Lecture 2: March 21, 2002

Lecturer: Ron Shamir *Scribe: Orly Stettiner and Gadi Kimmel*

2.1 Introduction

2.1.1 Sequencing by Hybridization

DNA Arrays or DNA Chips were proposed in the late 1980's by several researchers independently for the purpose of DNA sequencing (for example [3], [10] and [11]), and the technology was named *DNA Sequencing By Hybridization* (*SBH*). This method may also be referred to as "sequencing by k-tuple composition". The idea is to build a two-dimensional grid (or matrix) of all possible k-tuples (or "k-mers") for a given k. At each (i, j) entry a distinct k-tuple or probe is attached. The matrix of probes will be referred to as the "k-chip", C(k), or the "sequencing chip". The DNA probes are referred to as *oligonucleotides*, or *oligos* for short. Then, a sample of the single stranded DNA to be sequenced is presented to the matrix. This DNA is labelled with a radioactive or fluorescent material. Each k-tuple present in the sample hybridizes with its reverse complement in the matrix. After washing unhybridized DNA from the chip, the hybridized k-tuples can be determined by a device, which detects the labelled DNA. One can distinguish between two major formats of DNA arrays:

- *Format I arrays*, in which the targets are attached to the chip and the probes (oligos) are "in the air". The major technique used for this format is Oligo-Fingerprinting.

- *Format II arrays*, where probes are on the chip and targets (to be sequenced) are "in the air". These chips are either Oligonucleotide Arrays or cDNA Microarrays, where each spot contains a cDNA clone from a known gene, instead of arbitrary short oligos. A typical experiment with an oligonucleotide array is described in Figure 1.17, Lecture 1, [14].

2.1.2 SBH Technology

DNA Arrays can be manufactured with the use of *VLSIPS* (very large scale immobilized polymer synthesis), where probes are grown one nucleotide at a time through a photolithographic process [1]. Every nucleotide carries a photolabile protection group protecting the

probe from further growing. This group can be removed by illuminating the probe with light. In each chemical step, a pre-defined region of the array is illuminated (by using masks), thus removing a photolabile protecting group from that region and activating it for further nucleotide growth. The entire array is then exposed to a particular nucleotide (carrying its own protecting group), and reactions only occur in the activated regions. The light-directed synthesis allows random access to all positions of the array and can be used to make arrays with any probes at any site.

One of the companies specializing in designing and manufacturing "Format II" arrays is Affymetrix. The following section describes the technologies used for manufacturing the GeneChip©array family. Affymetrix's GeneChip© technology provides efficient access to genetic information using miniaturized, high-density arrays of oligonucleotide probes. GeneChip©probe arrays are created and utilized through the following processes [15] (see Figure 1.16, Lecture 1):

Probe Array Design: First, the set of oligonucleotide probes to be synthesized is defined, based on its ability to hybridize to the target loci or genes of interest. With this information, computer algorithms are used to design photolithographic masks for use in manufacturing the probe arrays.

Manufacturing: Probe arrays are manufactured by Affymetrix's proprietary, light-directed chemical synthesis process, which combines solid-phase chemical synthesis with photolithographic fabrication techniques employed in the semiconductor industry. Using a series of photolithographic masks to define chip exposure sites, followed by specific chemical synthesis steps, the process constructs high-density arrays of oligonucleotides, with each probe in a predefined position in the array. Multiple probe arrays are synthesized simultaneously on a large glass wafer. This parallel process enhances reproducibility and helps achieve economies of scale (decreases price per unit). The wafers are then diced, and individual probe arrays are packaged in injection-molded plastic cartridges, that protect them from the environment and serve as chambers for hybridization.

Hybridization and Detection: Once fabricated, the GeneChip©probe arrays are ready for hybridization. The nucleic acid sequence to be analyzed (the target) is isolated, amplified and labelled with a fluorescent reporter group. The labelled target is then incubated with the array and stained with florescent dye using the fluidics station and hybridization oven. After the hybridization reaction is complete, the array is inserted into the scanner, where patterns of hybridization are detected. The hybridization data are collected as light emitted from the fluorescent reporter groups already incorporated into the target, which is now bound to the probe array. Probes that most clearly match the target generally produce stronger signals than those that have mismatches. Since

the sequence and position of each probe on the array are known, by complementarity, the identity of the target sequence applied to the probe array can be determined.

2.2 Sequence Reconstruction from Spectrum

SBH provides information about k-mers present in the DNA string, but does not provide information about the positions of these k-mers.
Definition: S is said to be the *spectrum* of sequence T if S is a multiset of all k-long substrings of T (we assume that the number of occurrences of each k-mer is also known).
Example: T = ATGCAGGTCC, S = {ATG, AGG, CAG, GCA, GGT, GTC, TCC, TGC}.

Problem 2.1 Sequence Reconstruction from Spectrum:
INPUT: A multi-set of k-mers $S = \{s_1, ..., s_{n-k+1}\}$.
QUESTION: Is S the spectrum of the sequence T? If yes, reconstruct the sequence T from its spectrum.

2.2.1 A Simple Solution: TSP / Hamiltonian Path

The problem can be regarded as a Traveling Salesman Problem (TSP) or the Hamiltonian path problem. We define the following directed graph:

- Every occurrence of a k-mer in the spectrum is represented by a vertex in the graph (a k-mer that appears more than once is represented by multiple vertices).

- Every pair of vertices $u, v \in V$ are connected by a directed edge e from u to v iff the $k-1$ suffix of u is identical to the $k-1$ prefix of v. For example: {GCA, CAT} are 3-mers that are connected by a directed edge, since the 2-mer suffix of GCA equals the 2-mer prefix of CAT.

Problem 2.2 Hamiltonian Path:
INPUT: A directed graph H as defined above.
QUESTION: Find a path that visits each vertex exactly once (a Hamiltonian path) in H.
The problem corresponds exactly to finding a DNA sequence T with the spectrum S.

Example: In Figure 2.1 we can see a spectrum which is represented as a graph. In this example, the solution is simple and unique, but in general, the Hamiltonian path is known to be NP-complete. As a result, this approach is not practical for DNA sequence reconstruction.

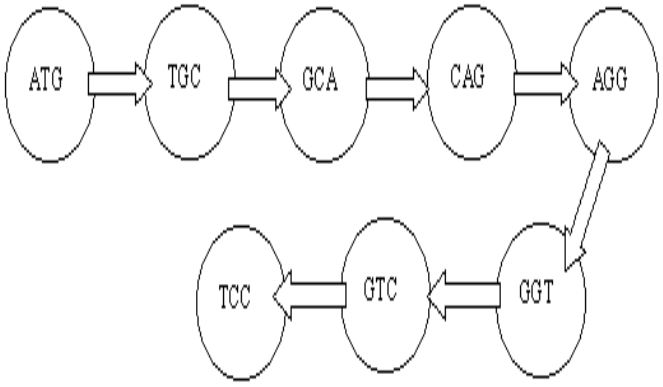

Figure 2.1: A spectrum for T = ATGCAGGTCC which is represented as a graph: each k-mer is represented by a vertex.

2.2.2 SBH and the Eulerian Path Problem (EPP)

Pevzner [2] proposed a different approach, which reduces the SBH problem to the Eulerian path problem, leading to a simple linear-time algorithm for sequence reconstruction. The idea is to construct a graph, whose edges (rather than vertices) correspond to k-mers and to find a path in the graph that visits every edge exactly once. In this approach, the vertices are the full set of (k-1)-mers appearing in the spectrum (each appears as a single vertex). A (k-1)-mer v is joined by a directed edge with a (k-1)-mer w if the spectrum S contains a k-mer, for which the first (k-1) nucleotides are identical to v and the last (k-1) nucleotides are identical to w. A k-mer appearing more than once will be translated into parallel edges.

Problem 2.3 Eulerian Path:
INPUT: A directed graph as defined above, G.
QUESTION: Find a path visiting all edges of G (Eulerian path).

Example: In Figure 2.2 we can see the graph G representing the sequence T=ACAAACGCACTTAA with the spectrum S= {AAA, AAC, ACA, CAC, CAA, ACG, CGC, GCA, ACT, CTT, TTA, TAA}. Each 2-mer is represented by a vertex.

This problem is known to be a simple one, with linear-time solution [12]. However, several difficulties may occur that make the problem non-trivial:

- The solution is not necessarily unique (for example, we may detect an Eularian cycle, instead of a path, which corresponds to multiple ambiguous solutions).

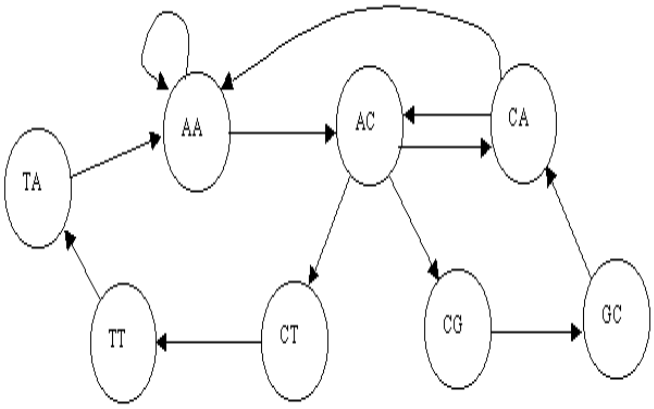

Figure 2.2: Pevzner's graph for T=ACAAACGCACTTAA.

- The input data (the spectrum S) may contain errors (false positives, false negatives, non-certain number of occurrences of each k-mer).

- Multiple parallel edges may lead to ambiguous solutions.

During the 1990's it was shown that for "clean", relatively small sequences (∼100-150), SBH is solvable using DNA chips with k=5, 6.

2.3 Ambiguous vs. Unique Solutions to SBH

2.3.1 Branching

Ambiguity of a SBH solution occurs if it is impossible to reconstruct the original sequence T from Pevzner's graph. This might happen when a branching exists in the graph, as shown in Figure 2.3 (more details can be found in section 2.4). Note that a branching does not necessarily lead to ambiguity. There may be branching that lead to a unique solution, as shown in Figure 2.4.

Consider the following problem: given all k-mers chip C(k), with a given branching probability p, find the expected length of an unambiguously reconstructed sequence. Pevzner et al. [3] have shown that when assuming a random, equally distributed, mutually independent sequence of nucleotides (where $p(A) = p(T) = p(G) = p(C) = \frac{1}{4}$), the expected length is approximately $\frac{1}{3}4^k p$. This means that for $k = 8, p = 0.01$ we can reconstruct approximately 210-long sequences. Theoretical designs of different chips enable us to get somewhat better

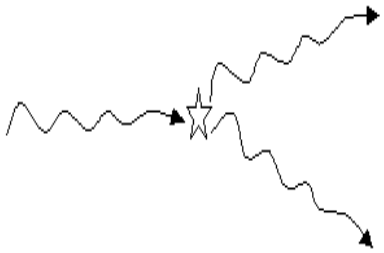

Figure 2.3: Ambiguity of a SBH solution.

Figure 2.4: A branching that lead to a unique solution.

lengths, but SBH still does not appear to present a true alternative to standard sequencing methods.

2.3.2 Probability of Unique Reconstruction

We now study the problem of calculating the probability of unique reconstruction: Given a n-long target for reconstruction from a k-mer chip, what is the probability of a unique reconstruction? For simplicity, we again assume that the nucleotides are independent and identically distributed (i.i.d.) with probability $p = \frac{1}{4}$ for each of A, T, G and C.

A crude heuristic is based on the observation that k-mers repeats often lead to non-unique reconstruction. We can therefore claim that a *sufficient condition* for unique reconstruction is: having no (k-1)-long repeats. There are approximately $\binom{n}{2}$ potential repeats of length $k-1$ corresponding to pairs of positions in the DNA sequence of length n. Each of them occurs with probability $(\frac{1}{4})^{k-1}$. Since $\binom{n}{2} \simeq \frac{n^2}{2}$, the expected number of repeats is approximately $\frac{n^2}{2} \cdot (\frac{1}{4})^{k-1}$. Solving $\frac{n^2}{2} \cdot (\frac{1}{4})^{k-1} \simeq 1$ yields $4^k \simeq 2n^2$. Hence, a rough estimation of k and n is the following:

$$k \simeq \log_4(2n^2) \tag{2.1}$$

$$n \simeq \sqrt{\frac{1}{2}4^k} \tag{2.2}$$

We note that the number of sites on the DNA chip, which contains all possible combinations of k-long probes of the four possible nucleotides is 4^k.
Hence, the length of a uniquely reconstructible target is approximately $\sqrt{\frac{1}{2} \times \text{chip size}}$.

2.4 Alternating Cycles in Colored Graphs

Consider an undirected 2-colored graph $G(V, E)$.
Definition: An *alternating path* in G is a path in which no two consecutive edges have the same color.

2.4.1 Order Transformations of Alternating Paths

Let $F = ...x...y...x...y...$ be an alternating path in a 2-colored graph G, where $x, y \in V$. Vertices x and y partition F into five sub-paths $F = F_1F_2F_3F_4F_5$.
Definition: The transformation $F = F_1F_2F_3F_4F_5 \Rightarrow F^* = F_1F_4F_3F_2F_5$ is called *order exchange* if F^* is an alternating path (see Figure 2.5).

Let $F = ...x...x...$ be an alternating path in a 2-colored graph G, where $x \in V$. Vertex x partitions F into three sub-paths $F = F_1F_2F_3$.
Definition: The transformation $F = F_1F_2F_3 \Rightarrow F^* = F_1\overleftarrow{F_2}F_3$ is called *order reflection* if F^* is an alternating path (see Figure 2.6).
Note that the order reflection $F \Rightarrow F^*$ in a 2-colored graph exists if and only if F_2 is an odd cycle. Otherwise, if F_2 is an even cycle, than the first and last edges in F_2 must be from different colors, and thus $F_1\overleftarrow{F_2}F_3$ would not be an alternating path in G.

2.4.2 Alternating Eulerian Cycles in 2-Colored Graphs

Theorem 2.1 *[4] Every two alternating Eulerian cycles in a 2-colored graph G can be transformed into each other by a series of order transformations (exchanges and reflections).*

Proof: Let X and Y be two alternating Eulerian cycles in G. Consider the set of alternating Eulerian cycles C obtained from X by all possible series of order transformations. Let $X^* = x_1...x_m$ be a cycle in C having the longest common prefix with $Y = y_1...y_m$, i.e. $x_1...x_l = y_1...y_l$ for $l \leq m$. If $l = m$, the theorem holds. Otherwise, let $v = x_l = y_l$, i.e. $e_1 = (v, x_{l+1})$ and $e_2 = (v, y_{l+1})$ are the first different edges in X^* and Y, respectively (see Figure 2.7). Since X^* and Y are alternating paths, the edges e_1 and e_2 must have the same color. Since X^* is Eulerian path, X^* contains the edge e_2. Clearly, e_2 succeeds e_1 in X^*.

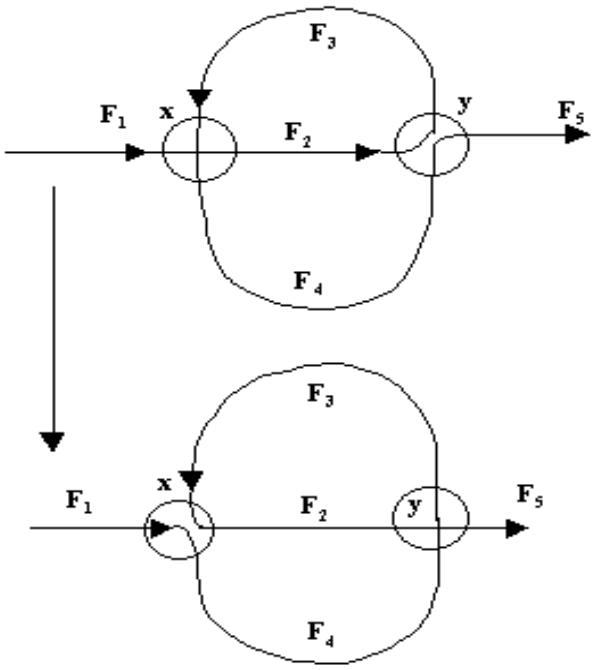

Figure 2.5: Source: [1]. Order exchange.

There are two cases (see Figure 2.7), depending on the direction of the edge e_2 in the path X^* (towards or from vertex v):

Case 1: Edge $e_2 = (y_{l+1}, v)$ in the path X^* is directed towards v. In this case, $X^* = \{x_1...vx_{l+1}...y_{l+1}v...x_m\}$. Since the colors of the edges e_1 and e_2 coincide, the transformation $X^* = F1F2F3 \Rightarrow F1\overleftarrow{F2}F3 = X^{**}$ is an order reflection (Figure 2.8). Therefore, $X^{**} \in C$ and at least $l+1$ initial vertices in X^{**} and Y coincide, contradicting the choice of X^* as the longest prefix of Y.

Case 2: Edge $e_2 = (v, y_{l+1})$ in the path X^* is directed from v. In this case vertex v partitions the path X^* into three parts: prefix X_1 ending at v, cycle X_2, and suffix X_3 starting at v (see Figure 2.9). It is easy to see that X_2 and X_3 have a vertex $x_j = x_k$ $(l < j < k < m)$ in common (otherwise, Y would not have been an Eularian cycle). Therefore, the cycle X^* can now be rewritten as $X^* = F_1F_2F_3F_4F_5$ (Figure 2.9). Consider the edges (x_k, x_{k+1}) and (x_{j-1}, x_j) that are drawn with thick lines.

- If the colors of these edges are different, then $X^{**} = F_1F_4F_3F_2F_5$ is the alternating

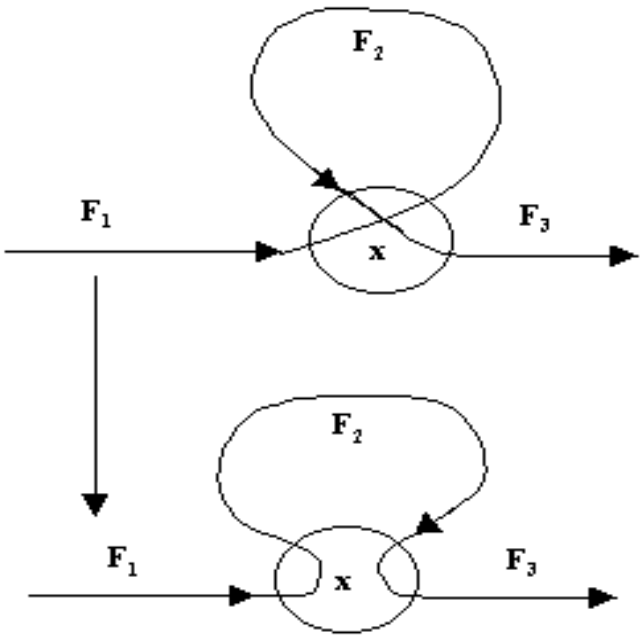

Figure 2.6: Source: [1]. Order reflection.

cycle obtained from X^* by means of the order exchange shown in Figure 2.9, top. At least $l+1$ initial vertices of X^{**} and Y coincide, contradicting the choice of X^*.

- If the colors of the edges (x_k, x_{k+1}) and (x_{j-1}, x_j) coincide (Figure 2.9, bottom), then $X^{**} = F_1F_4\overleftarrow{F_2}\overleftarrow{F_3}F_5$ is obtained from X* by means of two order reflections g and h:
  $g : F_1F_2F_3F_4F_5 \Rightarrow F_1F_2(\overleftarrow{F_3F_4})F_5 = F_1F_2\overleftarrow{F_4}\overleftarrow{F_3}F_5$
  $h : F_1F_2\overleftarrow{F_4}\overleftarrow{F_3}F_5 \Rightarrow F_1(\overleftarrow{F_2\overleftarrow{F_4}})\overleftarrow{F_3}F_5 = F_1(\overleftarrow{\overleftarrow{F_4}})\overleftarrow{F_2}\overleftarrow{F_3}F_5 = F_1F_4\overleftarrow{F_2}\overleftarrow{F_3}F_5$

  At least $l+1$ initial vertices of X^{**} and Y coincide, contradicting the choice of X^*.

This conclude the proof for theorem 2.1.■

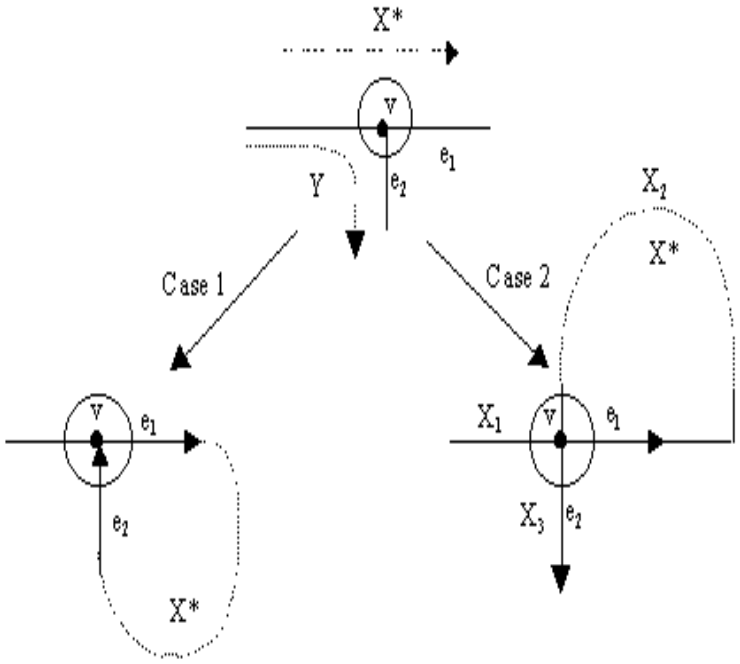

Figure 2.7: Source: [1]. Transformations between alternating Eulerian cycles.

2.5 Alternative Solutions in SBH

2.5.1 String Rearrangements

Quite often we can detect two DNA sequences that produce the same SBH spectrum, due to some "branching". For example, the spectrum: $S = \{$ATG, TGG, GTC, GTG, GGC, GCA, GCG, CGT$\}$ can be read in two ways (two different permutations of its 2-mers): either as $\{$AT, TG, GC, CG, GT, GG, GC, CA$\}$, which gives the sequence: ATGCGTGGCA, or as $\{$AT, TG, GG, GC, CG, GT, GC, CA$\}$, which is equivalent to: ATGGCGTGCA. We can see that we have a "branching" vertex TG: we can not decide which 3-tuple (TGC or TGG) follows ATG in the original sequence (there are two ambiguous solutions). Such ambiguity may be resolved by performing an additional biochemical experiment, for example, by trying to hybridize ATGC with a target DNA fragment (if the first reconstruction is correct, it will hybridize, otherwise not). In order to analyze additional biochemical experiments, one needs to characterize all DNA sequences with the given spectrum.

In 1992, Ukkonen [5] conjectured that any two strings with the same k-mer composition can be transformed into each other by simple operations called *transpositions* and *rotations*,

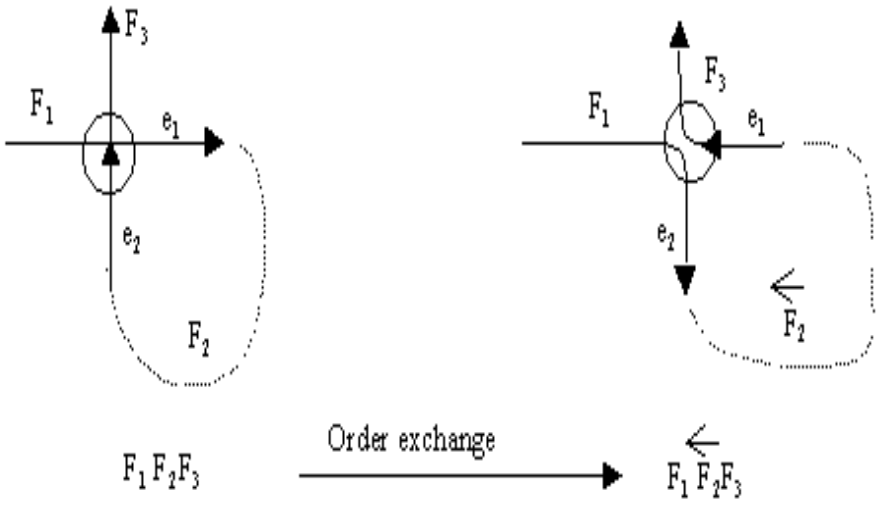

Figure 2.8: Source: [1]. Case 1: Edge $e_2 = (y_{l+1}, v)$ in the path X^* is directed towards v.

defined as follows:

Let T be a string, represented by its (k-1)-mers (meaning that it is written as a sequence of its (k-1)-mers), for example: The sequence {ACTGGGAATCTGATGAATCC} with $k = 4$ will be written as: T = {ACT, **CTG**, TGG, GGG, GGA, **GAA**, AAT, ATC, TCT, **CTG**, TGA, GAT, ATG, TGA, **GAA**, AAT, ATC, TCC}.

Definition: If the string T (written in (k-1)-mers notation) contains interleaving pairs of (k-1)-mers x and y, such that: $T = ...x\underline{...}z...x\underline{\underline{...}}z...$ then the string $T = ...x\underline{\underline{...}}z...x\underline{...}z...$ (where $\underline{...}$ and $\underline{\underline{...}}$ change places) is called a transposition of T. If $T = ...x\underline{...}x\underline{\underline{...}}x...$, where x is a (k-1)-mer, then $...x\underline{\underline{...}}x\underline{...}x...$ is also called a *transposition* of T. In the above example, we have: x = CTG, z = GAA.

Definition: If a string $T = x\underline{...}z\underline{\underline{...}}x$ starts and ends with the same (k-1)-mer x, then the string $z\underline{\underline{...}}x\underline{...}z$ is called a *rotation* of T. For example:
ACTGGGAAT**ACT** $\Rightarrow$ {**ACT**, CTG, TGG, GGG, **GGA**, GAA, AAT, ATA, TAC, **ACT**} $\Rightarrow$ **GGA**ATACTG**GGA**.
Clearly, transpositions and rotations do not change k-mers composition.

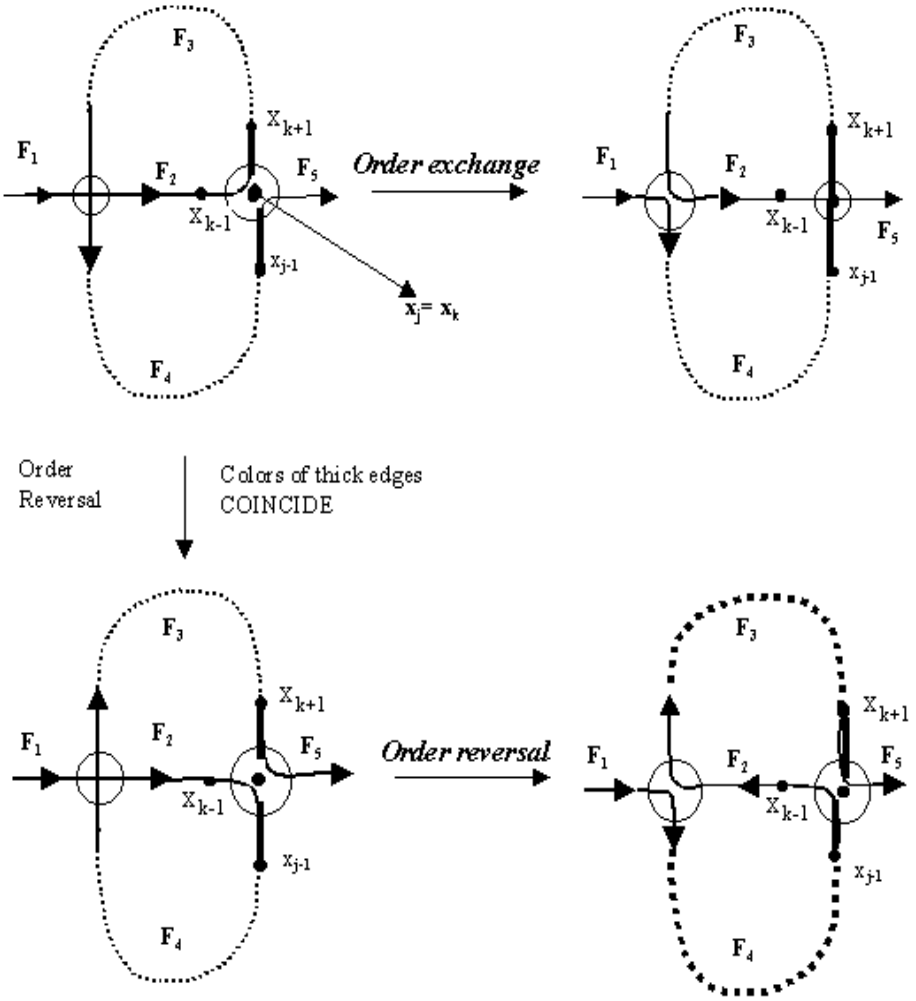

Figure 2.9: Source: [1]. Case 2: depending on the colors of the thick edges, there exists either an order exchange or two order reflections transforming X^* into a cycle with a longer common prefix with Y.

2.5.2 Ukkonen's Theorem

Theorem 2.2 *[5] Every two strings with the same k-mer composition can be transformed into each other by transpositions and rotations.*

Proof: (Pevzner, 1995 [4]) Consider the de-Bruijn graph G of the spectrum S. Strings with a given spectrum correspond to Eulerian paths in the directed graph G (see Figure 2.10). Graph G is either Eulerian (i.e. contains an Eulerian cycle) or contains an Eulerian path. If it is Eulerian, then there exists a rotation of the corresponding string. In this case, the rotation corresponds simply to a choice of the initial vertex of the Eulerian cycle, and the theorem holds.

Otherwise, there is an Eulerian path. Create the 2-colored graph G^* by substituting each directed edge $a = (v, w)$ in G with two undirected edges, (v, a) colored white (dashed line in Figure 2.10), and (a, w) colored black (solid line in Figure 2.10). Obviously, each alternating path in the new graph G^* is a directed path in G and vice versa. According to Theorem 2.1, order exchanges and reflections generate all Eulerian paths in G^*, and therefore, all strings with a given k-mer composition.

Notice that the transposition operation corresponds to order exchange in G^*. On the other hand, every cycle in G^* is even (due to the way the graph was built). Therefore there are no order reflections in G^*. Therefore, transpositions and rotations generate all strings with a given k-mer composition. ■

2.5.3 Ukkonen's Theorem and Unique Reconstruction

Given the above theorem, we can define the necessary and sufficient conditions for the unique reconstruction of a sequence, given its k-spectrum. To achieve a unique reconstruction we should prevent the two situations that enable the transpositions and rotations described above, i.e.:

- No interleaving (k-1)-mer.
- The first and last (k-1)-mers should not be identical.

Given a spectrum of all k-mers, achieved from an n-long target, we want to find the probability of unique reconstruction, based on Ukkonen's theorem. We again assume that the nucleotides are independent and identically distributed with probability $p = \frac{1}{4}$ for each of A, C, G and T. As we claimed above:

$$\Pr(\text{non-unique reconstruction}) = \Pr(\text{rotation}) + \Pr(\text{interleaved repeats}) \qquad (2.3)$$

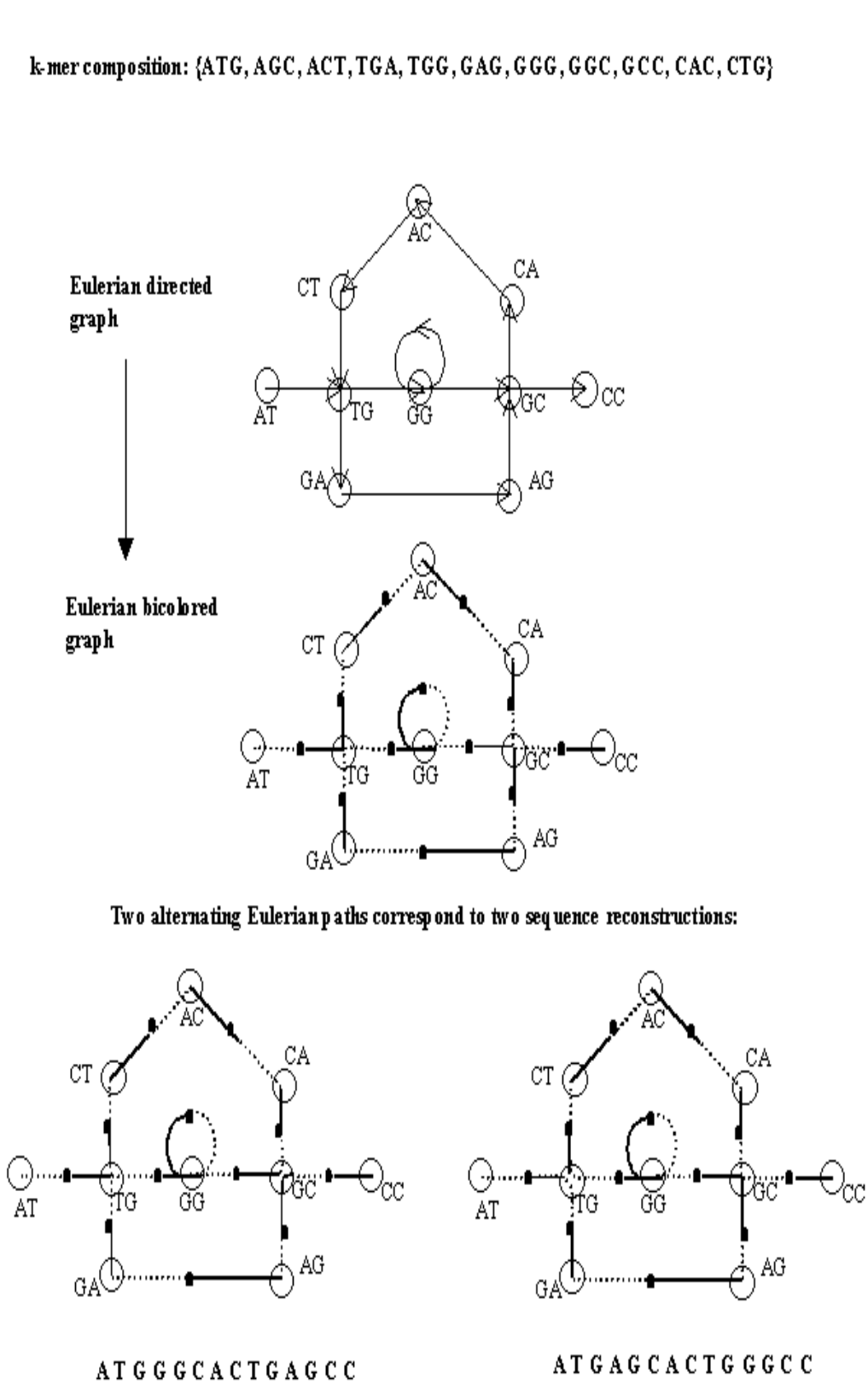

Figure 2.10: Source: [1].Ukkonen's conjecture. The sequence (Eulerian path) on the right is obtained from the sequence on the left by a transposition defined by the interleaving pairs of dinucleotides TG and GC. k-mer composition: {ATG, AGC, ACT, TGA, TGG, GAG, GGG, GGC, GCC, CAC, CTG}.

We continue, by claiming:

$$\Pr(\text{rotation}) \simeq 4^{-(k+1)} \tag{2.4}$$

$$\Pr(\text{interleaved repeats}) \simeq \text{ (number of interleaved repeats)} \times 4^{-2(k-1)} \times (\frac{3}{4})^2 \tag{2.5}$$

Where the last factor $(\frac{3}{4})^2$ results from the requirement to have the paths F_2 and F_4 different from each other.

The number of interleaved repeats can be estimated by:

$$\text{number of interleaved repeats} \simeq \binom{n}{4} + \binom{n}{3} \simeq \binom{n+1}{4} \tag{2.6}$$

Since we are interested in unique reconstruction probability, we get $\frac{n^4}{4^{2k}} \simeq 1$ and thus $n \simeq \sqrt{4^k}$. We conclude that the length of a uniquely reconstructible target is approximately $\sqrt{\text{chip size}}$. It can be shown that this bound is a lower as well as an upper bound.

2.6 Positional SBH

2.6.1 Introduction

As we saw in previous sections, the resolving power of DNA arrays is rather low. With 64-Kb arrays, only DNA fragments of up to 200 bp can be reconstructed in a single SBH experiment. To improve the resolving power of SBH, several authors ([13], [7]) have suggested enhancements of SBH based on adding location information to the spectrum. With additional experimental work, the measurement of approximate positions of every k-mer in the target sequence is registered. This additional information makes the reconstruction less ambiguous. Thus, in addition to the previously obtained k-mer composition, we can now know, for each k-mer in the spectrum, its allowed positions along the target sequence. This method is called *positional sequencing by hybridization* (PSBH).

The reduction of PSBH to Eulerian path problem still applies, but for each edge in Pevzner's graph we now have constraints restricting its position in the Eulerian path. Mathematically, this gives rise to the *positional Eulerian path problem* (PEP): Given a directed graph with a list of allowed positions on each edge, decide if there exists an Eulerian path, in which each edge appears in one of its allowed positions. In other words, not only do we demand an existence of an Eulerian path in Pevzner's graph, we also add constrains for each edge about its order in the path. Hannenhalli et al. [8] showed that PEP is NP-complete, even if all the lists of allowed positions are intervals of equal length. Note that this leaves open the complexity of PSBH. They also gave a polynomial algorithm for the problem when the range of the allowed position for any edge is bounded by a constant.

Ben-Dor, Pe'er, Shamir and Sharan [6] addressed the problem of positional sequencing by hybridization in the case that the number of allowed positions per k-mer is bounded, and the positions need not be consecutive. A linear time algorithm was suggested for solving PEP, hence, PSBH problem, in the case that each edge is allowed at most two positions (2-PEP problem). On the negative side, it was shown that PSBH is NP-complete, even if each k-mer has at most three allowed positions and multiplicity one. This was done by proving that the 3-positional Eulerian path problem (3-PEP) is NP-complete and showing a reduction from 3-PEP to 3-PSBH.

2.6.2 Preliminaries

Let $D = (V, E)$ be a directed, simple and finite graph. We denote $m = |E|$ throughout. For a vertex $v \in V$, we define its in-neighbors to be the set of all vertices from which there is an edge directed into v. We denote this by $N_{in}(v) = \{u : (u, v) \in E\}$. The out-neighbors $N_{out}(v)$ and out-degree are similarly defined.

Let $E = \{e_1, ...e_m\}$ and let P be a function mapping each edge of D to a non-empty set of integer labels in the range $[1...m]$ (its allowed positions). We call such a pair (D, P) a *positional graph*. If for all e, $|P(e)| \leq k$, then (D, P) is called a *k-positional graph*. An Eulerian path π in D is said to be compliant with the positional graph (D, P), if $\pi^{-1}(e) \in P(e)$ for each $e \in E$, that is, each edge in π occupies an allowed position. The positional SBH problem is defined as follows:

Problem 2.4 Positional SBH:
INPUT: A multi-set S of p-long strings, where for each $s \in S$, a set $P(s) \subseteq \{0, ...|S| - 1\}$.
QUESTION: Is S the p-spectrum of some string X, such that for each $s \in S$ its positions along X is in $P(s)$?

If the set of allowed positions for each string is of size of at most k, then the corresponding problem is called k-positional SBH, or k-PSBH. k-PSBH is reducible to k-PEP in an obvious manner.

2.6.3 A Linear Algorithm for 2-PEP

The authors [6] give a linear algorithm for 2-PEP, by proving the linear reduction:

$$2 - PEP \propto_{linear} 2 - SAT$$

Let $(D = (V, E), P)$ be the input 2-positional graph. For every $1 \leq t \leq m$ define $\Delta(t)$ to be the set of edges allowed at position t. For every vertex $v \in V$, define $In(v, t)$ as the set of t-labelled edges entering v. Thus, $In(v, t) \equiv \{(u, v) : (u, v) \in \Delta(t)\}$ and similarly, $Out(v, t) \equiv \{(v, u) : (v, u) \in \Delta(t)\}$.

The following preprocessing step is done initially:
while $\exists t$ such that $\Delta(t) = \{e\}$ is a singleton, do: Set $P(e) \leftarrow \{t\}$.

Lemma 2.3 *The preprocessing step does not change the set of Eulerian paths compliant with* (D, P).

If at any stage it is discovered that some set $\Delta(t)$ is empty, then the algorithm outputs False and halts, since no edge can be labelled t. The preprocessing phase can be implemented in linear time. In the following we denote by (D, P) the positional graph obtained after the preprocessing step.

Lemma 2.4 *In* (D, P) *each position is allowed by at most two edges.*

Proof: The preprocessing step ensures that if for some position t, $|\Delta(t)| = 1$, then $e \in \Delta(t)$ satisfies $|P(e)| = 1$. Let R be the set of positions t with $|\Delta(t)| = 1$, and let $r = |R|$. Then there are $m - r$ positions which $|\Delta(t)| \geq 2$, and $r' \geq r$ edges with $|P(e)| = 1$. Thus,

$$2(m-r) \leq \sum_{t \notin R} |\Delta(t)| = \sum_{t} |\Delta(t)| - r = \sum_{e} |P(e)| - r = 2m - r' - r \leq 2(m-r)$$

Hence, $r = r'$ and each label $t \notin R$ occurs exactly twice, implying that $|\Delta(t)| \in \{1, 2\}$ for all t. ■

We say that vertex v is *fixed to position* t in (D, P) if $In(v, t) = \Delta(t)$ or $Out(v, t+1) = \Delta(t+1)$. That is, any Eulerian path compliant with (D, P) must visit v at position t. Define boolean variables X_e^t for all $t \in P(e)$. Further define the following boolean clauses:

(1)	X_e^t	for every $e \in E$ where $P(e) = \{t\}$
(2)	$X_{e_1}^t \oplus X_{e_1}^t$	for every $t \notin R$ where $\Delta(t) = \{e_1, e_2\}$
(3)	$X_e^{t_1} \oplus X_e^{t_2}$	for every $e \in E$ where $P(e) = \{t_1, t_2\}$
(4)	$X_{(a,b)}^t \Leftrightarrow X_{(b,c)}^t$	for every $t \in P(a, b), (t+1) \in P(b, c)$ and b is not fixed to t
(5)	$\overline{X}_{(u,v)}^t$	for every $t \in P(u, v), t < m$, s.t. $Out(v, t+1) = \emptyset$
(6)	$\overline{X}_{(u,v)}^t$	for every $t \in P(u, v), t > 1$, s.t. $In(u, t-1) = \emptyset$

Lemma 2.5 *There is an Eulerian path compliant with* (D, P) *iff the set of clauses (1)-(6) is satisfiable.*

Proof: Suppose that satisfying truth assignment Φ exists. We shall assign an edge e to position t iff $\Phi(X_e^t) = True$. Clauses (1) and (2) guarantee that exactly one edge is assign to each position. Clauses (1) and (3) guarantee that each edge is assigned to exactly one position, and that this position is allowed to the edge.

It remains to show that the above assignment of edges to positions yields a path in D. Suppose the contrary that both $X^t_{(a,b)}$ and $X^{t+1}_{(b',c')}$ are assigned true, with $b \neq b'$. Then clauses (5) guarantee the existence of an edge $(b,c) \in \Delta(t+1)$, while clauses (6) guarantee the existence of an edge $(a',b') \in \Delta(t)$. Hence, b is not fixed to t and a contradiction follows from clauses (4). Thus, Φ defines an Eulerian path compliant with (D,P).

The converse can be shown in a similar way. ■

Theorem 2.6 *2-PEP is solvable in linear time.*

Proof: The preprocessing step is linear. The number of clauses (1)-(6) is $O(m)$. Each XOR cluase in (2)-(3) and each equivalence clause in (4) can be written as two OR clauses. Moreover, one can generate all clauses in linear time. By Lemma 2.3 the problem is reduced to an instance of 2-SAT, which is solvable in linear time [9]. ■

2.6.4 3-PEP is NP-Complete

Lemma 2.7 *3-PEP is NP-Complete.*

We introduce only the main idea of the proof. The full proof details can be found in [6]. To prove NP-hardness we show a reduction from 3-SAT. Let F be a 3-CNF formula with N variables $x_1, ..., x_n$ and M clauses $C_1, ..., C_M$. We assume, w.l.o.g., that each clause contains three distinct variables, and that all $2N$ literals occur in F. We shall construct a directed graph $D = (V, E)$ and a map P from E to integer sets of size at most 3, such that F is satisfiable iff (D, P) has a compliant Eulerian path.

For each occurrence of a variable in the formula, a *special vertex* is introduced. Special vertices corresponding to the same literal form a *literal path*. Two literals paths of a variable and its negation are connected in parallel to form a *variable subgraph*. For each clause in the formula, the corresponding special vertices are connected by three edges to form a *clause triangle*. Finally, for each special vertex we introduce a triangle incident on it, called a *bypass triangle*. Figure 2.11 shows an example of this constructed graph.

The sets of allowed positions are chosen so that they force every compliant Eulerian path to visit the literal paths one by one. A compliant Eurlerian path corresponds to a satisfying truth assignment. When a special vertex is visited, either its clause triangle, or its bypass triangle are traversed. Traversing the clause triangle while passing through a certain literal's path corresponds to this literal satisfying the clause. We make sure that for one of x_i or $\overline{x}_i$, no clause triangle is visited while passing through its literal path. Eventually, we enable visiting all unvisited bypass triangles. In this way, it can be proven that the 3-CNF formula F is satisfiable iff the constructed graph (D, P) is a "yes" instance of the 3-PEP probelm.

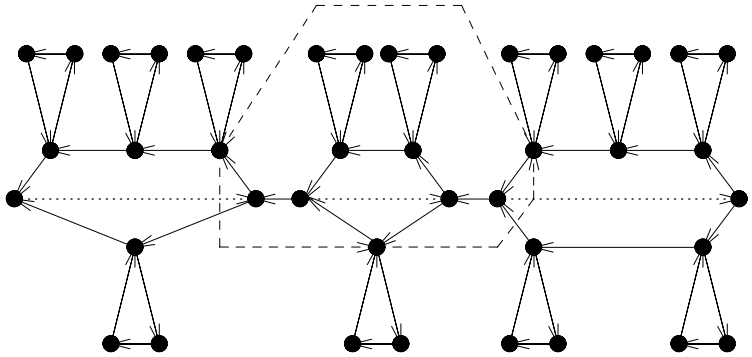

Figure 2.11: Source: [6]. A schematic sketch of the main elements in the graph construction. The figure includes three variable subgraphs, with the first variable (whose subgraph is rightmost) having three positive occurrences and two negated occurrences, etc. One of the clause triangle is also drawn, using a dashed line.

2.6.5 3-Positional SBH is NP-Complete

It can be shown that the problem of sequencing by hybridization with at most 3 positions per spectrum element is NP-Complete, even if each element in the spectrum is unique. The proof is by reduction from (3,4)-PEP and is not given here. The reader is referred to [6].

Bibliography

[1] P.A. Pevzner. *Computational Molecular Biology - An Algorithmic Approach.* The MIT Press, 2000.

[2] P.A. Pevzner. *l-tuple DNA Sequencing: Computer Analysis.* Journal of Biomolecular Structure and Dynamics, vol. 7: 63-73, 1989.

[3] P.A. Pevzner and R. Lipshutz. *Towards DNA Sequencing Chips.* in Proceedings of the 19^{th} International Conference on Mathematical Foundations of Computer Science, vol. 841 of Lecture Notes in Computer Science: 143-158, Kosice, Slovakia, 1994.

[4] P.A. Pevzner. *DNA Physical Mapping and Alternating Eulerian Cycles in Colored Graphs.* Algorithmica, vol. 13: 77-105, 1995.

[5] E. Ukkonen. *Approximate String Matching with q-grams and Maximal Matches.* Theoretical Computer Science, vol. 92: 191-211, 1992.

[6] A. Ben-Dor, I. Pe'er, R. Shamir and R. Sharan. *On the Complexity of Positional Sequencing by Hybridization.* Proceedings of the 10^{th} Annual Symposium on Combinatorial Pattern Matching, LNCS 1645: 88-100, 1999.

[7] L. M. Adleman. *Location Sensitive Seuquencing of DNA.* Technical report, University of Southern California, 1998.

[8] S. Hannenhalli, P. Pevzner, H. Lewis and S. Skiena. *Positional Sequencing by Hibridization.* Computer Applications in the Biosciences, vol. 12: 19-24, 1996.

[9] B. Apsvall, M.F. Plass and R.E. Tarjan. *A Linear Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas.* Information Processing Letters, vol. 8(3): 121-123, 1979.

[10] R. Dramanac, I. Labat, I. Brunker and R. Crkvenjakov. *Sequencing of Megabase Plus DNA by Hybridization: Theory of the Method.* Genomics, vol. 4: 114-128, 1989.

[11] R. Dramanac, L. Hood, R. Crkvenjakov. *DNA Sequence Determination ny Hybridization: a Strategy for Effiecient Large - Scale Sequenceing.* Science, vol. 260: 1649-1652, 1993.

[12] S. K. Stein. *The Mathematician as an Explorer.* Scientific American: 149-159, May 1961.

[13] N. Broude, T. Sano, C. Smith and C. Cantor. *Enhanced DNA Sequencing by Hybridization.* Proc. of the National Academy of Science USA, vol. 91: 3072-3076, 1994.

[14] `http://www.affymetrix.com/products/gc_exp1_content.html/`

[15] `http://www.affymetrix.com/technology/tech_probe.html/`