specify position-specific weights in the MCTPP which make it necessary to explore all edges in order to find a shortest path. As a consequence, $O(n^2)$ is the best we can do without making any assumptions. However, weights are chosen by users to reflect their higher interests in particular regions of the genome or prior information and we can safely assume their choices to be *reasonable*. Thus, even in the general case the distance penalties will dominate the remaining penalties. Note that typically $\max_l pt(l) < 0.3$; the quality penalties are equally bounded. Therefore, we can again limit the space of possible $0, \infty$-paths apriori to speed up computations. We will argue that there exists a positive integer k, such that the shortest $0, \infty$-path in an MCTPP instance graph is actually contained in a proper subgraph of G, namely $G_k = (V, E_k)$. Here E_k is the restriction of E to edges connecting vertices i, j with $j - i \leq k$: $E_k = E \setminus \{(i,j) | j - i > k\}$. We will show that k is independent of $|V|$, but dependent on $d^\star$, exploiting the fact that $p(j) - p(i) \geq j - i$, and consequently that the order of G_k is $n+2$ but the size is only $k \cdot n$.

As a consequence, larger arrays containing more oligonucleotide probes, with lower $d^\star$ for the same genome length, are actually faster to compute.

We will only consider the unweighted, distance-inhomogeneous case. We expect similar, likely weaker, results to hold for the weighted case, but proofs will get increasingly tedious. Possibly a modification of the distance penalties could simplify matters.

Lemma 2: Let T be a solution to the unweighted, distance-inhomogeneous MCTPP. Any two consecutive probes i, j in T have the property that $j - i < 2 \cdot max_x d^\star(x) + 1$, provided $\max_l pt(l) + \max_l pq(l) < \frac{min_x d^\star(x)}{max_x d^\star(x)}$.

Proof: We assume that T is a minimal-cost solution but that $j - i \geq 2 \cdot max_x d^\star(x) + 1$. We will show that we can lower the cost by replacing the edge (i, j) by an i, j-path using intermediate sequences. From the lower bound for $j - i$ it follows that we can choose i' such that $p(i') - p(i) > d^\star(p(i))$ and $p(j) - p(i') > d^\star(p(i'))$. Using (9), we obtain from $w(i, j) \geq w(i, i') + w(i', j)$, recall that we are in the unweighted case, $0 \geq \left(\frac{d^\star(p(i))}{d^\star(p(i'))} - 1\right) \cdot \big(p(j) - p(i')\big) - d^\star(p(i)) + d^\star(p(i)) \big(pt(i') + pq(i')\big)$. As $\max_l pt(l) + \max_l pq(l) < 1$ this clearly holds if $d^\star(p(i')) \geq d^\star(p(i))$, as $p(j) > p(i')$ by definition. If for all $i < i' < j$ such that $p(j) - p(i')$ positive $d^\star(p(i')) < d^\star(p(i))$ holds, we have to use a more complicated construction with several intermediate vertices i_m, which is straightforward but cumbersome and which we will omit here. ■

In practice we resort to choosing $k > 2 \cdot max_x d^\star(x)$, based on experimental results and manual inspection of gap size, melting temperature and probe quality distributions. A shortest path in the pruned MCTPP instance graphs G_k can consequently be computed for example by Dijkstra's shortest path algorithm [32] using a Fibonacci heap based priority queue [33] or, exploiting the fact that the G_ks are DAGs and we are given the vertices in topological order, by a simplified algorithm [34]. In experiments we found through profiling that the computation of edge weights takes up well over 99% CPU time; consequently the choice of algorithmic variant is not overly important. The computational complexity is governed by the $k \cdot n$ edge weights. Note that typical values for k are on the order of several hundreds. Nevertheless, Dijkstra's shortest path algorithm using a Fibonacci heap based priority queue computes an optimal solution of MCTPP in log-linear time, $O(n \times \log(n))$, as $|V| \log |V| > k|V| > |E|$ for $|V| \to \infty$; similarly the linear time shortest path algorithm in DAGs, which does not require a priority queue, yields a $O(n)$ running time on MCTPP instances.

C. Implementation issues

Problem instances range from order 2,000,000 and size 80,000,000 up to order 30,000,000 and size 1,200,000,000 for a bacterium like *M. tuberculosis* and a human chromosome respectively. Even state-of-the-art libraries like Boost (`http://www.boost.org`) or LEDA [35] cannot effectively cope with graphs of this size. For example for $n = 3.200.000$ and $k = 300$ a Boost-based implementation needed over 50GB of memory and over 20 minutes of CPU time for *allocating* G_k. This does not include time for computing edge weights or the shortest path. This makes use of online algorithms mandatory. We can adapt algorithms to compute the neighborhood of vertices and weights of incident edges on the fly, instead of precomputing all neighbors and storing them as a graph. This quite obvious optimization of shortest path computations was implemented for example in [36].

Our method is implemented in Python (`http://www.python.org` using the numpy (`http://numpy.scipy.org/`) package for linear algebra and a priority queue implementation from `http://py.vaults.ca/apyllo.py/514463245.769244789.44776582`. We used David Eppstein's PADS library `http://www.ics.uci.edu/~eppstein/PADS` for the online linear time algorithm.

IV. Candidate Generation

For small genomes of viruses or *Mycoplasma*, it would be feasible to enumerate every $n$mer in the genome and calculate the parameters such as similarity to other regions in the genome, melting temperature, quality scores and compute the tiling path. For larger genomes, this computation becomes too demanding and specialized tools need to be employed as complex features, repetitive regions in particular, need to be considered prior to candidate generation.

We do not use all possible oligonucleotides for the candidate set but restrict ourselves to those that adhere to certain parameters. Depending on the settings, we used 5% to 50% of the possible probes in a given genome as the candidate set.

A. Selecting suitable oligonucleotides

An ideal candidate set of genome sequence for a tiling array would consist of $n$mers that are unique within the genome, have the same melting temperature and do not contain particular sequence properties, such as long runs of the same nucleotide, all of which would compromise the hybridization results. A naive approach would be to compute all $n$mers in a genome and eliminate all those that do not fulfill particular criteria, thus separating the bad from the good oligonucleotides.

There is however no theoretical approach that would give us a handle on a oligonucleotide's potential for cross-hybridization. Moreover, it is evident that sequence similarity is not a sufficient predictor for hybridization due to the complex dynamics of DNA hybridization. Nevertheless it is predominantly used due to simplicity and efficiency. Experimentally validated approaches usually cover only small sets and are aimed at providing rather simple rules of thumb. Kane *et al.* provided an estimate that a stretch of 15 contiguous bases in a 50mer has a high potential for cross-hybridization [11]. This is often referred to as Kane's 1st criterion, which has become an accepted albeit not loved description of hybridization properties for oligonucleotides with lengths in the range of 40 to 70. From these analyses it remains unclear to what extent substrings of different lengths influence this potential. Also, it would be unwise to eliminate an oligonucleotide with substring of length ≤ 15, if it would be the best probe in a region, as the signal is still useful. The minimal substring is also dependent on the size of the genome. Maintaining the same parameter for eukaryotic genomes as established for a bacterium would result in a sparse set of candidate oligonucleotides. It is obvious though that avoiding probes that are highly similar to others is beneficial for the probe set. We have incorporated deviation from a particular quality value as costs in the MCTPP. We use the minimal length of shared substrings of the probes as the quality criterion $q(i)$.

Kane's 2nd criterion regards the global similarity of a probe to another one and is described as a threshold of 75% identity for cross-hybridization. In our hands, only few probes in whole genomes pass the global quality criterion and do not share substrings on the order of length 15. We therefore do not incorporate this criterion in our quality assessment but remove oligonucleotides that do not pass this threshold from the candidate set.

Unfortunately, real genomes typically contain many duplicated and repetitive regions of various similarity to each other. Simply masking areas that are similar to other regions results in poor coverage for the overall design.

B. Treatment of repetitive elements

For bacterial genomes, the problem is limited to few larger regions of duplications and a number of transposons that appear in the genome in larger copy numbers. For eukaryotic genomes, the issue is of importance for any analysis. Consequently several tools have been developed to address the issue, notably RepeatMasker [17]. However, they typically are built to identify areas that are repetitive in nature but not necessarily exact duplicates of other regions of the genome. Many repetitive elements have biological functions and are of great interest irrespectively of the complications they bring. Thus, regarding them as junk-DNA and removing them from the set is not a very elegant treatment that affects at least 50% of the genome.

The procedure we use for oligonucleotide selection retrieves only unique elements in a genome. To ensure that duplicated regions become part of the tiling path, we need a different approach than masking all repetitive elements. We devised our own scheme and identify all repeated element that are highly similar to other regions of the genome irrespective of the type of repetitive elements. We want to preserve features in the genome that are repetitive in their origin but unique due to acquired mutation as we could find suitable oligonucleotides in this positions. This way, duplicated regions that contain oligonucleotides of high quality will be present in the probe set, yielding a high coverage of the genome.

C. Implementation

Computation of the candidates is incorporated in the Python program selecting the minimal cost tiling path (see Section III-C). We use previously described, available programs for the generation of the candidate probe set.

Computation of duplicated regions: We used Vmatch [37] with the shortest minimal lengths and maximal edit distance to identify all duplicated regions of high similarity. Query regions were subsequently masked by replacing the sequence with a corresponding number of N.

Candidate enumeration using `Flog`*:* We used `Flog` [38], which yields all oligonucleotide probes of a certain length range for a given sequence that are minimally redundant in the genome by use of a suffix array [39]. It employs filters for GC-content, patterns of low complexity, melting temperature T_m and the quality parameter $q(i)$, which is the minimal shared substring with other probes (Kane's 1st criterion). Quality values were obtained by post-processing the output of `Flog` runs.

Filtering of Oligonucleotides: To limit cross-hybridization following Kane's second, global criterion, SSAHA [40] was employed to compare all candidate oligonucleotides to the genome. $n$mers that were contained in regions of more than 75% over the oligonucleotide length were removed from the set. To accelerate the process, we assembled overlapping oligonucleotides into contiguous regions, or contigs, and compared contigs instead of individual candidates to the genomes. Matches in contigs were mapped back to candidates for the filtering.

To remove palindromic sequences that could form hairpins, we used the `palindrome` implementation of EMBOSS [41].

Hardware requirements: The minimal tiling paths were calculated on a dual-processor AMD Athlon 64 X2 Dual Core Processor 3800+ desktop computer with clock speed of 2.80GHz and 2GB of RAM under Linux. We found that less memory is required for the computation of the tiling path (see below). Computation of the candidate set requires more resources in the case of eukaryotic genomes, in particular the SSAHA runs. We employed 3 Linux machines with 64 Gb of RAM and 4 AMD Opteron 854 processors each.

V. Results

In the following, we demonstrate the application of the methods presented in this work on two examples, the design of an array for *Mycobacterium bovis* BCG with 44,000 spots and a design for the human genome; we limit the display to the longest chromosome 1 for simplicity. Preprocessing and candidate generation were performed for all chromosomes.

A. Mycobacterium bovis BCG

We initially applied our approach to several Mycobacteria and designed arrays with 44,000 spots per chip. Mycobacteria have GC-rich genomes and contain several repetitive regions of varying degree of similarity, which needs to be taken into account in the design. As an example we have selected the 4.4Mbp genome sequence of *Mycobacterium bovis* BCG [20].

Candidate set: The shortest duplication reported by Vmatch was set to 300 using an edit distance of 10. The majority of regions without dense coverage map to the PE/PPE genes, a large, mycobacteria-specific family of proteins that have high GC-content and further known repeats.

As parameters for `Flog`, we used a desired probe length l of 40 to 45, a GC-content between 45% and 70% and we generated a set of 2,192,383 candidate oligonucleotides with these broad parameters with a maximal substring length q = 19. For the minimal cost tiling path, we chose $d^\star$ = 92 and preferred melting temperature $T_m^\star$ = 85°C. Subsequent filtering removed only 2.13% (46,914 oligonucleotides) due to Kane's 2nd criterion or palindromic sequences.

Shortest path: Computation of the standard distance-unweighted and unconstrained tiling path for BCG took 36 min, 28 secs and used a maximum of 630.90MB of memory.

Monge: The same task with identical parameters was performed using using the linear time algorithm, which only used a total of 3 min, 19 seconds. The population of the candidate set data structure alone took 43 seconds. Memory consumption was lower with maximally 537.52 MB, most of which reflects storing the candidates.

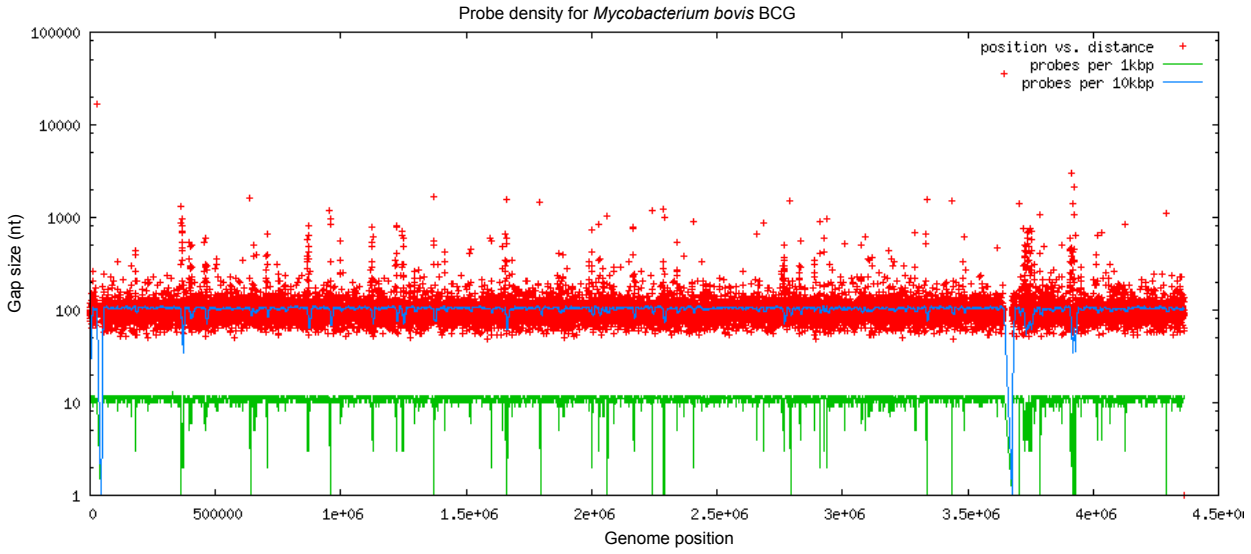


Fig. 3. Coverage of the genome of *Mycobacterium bovis* BCG Parameters were $d^{\star} = 92$ and 43,904 candidates were selected. The two regions for which no coverage is obtained are duplicated regions. Corresponding probes are present in the tiling path at upstream positions.

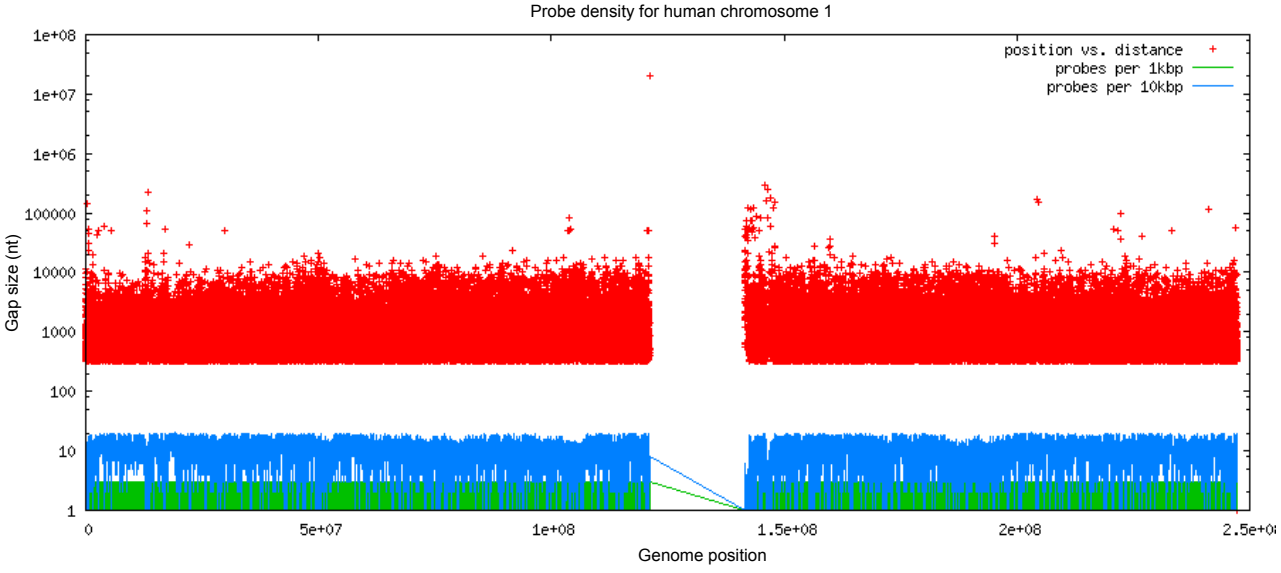


Fig. 4. Coverage of the genome of human chromosome 1. Parameters were $d^{\star} = 500$ and 225,469 oligonucleotides. The large gap in the center of the chromosome is the centromere, which is a highly repetitive structure that is not sequenced. Note the lower coverage compared to bacterial genome in Fig. 3.

B. Human genome

Chromosome 1 of *Homo sapiens* is the largest and contains 247 $Mbps$.

Candidate set: We selected a minimal length of 500 and an edit distance of 10 as parameters for Vmatch. Our set of candidates is built on the masked genome. We built a set for oligonucleotides with a length l of 50 to 60 bases and a GC content of 40% to 60%. The target melting temperature $T_m^{\star}$ was 70.5°C. These setting resulted in 50,659,674 candidates of which 4,179,058 were located on chromosome 1. For a real application, it would be advisable to use shorter oligonucleotide lengths and lower GC-content settings.

Candidate generation is by far the most time consuming element of the process and running times heavily depend on parameter settings. Typically, the most time consuming part is running SSAHA. For the human genome, SSAHA took about 900 CPU hours, which can be trivially distributed. However, genome processing and candidate generation needs to be performed only *once* for each genome; as we have seen, tiling paths can be calculated on this fixed set in minutes.

Shortest Path: The tiling path of human chromosome 1 was generated in 97 min. A total of 225,469 probes were selected for the tiling path.

Monge: The tiling path was computed in 36 minutes and required a maximal memory of 1097.58MB.

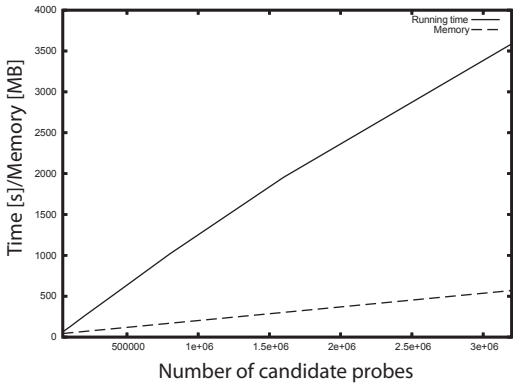


Fig. 5. Memory usage and running time of the algorithm for segments of a human chromosome. In this experiments, n denotes the number of probe candidates, $k = 500$ denotes the neighborhood cardinality. Further parameters were $d^\star = 150$, $T_m^\star = 87.5°\mathrm{C}$

C. Comparison to naive tiling arrays

For the 6.9 Mbp of *Mycobacterium smegmatis* we selected a desired $T_m^\star$ of 87.5°C and a desired distance of $d^\star$ = 150 bp using oligonucleotides of length 50 to 60, which yielded the tiling path comprising 44,000 probes, which was used in the laboratory. A naive equidistant tiling with $d^\star = 150$ selects more than 75% of probes rejected by our candidate generation and can be considered as of minor quality.

VI. Discussion

Tiling arrays are versatile tools for research in molecular biology. Here, we have presented a problem formulation for their design which is efficiently solvable by standard algorithms. Our approach is more flexible than previously published approaches while at least maintaining or improving running time for the computation of the tiling path compared to prior approaches. The one-time cost of processing a genome to compute candidate oligonucleotides can be neglected if the criteria for oligos are established and kept constant, which is the case for most laboratory applications. Once this is fixed, any number of tiling paths with different parameters can be computed quickly, even for complete eukaryotic genomes. This will be implemented as a web service at `http://tileomatic.org` to allow users to interactively generate custom tiling paths for their preferred experimental conditions and introduce further requirements such as required oligonucleotides, or different trade-offs between criteria for the path.

Future improvements in our framework will aim at a better description of the quality parameters. It would be highly useful to develop a more realistic model of cross hybridization potential of a candidate oligonucleotide that merges both global and local constraints and which still can be efficiently computed for a given genome.

One approach would be to use the uniqueness score U (see section I-B) introduced by Gräf *et al*. [22], which could give a better handle on treatments of minimal substrings for candidate generation. This could also improve on the handling of repeat sequences.

Another useful idea to quality treatments is described in the work of Lipson *et al*. [15]. Instead of selecting oligos by deviation from the desired quality, they rank the quality q in a window of fixed length and select by minimal deviation from best rank. Extending the idea to our approach would mean to select oligonucleotide quality in a moving window, which is easily done; it might be suitable to incorporate both Kane's 1st and 2nd criterion in one frame.

The selection of oligonucleotides could be improved further by finding narrower windows for melting temperatures as we currently rely on an oligonucleotide length range and prefer unique oligonucleotides over those that are isothermal in the current set up. This could be realized by post-processing oligonucleotides of overly high melting temperatures by shortening them while maintaining quality.

Our analysis method, which copes with the variable distances and the resulting variable spatial correlations, will be described together with the first experimental results elsewhere, as this is beyond the scope of this manuscript.

Acknowledgment

Thanks to Jörg Schreiber at the MPI for Infection Biology for helpful discussions and experiments with *M. smegmatis* tiling arrays, to Stefan Bienert and Stefan Kurtz for making `Flog` [38] available to us and to Janne Grunau for computational experiments. We would also thank one of the reviewers of the conference version for helpful comments regarding Monge theory.

References

[1] M. J. Buck and J. D. Lieb, "Chip-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments," *Genomics*, vol. 83, no. 3, pp. 349–360, Mar 2004.

[2] M. Barrett, A. Scheffer, A. Ben-Dor, N. Sampas, D. Lipson, R. Kincaid, P. Tsang, B. Curry, K. Baird, P. Meltzer, *et al.*, "Comparative Genomic Hybridization Using Oligonucleotide Microarrays and Total Genomic DNA," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 51, pp. 17 765–17 770, 2004.

[3] J. Mattick, "The Functional Genomics of Noncoding RNA," pp. 1527–1528, 2005.

[4] E. Birney, J. Stamatoyannopoulos, A. Dutta, R. Guigó, T. Gingeras, E. Margulies, Z. Weng, M. Snyder, E. Dermitzakis, J. Stamatoyannopoulos, *et al.*, "Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project," *Nature*, vol. 447, no. 7146, pp. 799–816, 2007.

[5] G. Yuan, Y. Liu, M. Dion, M. Slack, L. Wu, S. Altschuler, and O. Rando, "Genome-Scale Identification of Nucleosome Positions in S. cerevisiae," pp. 626–630, 2005.

[6] P. Sabo, M. Kuehn, R. Thurman, B. Johnson, E. Johnson, H. Cao, M. Yu, E. Rosenzweig, J. Goldy, A. Haydock, *et al.*, "Genome-scale mapping of DNase I sensitivity in vivo using tiling DNA microarrays," *Nature Methods*, vol. 3, pp. 511–518, 2006.

[7] D. Okou, K. Steinberg, C. Middle, D. Cutler, T. Albert, and M. Zwick, "Microarray-based genomic selection for high-throughput resequencing," *Nature Methods*, vol. 4, pp. 907–909, 2007.

[8] T. Albert, M. Molla, D. Muzny, L. Nazareth, D. Wheeler, X. Song, T. Richmond, C. Middle, M. Rodesch, C. Packard, *et al.*, "Direct selection of human genomic loci by microarray hybridization," *Nature Methods*, vol. 4, pp. 903–905, 2007.

[9] T. Mockler and J. Ecker, "Applications of DNA tiling arrays for whole-genome analysis," *Genomics*, vol. 85, no. 1, pp. 1–15, 2005.

[10] K. Breslauer, R. Frank, H. Blocker, and L. Marky, "Predicting DNA Duplex Stability from the Base Sequence," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 83, no. 11, pp. 3746–3750, 1986.

[11] M. D. Kane, T. A. Jatkoe, C. R. Stumpf, J. Lu, J. D. Thomas, and S. J. Madore, "Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays," *Nucleic Acids Res*, vol. 28, no. 22, pp. 4552–4557, Nov 2000.

[12] D. W. Selinger, K. J. Cheung, R. Mei, E. M. Johansson, C. S. Richmond, F. R. Blattner, D. J. Lockhart, and G. M. Church, "Rna expression analysis using a 30 base pair resolution escherichia coli genome array," *Nat Biotechnol*, vol. 18, no. 12, pp. 1262–1268, Dec 2000.

[13] P. Kapranov, S. Cawley, J. Drenkow, S. Bekiranov, R. Strausberg, S. Fodor, and T. Gingeras, "Large-Scale Transcriptional Activity in Chromosomes 21 and 22," *Science*, vol. 296, no. 5569, p. 916, 2002.

[14] P. Kapranov, J. Cheng, S. Dike, D. Nix, R. Duttagupta, A. Willingham, P. Stadler, J. Hertel, J. Hackermuller, I. Hofacker, *et al.*, "RNA Maps Reveal New RNA Classes and a Possible Function for Pervasive Transcription," *Science*, vol. 316, no. 5830, p. 1484, 2007.

[15] D. Lipson, Z. Yakhini, and Y. Aumann, "Optimization of probe coverage for high-resolution oligonucleotide acgh," *Bioinformatics*, vol. 23, no. 2, pp. 77–83, Jan 2007.

[16] E. Prak and H. Kazazian Jr, "Mobile elements and the human genome," *Nat Rev Genet*, vol. 1, no. 2, pp. 134–44, 2000.

[17] A. Smit, R. Hubley, and P. Green, "RepeatMasker Open-3.0," *Institute for Systems Biology. http://www. repeatmasker. org (January 10, 2007)*, 2004.

[18] E. Ryder, R. Jackson, A. Ferguson-Smith, and S. Russell, "MAMMOT-a set of tools for the design, management and visualization of genomic tiling arrays," pp. 883–884, 2006.

[19] E. Lander, L. Linton, B. Birren, C. Nusbaum, M. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, *et al.*, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860–921, 2001.

[20] R. Brosch, S. Gordon, T. Garnier, K. Eiglmeier, W. Frigui, P. Valenti, S. Dos Santos, S. Duthoy, C. Lacroix, C. Garcia-Pelayo, *et al.*, "Genome plasticity of BCG and impact on vaccine efficacy," *Proceedings of the National Academy of Sciences*, vol. 104, no. 13, p. 5596, 2007.

[21] P. Bertone, V. Trifonov, J. S. Rozowsky, F. Schubert, O. Emanuelsson, J. Karro, M. Y. Kao, M. Snyder, and M. Gerstein, "Design optimization methods for genomic dna tiling arrays," *Genome Res*, vol. 16, no. 2, pp. 271–281, Feb 2006.

[22] S. Gräf, F. G. Nielsen, S. Kurtz, M. A. Huynen, E. Birney, H. Stunnenberg, and P. Flicek, "Optimized design and assessment of whole genome tiling arrays," *Bioinformatics*, vol. 23, no. 13, pp. 195–204, Jul 2007.

[23] A. Schliep, D. Torney, and S. Rahmann, "Group testing with DNA chips: generating designs and decoding experiments," in *Proceedings of the 2nd IEEE Computer Society Bioinformatics conference*, 2003, pp. 84–93.

[24] O. Shai, Q. Morris, B. Blencowe, and B. Frey, "Inferring global levels of alternative splicing isoforms using a generative model of microarray data," *Bioinformatics*, vol. 22, no. 5, p. 606, 2006.

[25] Z. Galil and K. Park, "Dynamic programming with convexity, concavity, and sparsity," *Theoretical Computer Science*, vol. 92, no. 1, pp. 49–76, 1992. [Online]. Available: citeseer.ist.psu.edu/galil92dynamic.html

[26] R. E. Burkard, B. Klinz, and R. Rudolf, "Perspectives of Monge properties in optimization." *Discrete Applied Mathematics*, vol. 70, no. 2, pp. 95–161, 1996.

[27] R. Wilber, "The concave least-weight subsequence problem revisited," *J. Algorithms*, vol. 9, no. 3, pp. 418–425, 1988.

[28] D. Eppstein, "Sequence comparison with mixed convex and concave costs," *J. Algorithms*, vol. 11, no. 1, pp. 85–101, 1990.

[29] Z. Galil and K. Park, "A linear-time algorithm for concave one-dimensional dynamic programming," *Inf. Process. Lett.*, vol. 33, no. 6, pp. 309–311, 1990.

[30] A. Aggarwal, M. Klawe, S. Moran, P. Shor, and R. Wilber, "Geometric applications of a matrix searching algorithm," in *SCG '86: Proceedings of the second annual symposium on Computational geometry*. New York, NY, USA: ACM Press, 1986, pp. 285–292.

[31] A. Aggarwal, B. Schieber, and T. Tokuyama, "Finding a minimum weight k-link path in graphs with Monge property and applications," in *SCG '93: Proceedings of the ninth annual symposium on Computational geometry*. New York, NY, USA: ACM Press, 1993, pp. 189–197.

[32] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Numerische Mathematik*. Mathematisch Centrum, Amsterdam, The Netherlands, 1959, vol. 1, pp. 269–271.

[33] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, pp. 596–615, 1987.

[34] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, September 2001.

[35] "Leda," http://www.algorithmic-solutions.com/.

[36] A. Schliep, "The software GADAR and its application to extremal graph theory," in *Proceedings of the Twenty-fifth Southeastern International Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1994)*, vol. 104, 1994, pp. 193–203.

[37] S. Kurtz, "The Vmatch large scale sequence analysis software," *Ref Type: Computer Program*, pp. 4–12, 2003.

[38] S. Bienert, "Flexible combination of filters for oligodesign," Diplomathesis, Center for Bioinformatics, Universität Hamburg, 2006.

[39] M. Abouelhoda, S. Kurtz, and E. Ohlebusch, "Replacing suffix trees with enhanced suffix arrays," *Journal of Discrete Algorithms*, vol. 2, no. 1, pp. 53–86, 2004.

[40] Z. Ning, A. J. Cox, and J. C. Mullikin, "SSAHA: a fast search method for large dna databases," *Genome Res*, vol. 11, no. 10, pp. 1725–1729, Oct 2001.

[41] P. Rice, I. Longden, and A. Bleasby, "EMBOSS: The European Molecular Biology Open Software Suite," *Trends in Genetics*, vol. 16, no. 6, pp. 276–277, 2000.

[42] C. Garnis, W. Lockwood, E. Vucic, Y. Ge, L. Girard, J. Minna, A. Gazdar, S. Lam, C. MacAulay, and W. Lam, "High resolution analysis of non-small cell lung cancer cell lines by whole genome tiling path array CGH," *Int. J. Cancer*, vol. 118, pp. 1556–1564, 2006.

[43] A. Schliep and R. Krause, "Efficient Computational Design of Tiling Arrays Using a Shortest Path Approach," *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4645, p. 383, 2007.

[44] "Boost c++ libraries," http://www.boost.org/.

[45] "Automatically tuned linear algebra software (atlas)," http://math-atlas.sourceforge.net/.

[46] "Python," http://www.python.org/.

[47] "Numpy," http://numpy.scipy.org/.

[48] A. Pozhitkov, P. A. Noble, T. Domazet-Loso, A. W. Nolte, R. Sonnenberg, P. Staehler, M. Beier, and D. Tautz, "Tests of rRNA hybridization to microarrays suggest that hybridization characteristics of oligonucleotide probes for species discrimination cannot be predicted," *Nucleic Acids Res*, vol. 34, no. 9, 2006.

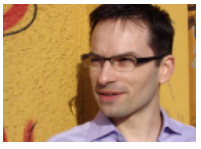
[49] Z. He, L. Wu, X. Li, M. Fields, and J. Zhou, "Empirical Establishment of Oligonucleotide Probe Design Criteria," *Applied and Environmental Microbiology*, vol. 71, no. 7, pp. 3753–3760, 2004.

[50] O. Matveeva, S. Shabalina, V. Nemtsov, A. Tsodikov, R. Gesteland, J. Atkins, and O. Journals, "Thermodynamic calculations and statistical correlations for oligo-probes design," *Nucleic Acids Research*, vol. 31, no. 14, pp. 4211–4217, 2003.

[51] S. Cole, R. Brosch, J. Parkhill, T. Garnier, C. Churcher, D. Harris, S. Gordon, K. Eiglmeier, S. Gas, C. Barry III, *et al.*, "Deciphering the biology of Mycobacterium tuberculosis from the complete genome sequence," *Nature*, vol. 393, pp. 537–544, 1998.

[52] J. SantaLucia Jr, H. Allawi, and P. Seneviratne, "Improved nearest-neighbor parameters for predicting DNA duplex stability," *Biochemistry*, vol. 35, no. 11, pp. 3555–3562, 1996.

Alexander Schliep received his Ph.D. in computer science from the Center for Applied Computer Science (ZAIK) at the University of Cologne, Germany, in 2001, working in collaboration with the Theoretical Biology and Biophysics group (T-10) at Los Alamos National Laboratory, NM. Since 2002, he is group leader of the bioinformatics algorithms group at the Department for Computational Molecular Biology at the Max Planck Institute for Molecular Genetics in Berlin. The research interests pursued in his group include data mining, statistical models and algorithms for analyzing complex, heterogeneous data from molecular biology.



Roland Krause received his doctorate in biochemistry from the University of Heidelberg in 2004 for works on the bioinformatic analysis of protein complexes, which was conducted at Cellzome AG in Heidelberg, Germany. His research interests are bacterial genomics, in particular regarding regulation of transcription and protein-protein interactions.