# 11 DNA arrays

This exposition is based on the following sources, which are recommended reading:

1. M.B. Eisen, P.T. Spellman, P.O. Brown and D. Botstein, Cluster analysis and display of genome-wide expression patterns, PNAS, 95:14863-14868, 1998.
2. Pavel Penzer, Computational Molecular biology - an algorithmic approach, MIT Press, 2000, chapter 5.
3. Ron Shamir, Analysis of Gene Expression Data, lectures 1 and 4, 2002.

## 11.1 DNA arrays

- Also known as: biochips, DNA chips, oligo arrays, DNA microarrays or gene arrays.
- An array is an orderly arrangement of (spots of) samples.
- Samples are either DNA or DNA products.
- Each spot in the array contains many copies of the sample.
- Array provides a medium for matching known and unknown DNA samples based on base-pairing (*hybridization*) rules and for automating the process of identifying the unknowns.
- Sample spot size in microarray less than 200 microns and an array contains thousands of spots.
- Microarrays require specialized robotics and imaging equipment.
- High-throughput biology: a single DNA chip can provide information on thousands of genes simultaneously.

## 11.2 Two possible formats

We are given an unknown *target* nucleic acid sample and the goal is to detect the identity and/or abundance of its constituents using known *probe* sequences. Single stranded DNA probes are called *oligo-nucleotides* or *oligos*.

There are two different formats of DNA chips:

- Format I: The target (500-5000 bp) is attached to a solid surface and exposed to a set of probes, either separately or in a mixture. The earliest chips where of this kind, used for *oligo-fingerprinting*.
- Format II: An array of probes is produced either *in situ* or by attachment. The array is then exposed to sample DNA. Examples are *oligo-arrays* and *cDNA* microarrays.

In both cases, the free sequence is fluorescently or radioactively labeled and hybridization is used to determine the identity/abundance of complementary sequences.

## 11.3 Oligo arrays $C(l)$

The simplest oligo array $C(l)$ consists of all possible oligos of length $l$ and is used e.g. in *sequencing by hybridization* (SBH).

| $C(4)$ | $A$<br>$A$ | $A$<br>$T$ | $A$<br>$G$ | $A$<br>$C$ | $T$<br>$A$ | $T$<br>$T$ | $T$<br>$G$ | $T$<br>$C$ | $G$<br>$A$ | $G$<br>$T$ | $G$<br>$G$ | $G$<br>$C$ | $C$<br>$A$ | $C$<br>$T$ | $C$<br>$G$ | $C$<br>$C$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $AA$ | | | | | | | | | | | | | | | | |
| $AT$ | | | | | | | | | | | | | | | | |
| $AG$ | | | | | | | | | | | | | | | | |
| $AC$ | | | | | | | | | | | | | | | | |
| $TA$ | | | | | | | | | | | | | | | | |
| $TT$ | | | | | | | | | | | | | | | | |
| $TG$ | | | | | | | | | | | | | | | | |
| $TC$ | | | | | | | | | □ | | | | | | | |
| $GA$ | | | | | | | | | | | | | | | | |
| $GT$ | | | | | | | | | | | | | | | | |
| $GG$ | | | | | | | | | | | | | | | | |
| $GC$ | | | | | | | | | | | | | | | | |
| $CA$ | | | | | | | | | | | | | | | | |
| $CT$ | | | | | | | | | | | | | | | | |
| $CG$ | | | | | | | | | | | | | | | | |
| $CC$ | | | | | | | | | | | | | | | | |

Example: oligo at □: $TCGA$

## 11.4 cDNA microarrays

The aim of this technology is to analyze the expression of thousands of genes in a single experiment and provides measurements of the differential expression of these genes.

Here, each spot contains, instead of short oligos, identical cDNA clones, which represents a gene. (Such *complementary DNA* is obtained by reverse transcription from some known mRNA.) The target is the unknown mRNA extracted from a specific cell. As most of the mRNA in a cell is translated into a protein, the total mRNA in a cell represents the genes expressed in the cell.

Since cDNA clones are much longer than the short oligos otherwise used, a successful hybridization with a clone is an almost certain match. However, because an unknown amount of cDNA is printed at each spot, one cannot directly associate the hybridization level with a transcription level and so cDNA chips are limited to to comparisons of a reference extract and a target extract.

## 11.5 Affymetrix chips

Affymetrix produces oligo arrays with the goal of capturing each coding region as specifically as possible. The length of the oligos is usually less than 25 bases. The density of oligos on a chip can be very high and a 1cm × 1cm chip can easily contain 100 000 types of oligos.

The chip contains both “coding” oligos and “control” oligos, the former corresponding to perfect matches to known targets and the controls corresponding to matches with one perturbed base.

When reading the chip, hybridization levels at controls are subtracted from the level of match probes to reduce the number of false positives. Actual chip designs use 10 match- and 10 mismatch probes for each target gene.

Today, Affymetrix offers chips for the entire (known) human or yeast genomes.

## 11.6 Oligo fingerprinting

Format I chips were the first type used, namely for *oligo fingerprinting* which is, in a sense, the opposite to what Affymetrix chips do. Such a chip consists of a matrix of target DNA and is exposed to a solution containing many identical oligos.

After the positions in the matrix have been recorded at which hybridization of the tagged oligos has occurred, the chip can be heated to separate the oligos from the target DNA and the experiment can be repeated with a different type of oligo.

Finally, we obtain a data matrix $M$, with each row representing a specific target DNA from the matrix and each column representing an oligo probe.

Example: cDNA's extracted from a tissue. Cluster cDNA's according to their fingerprints and then sequence representatives from each cluster to obtain a sequence that identifies the gene.

## 11.7 Manufacturing oligo arrays

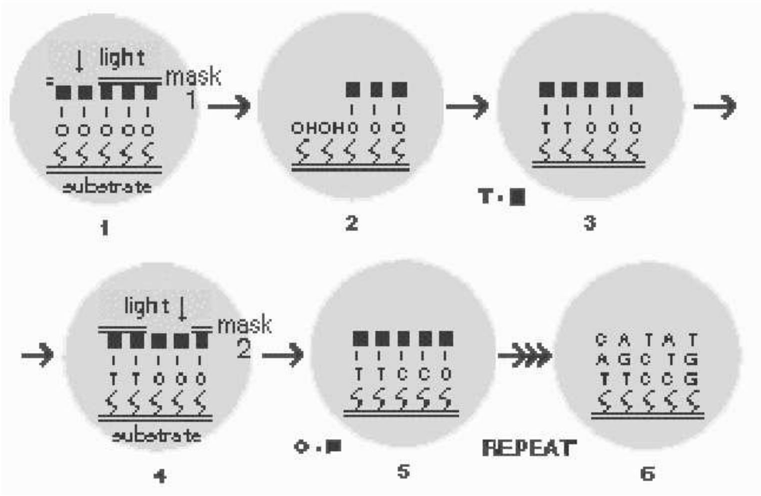
1. Start with a matrix created over a glass substrate.

2. Each cell contains a growing "chain" of nucleotides that ends with a *terminator* that prevents chain extension.

3. Cover the substrate with a mask and then illuminate the uncovered cells, breaking the bonds between the chains and their terminators.

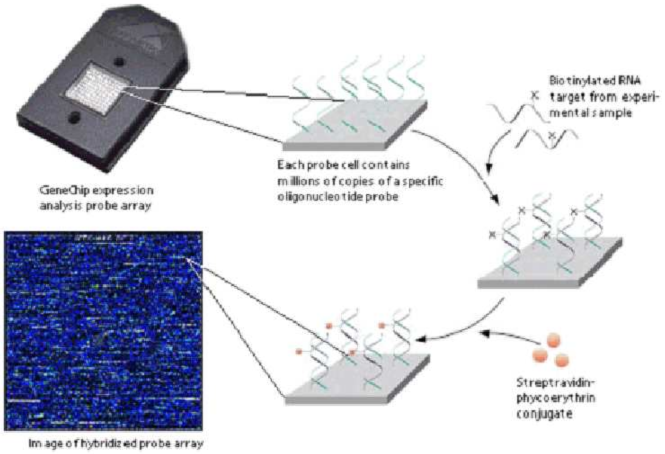
4. Expose the substrate to a solution of many copies a specific nucleotide base so that each of the unterminated chains is extended by one copy of the nucleotide base and a new terminator.

5. Repeat using different masks.



Exposure to light replaces the terminators by hydrogen bonds (1–2), and (3) bonds forms with nucleotide bases provided in a solution, and then the process is repeated with a different base (4–6).

## 11.8 Experiment with a DNA chip



Labeled RNA molecules are applied to the probes on the chip, creating a fluorescent spot where hybridization has occurred.

## 11.9 Functional genomics

With the sequencing of more and more genomes, the question arises of how to make use of this data. One area that is now opening up is *functional genomics*, the understanding of the functionality of specific genes, their relations to diseases, their associated proteins and their participation in biological processes.

The functional annotation of genes is still at an early stage: e.g., for the plant Arabidopsis (whose sequence was recently completed), the functions of 40% of the genes are currently unknown.

Functional genomics is being addressed using high-throughput methods: global gene expression profiling ("transcriptome analysis") and wide-scale protein profiling ("proteome analysis").

## 11.10 Gene expression

The existing methods for measuring gene expression are based on two biological assumptions:

1. *The transcription level of genes indicates their regulation:* Since a protein is generated from a gene in a number of stages (transcription, splicing, synthesis of protein from mRNA), regulation of gene expression can occur at many points. However, we assume that most regulation is done only during the transcription phase.
2. *Only genes which contribute to organism fitness are expressed*, in other words, genes that are irrelevant to the given cell under the given circumstances etc. are not expressed.

Genes affect the cell by being *expressed*, i.e. transcribed into mRNA and translated into proteins that react with other molecules.

From the pattern of expression we may be able to deduce the function of an unknown gene. This is especially true, if the pattern of expression of the unknown gene is very similar to the pattern of expression of a gene with known function.

Also, the level of expression of a gene in different tissues and at different stages is of significant interest.

Hence, it is highly interesting to analyze the *expression profile* of genes, i.e. in which tissues and at what stages of development they are expressed.

## 11.11 cDNA Clustering

It is not easy to determine which genes are expressed in each tissue, and at what level:

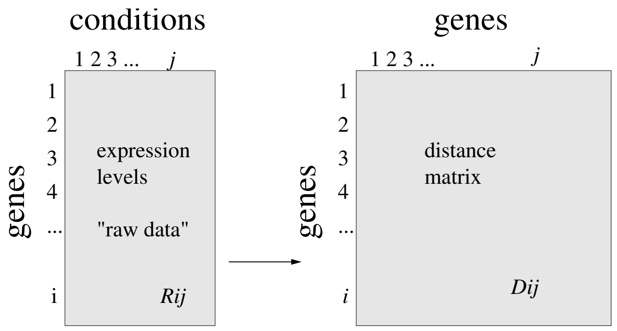
An average tissue contains more than 10 000 expressed genes, and their expression levels can vary by a factor of 10 000. Hence, we need to extract more than $10^5$ transcripts per tissue. There are about 100 different types of tissue in the body and we are interested in comparing different growth stages, disease stages etc., and so we should analyze more than $10^{10}$ transcripts.

⇒ Sequencing all cDNA's is infeasible and we need cheap, efficient and large scale methods.

## 11.12 Representation of gene expression data

Gene expression data is represented by a *raw data matrix* $R$, where each row corresponds to one gene and each column represents one tissue or condition. Thus, $R_{ij}$ is the expression level for gene $i$ in condition $j$. The values can be ratios, absolute values or distributions.

Before it is analyzed, the raw data matrix is preprocessed to compute a *similarity* or *distance* matrix.



## 11.13 Clustering

The first step in analyzing gene expression data is clustering.

Clustering methods are used in many fields. The goal in a clustering problem is to group elements (in our case genes) into clusters satisfying:

1. *Homogeneity:* Elements inside a cluster are highly similar to each other.
2. *Separation:* Elements from different clusters have low similarity to each other.

There are two types of clustering methods:

- *Agglomerative* methods build clusters by looking at small groups of elements and performing calculations on them in order to construct larger groups.
- *Divisive* methods analyze large groups of elements in order to divide the data into smaller groups and eventually reach the desired clusters.

Why would we want to cluster gene expression data? We assume that:

- Distinct measurements of same genes cluster together.
- Genes of similar function cluster together.
- Many cluster-function specific insights are gained.

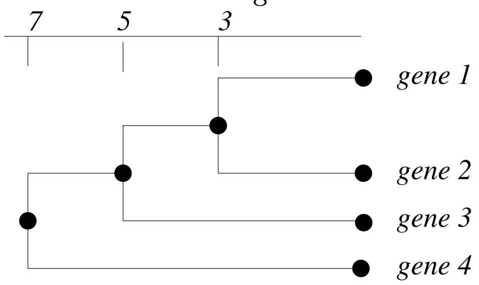
## 11.14 Hierarchical clustering

This approach attempts to place the input elements in a tree hierarchy structure in which distance within the tree reflects element similarity.

To be precise, the hierarchy is represented by a tree and the actual data is represented by the leaves of the tree. The tree can be rooted or not, depending on the method used.

Distance matrix → Dendrogram

| gene | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 3 | 5 | 7 |
| 2 | 3 | 0 | 5 | 7 |
| 3 | 5 | 5 | 0 | 7 |
| 4 | 7 | 7 | 7 | 0 |



## 11.15 Average linkage

Average linkage is similar to Neighbor-Joining, except that when computing the new distances of created clusters, the sizes of clusters that are merged are taken into consideration. This algorithm was developed by Lance and Williams (1967) and Sokal and Michener (1958).

1. Input: The distance matrix $D_{ij}$, initial cluster sizes $n_r$.
2. Iteration $k$: The same as in NJ, except that the distance from a new element $t$ is defined by:

$$D_{it} := D_{ti} := \frac{n_r}{n_r + n_s} D_{ir} + \frac{n_s}{n_r + n_s} D_{is}$$

## 11.16 Non-Hierarchical clustering

Given a set of input vectors. For a given clustering $P$ of them into $k$ clusters, let $E^P := \sum_c \sum_{v \in c} D(v, z_c)$ denote the *solution cost function*, where $z_c$ is the *centroid* (average vector) of the cluster $c$ and $D(v, z_c)$ is the distance from $v$ to $z_c$.

The *k-means clustering* due to Macqueen (1965) operates as follows:

1. Initialize an abitrary partition $P$ into $k$ clusters.
2. For each cluster $c$ and element $e$:
   Let $E^P(c, e)$ be the cost of the solution if $e$ is moved to $c$.
3. Pick $c, e$ so that $E^P(c, e)$ is minimum.
4. Move $e$ to $c$, if it improves $E^P$.
5. Repeat until no further improvement is achieved.

## 11.17 Application to fibroblast cells

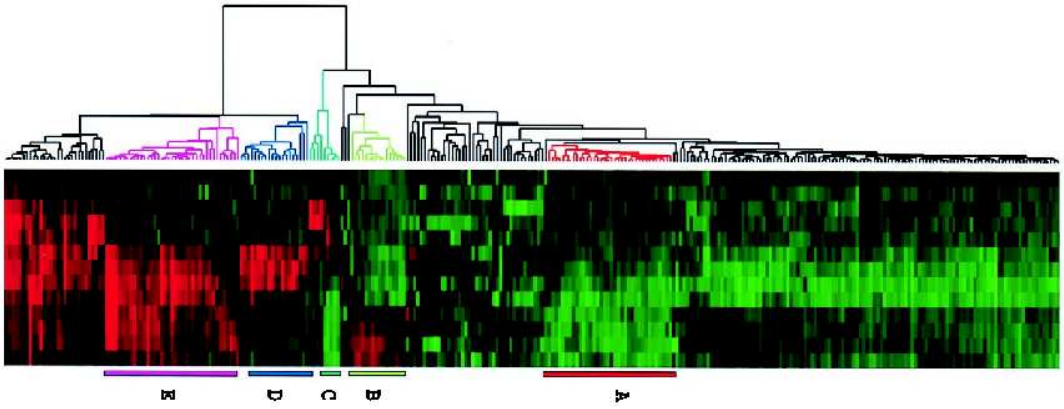
Eisen et al. (1998) performed a series of experiments on real gene expression data. One goal was to check the growth response of starved human fibroblast cells, which were then given serum. The expression level of about $n = 8600$ genes were monitored over $N_{cond} = 13$ time points.

The original data of test to reference ratios was first log transformed, and then normalized to have mean 0 and variance 1. Let $N_{ij}$ denote these normalized levels. A similarity matrix was constructed from $N_{ij}$ as follows:

$$S_{kl} := \frac{\sum_{j=1}^{n} N_{kj} N_{jl}}{N_{cond}},$$

where $N_{cond}$ is the number of conditions checked.

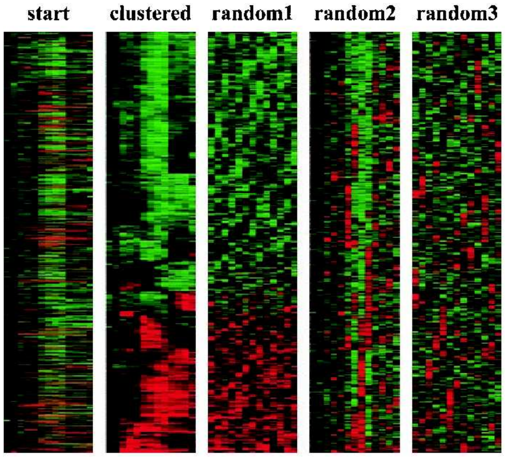
The average linkage method was then used to generate the following tree:



The Dendrogram resulting from the starved human fibroblast cells experiment. Five major clusters can be seen, and many non clustered genes. The cells in the five groups server similar functions: (A) cholesterol bio-synthesis, (B) the cell cycle, (C) the immediate-early response, (D) signaling and angiogenesis, and (E) wound healing and tissue remodeling.
(Color scale red-to-green corresponds to higher-to-lower expression level than in the control state.)

## 11.18 Testing the significance of the clusters

A standard method for testing the significance of clusters is to randomly permute the input data in different ways.



Original expression data (1), clustered data (2), and the results of clustering after random permutations within rows (3), within columns (4) and within both (5).

## 11.19 Sequencing by Hybridization (SBH)

Originally, the hope was that one can use DNA chips to sequence lage unknown DNA fragments using a large array of short probes:

1. Produce a chip $C(l)$ spotted with all possible probes of length $l$ ($l = 8$ in the first SBH papers),
2. Apply a solution containing many copies of a fluorescently labeled DNA target fragment to the array.
3. The DNA fragments hybridize to those probes that are complementary to substrings of length $l$ of the fragment
4. Detect probes that hybridize with the DNA fragment and obtain the $l$-tuple composition of the DNA fragment
5. Apply a combinatorial algorithm to reconstruct the sequence of the DNA target from the $l$-tuple composition

## 11.20 The Shortest Superstring Problem

SBH provides information of the $l$-tuples present in a target DNA sequence, but not their positions. Suppose we are given the *spectrum* $S$ of all $l$-tuples of a target DNA sequence, how do we construct the sequence?

This is a special case of the *Shortest Common Superstring Problem (SCS)*: A *superstring* for a given set of strings $s_1, s_2, \ldots, s_m$ is a string that contains each $s_i$ as a substring. Given a set of strings, finding the shortest superstring is NP-complete.

Define $overlap(s_i, s_j)$ as the length of a maximal prefix of $s_j$ that matches a suffix of $s_i$. The SCS problem can be cast as a Traveling Salesman Problem in a complete directed graph $G$ with $m$ vertices $s_1, s_2, \ldots, s_m$ and edges $(s_i, s_j)$ of length $-overlap(s_i, s_j)$.

## 11.21 The SBH graph

SBH corresponds to the special case in which all substrings have the same length $l$. We say that two *SBH* probes $p$ and $q$ *overlap*, if the last $l - 1$ letters of $p$ coincide with the first $l - 1$ of $q$.

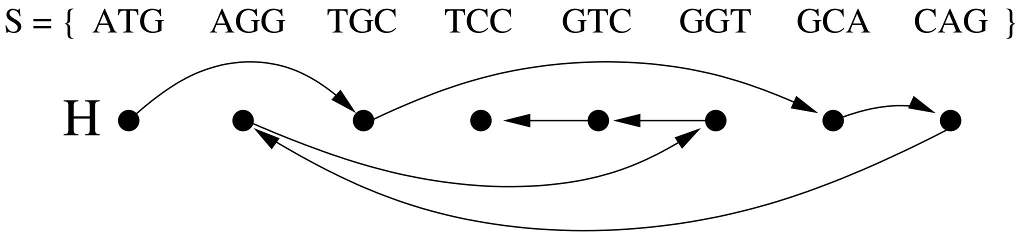
Given the spectrum $S$ of a DNA fragment, construct the directed graph $H$ with vertex set $S$ and edge set

$$E = \{(p, q) \mid p \text{ and } q \text{ overlap}\}.$$

There exists a one-to-one correspondence between paths that visit each vertex of $H$ at least once and the DNA fragments with the spectrum $S$.

## 11.22 Example of the SBH graph

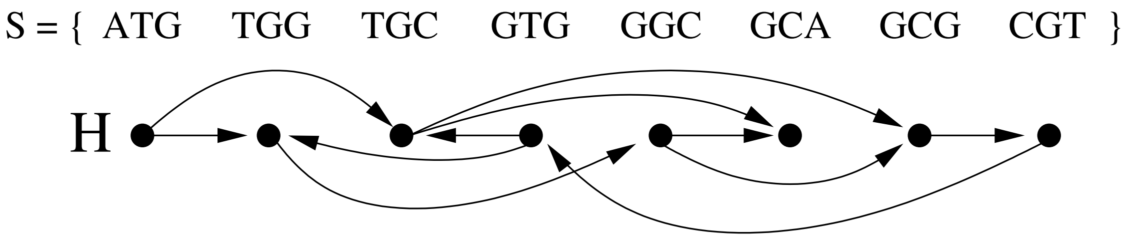
Vertices: $l$ tuples of the spectrum $S$, edges: overlapping $l$-tuples:



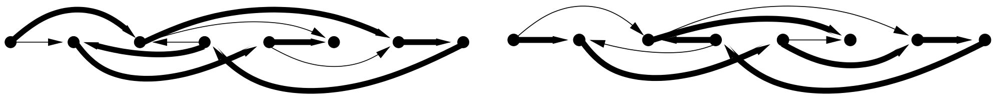
The path visiting all vertices corresponds to the sequence reconstruction `ATGCAGGTCC`.

A path that visits all nodes of a graph exactly once is called a *Hamiltonian* path. Unfortunately, the Hamiltonian Path Problem is NP-complete, so for larger graphs we cannot hope to find such paths.

## 11.23 Second example of the SBH graph



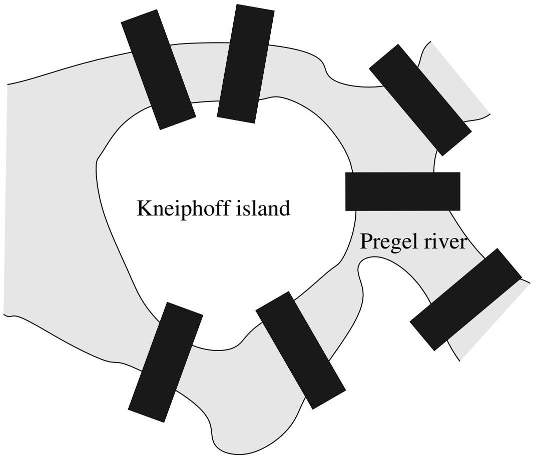
This example has two different Hamiltonian paths and thus two different reconstructed sequences:



`ATGCGTGGCA` `ATGGCGTGCA`

## 11.24 Euler Path

Leonard Euler wanted to know whether there exists a path that uses all seven bridges in Königsberg exactly once:



Birth of graph theory...

## 11.25 SBH and the Eulerian Path Problem

Let $S$ be the spectrum of a DNA fragment. We define a graph $G$ whose set of nodes consists of *all possible* $(l-1)$*-tuples.*

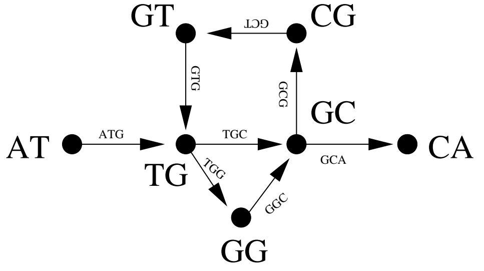
We connect one $l-1$-tuple $v = v_1 \dots v_{l-1}$ to another $w = w_1 \dots w_{l-1}$ by a directed edge $(v, w)$, if the spectrum $S$ contains an $l$-tuple $u$ with prefix $v$ and suffix $w$, i.e. such that $u = v_1 \dots v_{l-1} w_1 = v_{l-1} w_1 \dots w_{l-1}$.

Hence, in this graph the probes correspond to edges and the problem is to find a path that visits all *edges* exactly once, i.e., an *Eulerian path.*

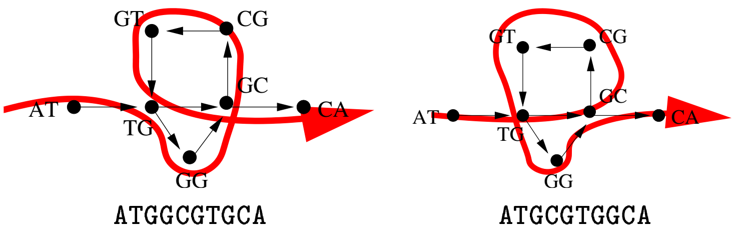
Finding all Eulerian paths is simple to solve.

(To be precise, the Chinese Postman Problem (visit all edges at least once in a minimal tour) can be efficiently solved for directed or undirected graphs, but not in a graph that contains both directed and undirected edges.)

S = { ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT }



Vertices represent $(l-1)$-tuples, edges correspond to $l$-tuples of the spectrum. There are two different solutions:



## 11.26 Eulerian graphs

A directed graph $G$ is called *Eulerian*, if it contains a cycle that traverses every edge of $G$ exactly once.

A vertex $v$ is called *balanced*, if the number of edges entering $v$ equals the number of edges leaving $v$, i.e. $indegree(v) = outdegree(v)$. We call $v$ *semi-balanced*, if $|indegree(v) - outdegree(v)| = 1$.

**Theorem** A directed graph is Eulerian, iff it is connected and each of its vertices is balanced.

**Lemma** A connected directed graph is Eulerian, iff it contains at most two semi-balanced nodes.

## 11.27 When does a unique solution exist?

Problem: Given a spectrum $S$. Does it possess a unique sequence reconstruction?

Consider the corresponding graph $G$. If the graph $G$ is Eulerian, then we can decompose it into *simple* cycles $C1, \dots, C_t$, that is, cycles without self-intersections. Each edge of $G$ is used in exactly one cycle, although nodes can be used in many cycles. Define the *intersection graph* $G_I$ on $t$ vertices $C_1, \dots C_t$, where $C_i$ and $C_j$ are connected by $k$ edges, iff they have precisely $k$ original vertices in common.

**Lemma** Assume $G$ is Eulerian. Then $G$ has only one Eulerian cycle iff the intersection graph $G_I$ is a tree.

## 11.28 Probability of unique sequence reconstruction

What is the probability that a randomly generated DNA fragment of $n$ can be uniquely reconstructed using a DNA array $C(l)$? In other words, how large must $l$ be so that a random sequence of length $n$ can be uniquely reconstructed from its $l$-tuples?

We assume that the bases at each position are chosen independently, each with probability $p = \frac{1}{4}$.

Note that a repeat of length $\geq l$ will always lead to a non-unique reconstruction. We expect about $\binom{n}{2}p^l$ repeats of length $\geq l$. Note that $\binom{n}{2}p^l = 1$ implies $l = \log_{\frac{1}{p}} \binom{n}{2}$.

$\Longrightarrow$ For a given $l$ one should choose $n \approx \sqrt{2 \cdot 4^l}$, but not larger. (However, this is a very loose bound and a much tighter bound is known.)

## 11.29 SBH currently infeasible

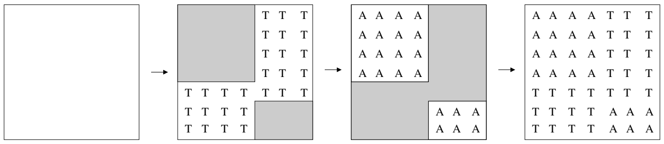
The Eulerian path approach to SBH is currently infeasible due to two problems:

- Errors in the data
  - False positives arise, when the the target DNA hybridizes to a probe even though an exact match is not present
  - False negatives arise, when an exact match goes undetected
- Repeats make the reconstruction impossible, as soon as the length of the repeated sequence is longer than the word length $l$

Nevertheless, ideas developed here are employed in a new approach to sequence assembly that uses sequenced reads and a Eulerian path representation of the data (Pavel Pevzner, Recomb'2001).

## 11.30 Masks for VLSIPS

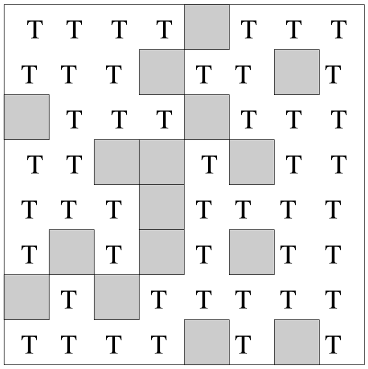
DNA arrays can be manufactured using *VLSIPS, very large scale immobilized polymer synthesis.* In VLSIPS, probes are grown one layer of nucleotides at a time through a photolithographic process. In each step, a different mask is used and only the unmasked probes are extended by the current nucleotide. All probes are grown to length $l$ in $4l$ steps.

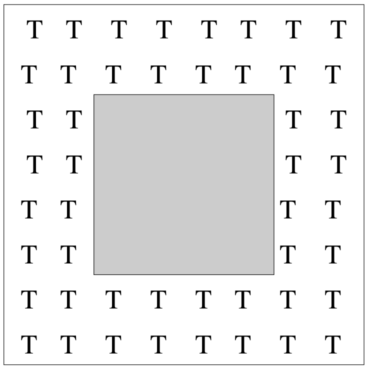


*Problem:* Due to diffraction, internal reflection and scattering, masked spots near an edge of the mask can be unintentionally illuminated.

**Idea:** To minimize the problem, design masks that have minimal border length!

For example, consider the $8 \times 8$ array for $l = 3$. Both of the following two masks add a $T$ to $\frac{1}{4}$ of the spots, with borders of very different length:





border length 58

border length 16

## 11.31 The $l$-bit Gray code

In the above example, we can mask $\frac{1}{4}$ of all spots using a mask that has a boundary of length $4 \cdot l$. Can we arrange the spots so that this minimal value is attained for every mask?

An *$l$-bit Gray code* is a permutation of the binary numbers $0, \ldots, 2-1$ such that any two neighboring numbers differ in exactly one bit.

The 4-bit "reflected" Gray code is:

$$\begin{array}{cccccccccccccccc}
\multicolumn{16}{c}{\xrightarrow{\hspace{10em}}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0
\end{array}$$

This is generated recursively from the 1-bit code $G_1 = \{0, 1\}$ using:

$$G_l = \{g_1, g_2, \ldots, g_{2^l-1}, g_{2^l}\} \longrightarrow$$

$$\mathtt{G}_{l+1} = \{0g_1, 0g_2, \ldots, 0g_{2^l-1}, 0g_{2^l}, 1g_{2^l}, 1g_{2^l-1}, \ldots, 1g_2, 1g_1\}.$$

## 11.32 The two-dimensional Gray code

We want to construct a two-dimensional Gray code for strings of length $l$ over the alphabet $\{A, C, G, T\}$, in which every $l$-tuple is present and differs from each of its four neighbors in precisely one position.

$$\text{Start: } G_1 := \begin{bmatrix} A & T \\ C & G \end{bmatrix}.$$

The induction step $G_l \to G_{l+1}$:

$$\text{Let } G_l = \begin{bmatrix} g_{1,1} & \ldots & g_{1,2^l} \\ & \ldots & \\ g_{2^l,1} & \ldots & g_{2^l,2^l} \end{bmatrix} \text{ and set}$$

$$G_{l+1} := \begin{bmatrix}
Ag_{1,1} & \ldots & Ag_{1,2^l} & Tg_{1,2^l} & \ldots & Tg_{1,1} \\
 & \ldots & & & & \\
Ag_{2^l,1} & \ldots & Ag_{2^l,2^l}, & Tg_{2^l,2^l} & \ldots & Tg_{2^l,1}, \\
Gg_{2^l,1} & \ldots & Gg_{2^l,2^l} & Cg_{2^l,2^l} & \ldots & Cg_{2^l,1} \\
 & \ldots & & & & \\
Gg_{1,1} & \ldots & Gg_{1,2^l}, & Cg_{1,2^l} & \ldots & Cg_{1,1}
\end{bmatrix}.$$

## 11.33 Additional ideas

**SBH with universal bases** Use universal bases such as inosine that stack correctly, but don't bind, and thus play the role of "don't care" symbols in the probes. Arrays based on this idea can be achieve the information-theoretic lower bound of the number of probes required for unambiguous reconstruction of an abitrary DNA string of length $n$. (The full $C(l)$ array has redundancies that can be eliminated using such universal bases.) (Preparata et al. 1999)

**Adaptive SBH** If a sequencing by hybridization is not successful, analyze the critical problems and then build a new array to overcome them. Skiena and Sundaram (1993) gives theoretical bounds for the number of rounds needed for sequence reconstruction (in the error free case).

**SBH-style shotgun sequencing** The idea is to collect sequence reads from the target DNA sequence using traditional sequencing methods and then to treat each such read of length $k$ as a set of $k - l + 1$ individual $l$-tuples, with $l = 30$, say. Then, the Eulerian path method is used. Idury and Waterman suggested this in 1995 and it leads to an efficient assembly algorithm in the error-free case. More recent work by Pevzner and others (2001) has led to promising software.

**Fidelity probes for DNA arrays** As a quality control measure when manufacturing a DNA chip, one can produce fidelity probes that have the same sequence as probes on the chip, but are produced in a different order of steps. A known target is hybridized to these probes and the result reflects the quality of the array. Hubbel and Pevnzer (1999) describe a combinatorial method for designing a small set of fidelity probes that can detect variations and can also indicate which manufacturing steps caused the errors.