

Microarray Layout and the Quadratic Assignment Problem

Sérgio A. de Carvalho Jr.^{1,2,3} Sven Rahmann^{1,2}

¹Algorithms and Statistics for Systems Biology, Genome Informatics,
Technische Fakultät, Universität Bielefeld, Germany

²International NRW Graduate School in Bioinformatics and Genome Research

³Graduiertenkolleg Bioinformatik

German Conference on Bioinformatics, 2006

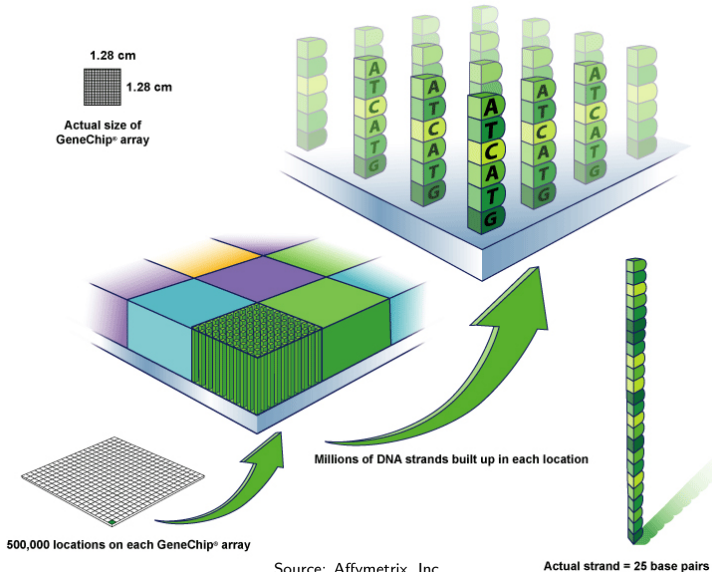
Outline

- 1 Introduction to Microarray Layout
- 2 Conflict Index Model
- 3 New Approach: Quadratic Assignment Problem (QAP)

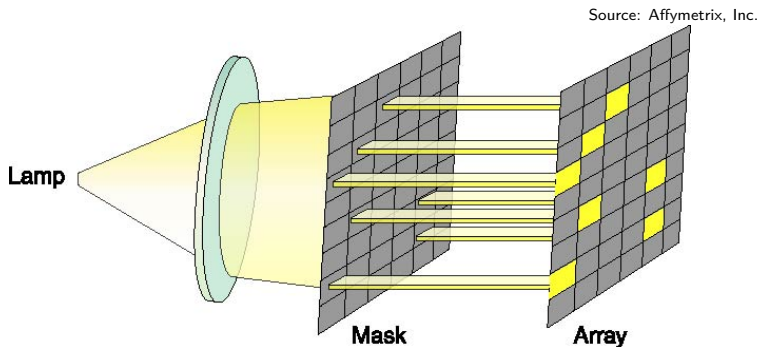
Outline

- 1 Introduction to Microarray Layout
- 2 Conflict Index Model
- 3 New Approach: Quadratic Assignment Problem (QAP)

High-Density Oligonucleotide Microarrays



Probe Synthesis with Photolithographic Masks



- Probes are synthesized on the chip in a **series of steps**
- Each step **appends a particular nucleotide** to selected regions
- Selection occurs by exposure to light directed by a **mask**

Deposition Sequence and Probe Embeddings

p_1 ACT	p_2 CTG	p_3 GAT
p_4 TCC	p_5 GAC	p_6 GCC
p_7 TAC	p_8 CGT	p_9 AAT

$S =$ ACGTACGTACGT
 $\varepsilon_1 =$ -----
 $\varepsilon_2 =$ -----
 $\varepsilon_3 =$ -----
 $\varepsilon_4 =$ -----
 $\varepsilon_5 =$ -----
 $\varepsilon_6 =$ -----
 $\varepsilon_7 =$ -----
 $\varepsilon_8 =$ -----
 $\varepsilon_9 =$ -----

Deposition Sequence and Probe Embeddings

p_1 A ^{blue} CT	p_2 CTG	p_3 GAT
p_4 TCC	p_5 GAC	p_6 GCC
p_7 TAC	p_8 CGT	p_9 A ^{blue} AT

$S =$ A^{blue}CGTACGTACGT
 $\varepsilon_1 =$ A^{blue}-----
 $\varepsilon_2 =$ -----
 $\varepsilon_3 =$ -----
 $\varepsilon_4 =$ -----
 $\varepsilon_5 =$ -----
 $\varepsilon_6 =$ -----
 $\varepsilon_7 =$ -----
 $\varepsilon_8 =$ -----
 $\varepsilon_9 =$ A^{blue}-----

Deposition Sequence and Probe Embeddings

p_1 ACT	p_2 CTG	p_3 GAT
p_4 TCC	p_5 GAC	p_6 GCC
p_7 TAC	p_8 CGT	p_9 AAT

$S =$ A**C**GTACGTACGT
 $\varepsilon_1 =$ A-----
 $\varepsilon_2 =$ -**C**-----
 $\varepsilon_3 =$ -----
 $\varepsilon_4 =$ -----
 $\varepsilon_5 =$ -----
 $\varepsilon_6 =$ -----
 $\varepsilon_7 =$ -----
 $\varepsilon_8 =$ -**C**-----
 $\varepsilon_9 =$ A-----

Deposition Sequence and Probe Embeddings

p_1 ACT	p_2 CTG	p_3 GAT
p_4 TCC	p_5 GAC	p_6 GCC
p_7 TAC	p_8 CGT	p_9 AAT

$S =$ A**C**GTACGTACGT
 $\varepsilon_1 =$ A-----
 $\varepsilon_2 =$ -C-----
 $\varepsilon_3 =$ --**G**-----
 $\varepsilon_4 =$ -----
 $\varepsilon_5 =$ --**G**-----
 $\varepsilon_6 =$ --**G**-----
 $\varepsilon_7 =$ -----
 $\varepsilon_8 =$ -**C****G**-----
 $\varepsilon_9 =$ A-----

Deposition Sequence and Probe Embeddings

p_1 ACT	p_2 CTG	p_3 GAT
p_4 TCC	p_5 GAC	p_6 GCC
p_7 TAC	p_8 CGT	p_9 AAT

$S =$ ACGTACGTACGT
 $\varepsilon_1 =$ A-----
 $\varepsilon_2 =$ -C-----
 $\varepsilon_3 =$ --G-----
 $\varepsilon_4 =$ ---T-----
 $\varepsilon_5 =$ --G-----
 $\varepsilon_6 =$ --G-----
 $\varepsilon_7 =$ ---T-----
 $\varepsilon_8 =$ -CGT-----
 $\varepsilon_9 =$ A-----

Deposition Sequence and Probe Embeddings

p_1 ACT	p_2 CTG	p_3 GAT
p_4 TCC	p_5 GAC	p_6 GCC
p_7 TAC	p_8 CGT	p_9 AAT

$S =$ ACGTACGTACGT
 $\varepsilon_1 =$ A-----C-T-----
 $\varepsilon_2 =$ -C-----T--G-
 $\varepsilon_3 =$ --G-A--T-----
 $\varepsilon_4 =$ ---T-C---C--
 $\varepsilon_5 =$ --G-A-----C--
 $\varepsilon_6 =$ --G--C---C--
 $\varepsilon_7 =$ ---TAC-----
 $\varepsilon_8 =$ -CGT-----
 $\varepsilon_9 =$ A---A-----T

Deposition Sequence and Probe Embeddings

p_1 ACT	p_2 CTG	p_3 GAT
p_4 TCC	p_5 GAC	p_6 GCC
p_7 TAC	p_8 CGT	p_9 AAT

$S =$ ACGTACGTACGT

$\varepsilon_1 =$ A----C-T----

$\varepsilon_2 =$ -C-----T--G-

$\varepsilon_3 =$ --G-A--T-----

$\varepsilon_4 =$ ---T-C---C--

$\varepsilon_5 =$ --G-A-----C--

$\varepsilon_6 =$ --G--C---C--

$\varepsilon_7 =$ ---TAC-----

$\varepsilon_8 =$ -CGT-----

$\varepsilon_9 =$ A---A-----T

$\varepsilon'_9 =$ A-----A--T

$\varepsilon''_9 =$ ----A---A--T

$\varepsilon'''_9 =$ A---A--T-----

Right-most:

Left-most:

Unintended Illumination Problem

p_1 ACT	p_2 CTG	p_3 GAT
p_4 TCC	p_5 GAC	p_6 GCC
p_7 TAC	p_8 CGT	p_9 AAT

- **Untargeted spots** can be accidentally activated
 - Diffraction of light
 - Internal reflection
- Production of defective probes
- More likely near the **borders** between masked and unmasked spots: **border conflict**

Border Length Minimization Problem (Hannenhalli et al., 2002)

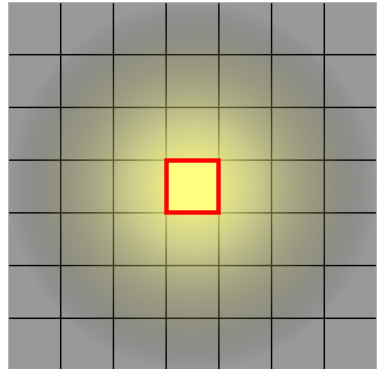
Find arrangement of the probes and embeddings with minimum number of border conflicts over all masks

Outline

- 1 Introduction to Microarray Layout
- 2 Conflict Index Model
- 3 New Approach: Quadratic Assignment Problem (QAP)

Motivation

- Border Length measures the quality of a particular mask
 - We are more interested in a **per-probe measure**
- Practical considerations:
 - a) Stray light might damage probes as far as **three cells away** from the targeted spot
 - b) Imperfections **in the middle** of a probe are more harmful than in its extremities



ATGACTACCATGCAGTACAACATAC

Definition

Conflict Index of a probe p

$$\mathcal{C}(p) := \sum_{t=1}^T \left(\omega(p, t) \sum_{nbs.p'} \delta(p, p', t) \right)$$

Distance-dependent weights

$$\delta(p, p', t) := \begin{cases} (d(p, p'))^{-2} & \text{if } p' \text{ is unmasked at step } t, \\ 0 & \text{otherwise,} \end{cases}$$

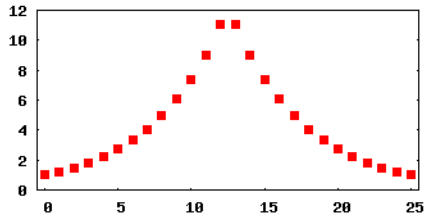
where $d(p, p')$ is the [Euclidean distance](#) between the spots of p and p' .

0.06	0.08	0.10	0.11	0.10	0.08	0.06
0.08	0.13	0.20	0.25	0.20	0.13	0.08
0.10	0.20	0.50	1.00	0.50	0.20	0.10
0.11	0.25	1.00	p	1.00	0.25	0.11
0.10	0.20	0.50	1.00	0.50	0.20	0.10
0.08	0.13	0.20	0.25	0.20	0.13	0.08
0.06	0.08	0.10	0.11	0.10	0.08	0.06

Definition

Conflict Index of a probe p

$$\mathcal{C}(p) := \sum_{t=1}^T \left(\omega(p, t) \sum_{p'} \delta(p, p', t) \right)$$



Position-dependent weights

$$\omega(p, t) := \begin{cases} c \cdot \exp(\theta \cdot \lambda(p, t)) & \text{if } p \text{ is masked at step } t, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\lambda(p, t) := 1 + \min(b_{p,t}, \ell_p - b_{p,t}),$$

$b_{p,t}$ denotes the number of nucleotides synthesized up to and including step t , ℓ_p is the length of probe p , $c > 0$ and $\theta > 0$ are constants.

New Problem

Conflict Index Minimization Problem

Find placement of the probes and embeddings such that

$$\sum_p \mathcal{C}(p) \rightarrow \min$$

Border Length and Conflict Index

Redefine δ and ω as

$$\delta(p, p', t) := \begin{cases} 1 & \text{if } p' \text{ is a direct neighbor of } p \\ & \text{and is unmasked at step } t, \\ 0 & \text{otherwise} \end{cases}$$
$$\omega(p, t) := \begin{cases} 1/2 & \text{if } p \text{ is masked at step } t, \\ 0 & \text{otherwise} \end{cases}$$

- Then $\sum_p \mathcal{C}(p) = \sum_{t=1}^T \mathcal{B}_t$
- Border length and conflict indices are equivalent for this choice of δ and ω
- For our choices, they are not equivalent but still **correlated**
 - A good layout has low border lengths and conflict indices

Outline

- 1 Introduction to Microarray Layout
- 2 Conflict Index Model
- 3 New Approach: Quadratic Assignment Problem (QAP)

Previous Work: Place and Re-embed

The problem has been traditionally approached in two phases:

- 1) **Placement** of probes given a fixed embedding
- 2) **Re-embedding** of probes once a placement is fixed

Placement: Row-epitaxial (Kahng *et al.*, 2003)

- Spots are filled in a pre-defined order
 - Select probe from a list Q such that conflicts with filled spots are minimized
- Restrict the maximum size of Q (e.g. $Q = 20\,000$)

Re-embedding: several algorithms (Kahng *et al.*, 2002, 2003)

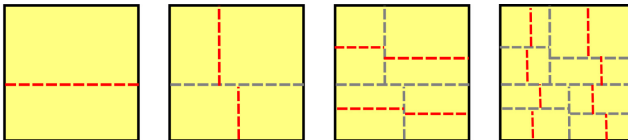
- Based on the Optimum Single Probe Embedding (OSPE)
 - Re-embed a probe **optimally** in regards to its neighbors
 - Dynamic programming, like a sequence alignment

Previous Work: Partitioning

- The placement problem can be **partitioned**
 - Divide the chip into sub-regions; assign sub-sets of probes to each sub-region
 - Sub-regions are processed independently, and can be **recursively partitioned**
 - A placement algorithm is called on each final sub-region

Pivot Partitioning (Carvalho & Rahmann, 2006)

- Alternate **horizontal** and **vertical** partitions
- Allow sub-regions to have different sizes



Quadratic Assignment Problem (QAP)

Definition

- Given $n \times n$ real-valued matrices $F = (f_{ij}) \geq 0$ and $D = (d_{kl}) \geq 0$
- Find a permutation π of $\{1, 2, \dots, n\}$ such that

$$\sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{\pi(i)\pi(j)} \rightarrow \min$$

Example: Facility Location Problem

- Assign n facilities to n locations
- f_{ij} : flow of materials from facility i to j
- d_{kl} : distance between locations k and l
- π : one-to-one assignment with minimum cost

QAP Formulation of Placement Problem

QAP

- Given $n \times n$ real-valued matrices F and D
- Find a permutation π of $\{1, 2, \dots, n\}$ such that

$$\sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{\pi(i)\pi(j)} \rightarrow \min$$

Placement as a QAP

- Given a set of probes with **fixed** embeddings
- Find **placement** on the chip such that

$$\sum_k C(k) \rightarrow \min$$

QAP Formulation of Placement Problem

Goal: find a placement with

$$\sum_k \mathcal{C}(k) \rightarrow \min$$

“Flow” f_{ij} : distance between spots i and j

$$f_{ij} := \begin{cases} (d(i, j))^{-2} & \text{if spot } j \text{ is “near” spot } i, \\ 0 & \text{otherwise} \end{cases}$$

“Distance” d_{kl} : conflicts between probes k and l

$$d_{kl} := \sum_{t=1}^T d_{klt},$$

$$d_{klt} := \begin{cases} c \cdot \exp(\theta \cdot \lambda(k, t)) & \text{if } k \text{ is masked and } l \text{ unmasked in step } t, \\ 0 & \text{otherwise} \end{cases}$$

QAP Heuristics

- The placement problem can be modeled as a QAP
- But QAP is known to be NP-hard
 - Generally impossible to solve (to optimality) for $n \geq 20$
- Several **heuristics** exist

GRASP (Li, Pardalos and Resende, 1994)

- Greedy Randomized Adaptive Search Procedure
- Comprised of two phases
 - 1) Construction: builds a random feasible solution
 - 2) Local search: search a local optimum in the neighborhood
- GRASP with Path-Relinking (Oliveira *et al.*, 2004)

Results on Small Artificial Chips

Border Length minimization

Dim	Random	Row-epitaxial			GRASP with Path-Relinking		
	Cost	Cost	Red.	Time	Cost	Red.	Time
6×6	1 989.20	1 714.60	13.80	0.01	1 672.20	15.94	2.73
7×7	2 783.20	2 354.60	15.40	0.02	2 332.60	16.19	6.43
8×8	3 721.20	3 123.80	16.05	0.03	3 099.13	16.72	12.49
9×9	4 762.00	3 974.80	16.53	0.05	3 967.20	16.69	25.96
10×10	5 985.20	4 895.60	18.20	0.06	4 911.40	17.94	47.57
11×11	7 288.40	5 954.40	18.30	0.10	5 990.73	17.80	87.48
12×12	8 714.00	7 086.20	18.68	0.11	7 159.80	17.84	152.42

Dim: chip dimension

Cost: total border length

Red.: reduction in %

Time: running time in seconds

Results on Small Artificial Chips

Conflict Index minimization

Dim	Random	Row-epitaxial			GRASP with Path-Relinking		
	Cost	Cost	Red.	Time	Cost	Red.	Time
6×6	524.28	495.15	5.56	0.05	467.08	10.91	3.68
7×7	558.25	521.90	6.51	0.07	489.32	12.35	8.84
8×8	590.51	551.84	6.55	0.09	515.69	12.67	19.48
9×9	613.25	568.62	7.28	0.11	533.79	12.96	38.83
10×10	628.50	576.49	8.28	0.11	539.69	14.13	73.09
11×11	642.72	588.91	8.37	0.12	551.41	14.21	145.67
12×12	656.86	598.21	8.93	0.12	561.21	14.56	249.19

Dim: chip dimension

Cost: [average conflict index](#)

Red.: reduction in %

Time: running time in seconds

Applications

- QAP is good for small regions...
 - But not feasible for an entire microarray chip

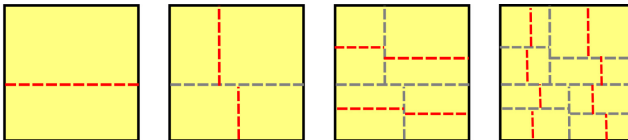
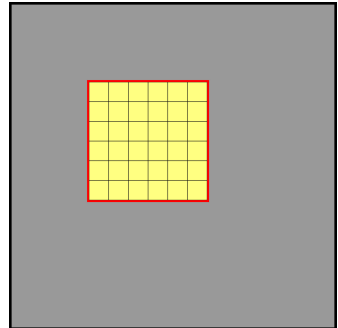
Applications

- QAP is good for small regions...
 - But not feasible for an entire microarray chip
- Two applications:
 - 1) Combine it with a [partitioning algorithm](#)



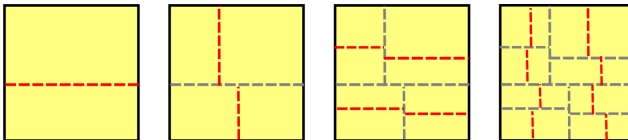
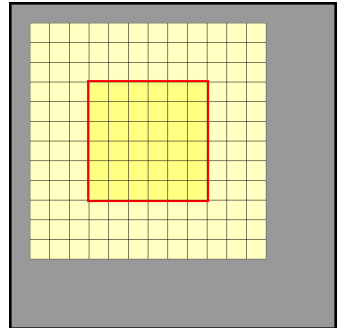
Applications

- QAP is good for small regions...
 - But not feasible for an entire microarray chip
- Two applications:
 - 1) Combine it with a [partitioning algorithm](#)
 - 2) Use it as a [post-placement optimization](#) inside a sliding-window



Applications

- QAP is good for small regions...
 - But not feasible for an entire microarray chip
- Two applications:
 - 1) Combine it with a [partitioning algorithm](#)
 - 2) Use it as a [post-placement optimization](#) inside a sliding-window



Summary

- Conflict Index
 - New model for evaluating microarray layouts
- New approach to placement
 - Based on the Quadratic Assignment Problem
- Challenges
 - Faster or better QAP heuristics?
 - Adapt QAP for post-placement optimization
 - Formulation considering all embeddings

Auf Wiedersehen!

More info on

<http://gi.cebitec.uni-bielefeld.de/assb/chiplayout>

QAPLIB

<http://www.seas.upenn.edu/qaplib>

- Thanks to Peter Hahn (University of Pennsylvania, USA)
- And **thank you** for your attention!