Algorithms for Universal DNA Tag Array Design and Optimization

Ion Mandoiu
Computer Science & Engineering Department
University of Connecticut

Overview
- Background on DNA Microarrays
- Universal DNA Tag Arrays
- Tag Set Design Problem
- Tag Assignment Problem
- Conclusions

Watson-Crick Complementarity
- Four nucleotide types: A,C,T,G
  - A’s paired with T’s (2 hydrogen bonds)
  - C’s paired with G’s (3 hydrogen bonds)

DNA Microarrays
- Labeled DNA/RNA mixture flushed over array of probes
- Laser activation of fluorescent labels
- Optical scanning used to identify probes with complements in the mixture

Applications
- Gene expression (transcription analysis)
- Genomic-based microorganism identification
- Single Nucleotide Polymorphism (SNP) genotyping

Gene Expression
- Cells express different subsets of genes under different environments
  - Transcription ➔ mRNA ➔ Translation ➔ Protein
Two-Color Technique

- Sample labeled RED
- Control labeled GREEN
- YELLOW probes hybridize to both sample and control
- BLACK probes hybridize to neither

Microarray Technologies

- Arrays of cDNAs
  - Obtained by reverse transcription from Expressed Sequence Tags (ESTs)
- Oligonucleotide arrays
  - Short (20-60bp) synthetic DNA strands

Robotic cDNA Arrayers

- Pin Technology
- Quill Pen Technology
- Ink jet Technology
- Pin Ring Technology

In-Place Oligonucleotide Synthesis

Probes to be synthesized
Overview

- Background on DNA Microarrays
- **Universal DNA Tag Arrays**
- Tag Set Design Problem
- Tag Assignment Problem
- Conclusions

Universal DNA Tag Arrays

- cDNA arrays are cheap, but work only for gene expression
- Oligonucleotide arrays are flexible, but expensive unless produced in large quantities
- “Programmable” oligonucleotide arrays [Brenner 97, Morris et al. 98]
  - Array consisting of application independent oligonucleotides called tags
  - Two-part “reporter” probes: application specific primers ligated to antitags
  - Detection carried by a sequence of reactions separately involving the primer and the antitag part of reporter probes

Universal Tag Array Experiment

Universal Tag Array Advantages

- Cost effective
  - Same array used in many analyses → can be mass produced
- Fast to customize
  - Only need to synthesize new set of reporter probes
- Reliable
  - Solution phase hybridization better understood than hybridization on solid support

Overview

- Background on DNA Microarrays
- **Universal DNA Tag Arrays**
- Tag Set Design Problem
- Tag Assignment Problem
- Conclusions

Tag Set Requirements

- Hybridization constraints
  - (H1) Antitags hybridize strongly to complementary tags
  - (H2) No antitag hybridizes to a non-complementary tag
  - (H3) Antitags do not cross-hybridize to each other
Hybridization Model

- **Melting temperature Tm:** temperature at which 50% of duplexes are in hybridized state
  - 2-4 rule
    \[ Tm = 2 \#(As and Ts) + 4 \#(Cs and Gs) \]
  - More accurate models exist, e.g., the near-neighbor model

Tag Set Design Problem

- [Ben-Dor et al. 00] Conservative formalization of (H1)+(H2) based on nucleation complex theory and 2-4 rule
  - (C1) Every tag must have total weight ≥ h
  - (C2) No substring of weight ≥ c appears twice in selected tags

Where
- \( w(A)=w(T)=1, w(C)=w(G)=2 \)
- \( c, h \) are given constant (Affymetrix uses “length=20” instead of “weight ≥ h” in C1)

Tag ↔ sequence of c-tokens

End pos: 2 3 4 5 6 7 8

Tag ↔ sequence of c-tokens

Token Content of a Tag

\( c=4 \)

- **CCAGATT**
- **CC**
- **CCA**
- **CAG**
- **AGA**
- **GAT**
- **GATT**

Layered c-token graph for length-l tags

Integer Program Formulation

\[
\begin{align*}
\text{maximize} & \quad \sum_{e \in \mathcal{V} \setminus \{s, t\}} x_e \\
\text{subject to} & \quad x_e = \sum_{e \in \partial (v)} g_{v} = \sum_{e \in \partial (v)} g_{v}, \quad v \in \mathcal{V} \setminus \{s, t\} \\
& \quad \sum_{e \in \partial (v)} x_e \leq 1, \quad 1 \leq i \leq N \\
& \quad x_e, g_v \in \{0, 1\}, \quad v \in \mathcal{V} \setminus \{s, t\}, e \in \mathcal{E}
\end{align*}
\]

- Maximum integer flow problem w/ set capacity constraints
- \( O(\text{hN}) \) constraints & variables, where \( N = \#\text{c-tokens} \)
Number of c-tokens

<table>
<thead>
<tr>
<th>Token type</th>
<th>Num tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;c\rightarrow S))</td>
<td>2 (G_{c,2})</td>
</tr>
<tr>
<td>(S\rightarrow c\rightarrow S))</td>
<td>4 (G_{c,3})</td>
</tr>
<tr>
<td>(&lt;c\rightarrow W))</td>
<td>2 (G_{c,1})</td>
</tr>
<tr>
<td>(S\rightarrow c\rightarrow W))</td>
<td>4 (G_{c,2})</td>
</tr>
<tr>
<td>Total</td>
<td>(G_{c} + 2G_{c,1})</td>
</tr>
</tbody>
</table>

- \(W=A\) or \(T\), \(S=C\) or \(G\)
- \(G_{n}\) = \#strings of weight \(n\)

\(G_1 = 2; G_2 = 6; G_n = 2G_{n-2} + 2G_{n-1}\)

---

Number of c-tokens

<table>
<thead>
<tr>
<th>(c)</th>
<th>Num c-tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>208</td>
</tr>
<tr>
<td>6</td>
<td>568</td>
</tr>
<tr>
<td>7</td>
<td>1552</td>
</tr>
<tr>
<td>8</td>
<td>4240</td>
</tr>
<tr>
<td>9</td>
<td>11584</td>
</tr>
<tr>
<td>10</td>
<td>31648</td>
</tr>
</tbody>
</table>

---

Packing LP Formulation

0/1 variable \(x_p\) for every s-t path \(p\)

\[
\text{maximize } \sum_{p \in \mathcal{P}} x_p \\
\text{subject to } \sum_{p \in \mathcal{P}} |p \cap V_i| x_p \leq 1, \quad 1 \leq i \leq N \\
x_p \in \{0, 1\}, \quad p \in \mathcal{P}
\]

---

Garg-Konemann Algorithm

1. \(x = 0; y = \delta\) \(\delta\) are variables of the dual LP
2. Find min weight s-t path \(p\), where weight(v) = \(y_i\) for every \(v \in V_i\)
3. While weight(p) < 1 do
   - \(M \leftarrow \max |p \cap V_i|\)
   - \(x_p \leftarrow x_p + \frac{1}{M}\)
   - For every \(i, y_i \leftarrow y_i + \frac{\epsilon}{\delta} \cdot |p \cap V_i| M\)
4. For every \(p, x_p \leftarrow x_p / (1 - \log_1^\epsilon \delta)\)

[\text{GK98}]\) The algorithm computes a factor \((1 - \epsilon)^2\) approximation to the optimal LP solution with \(N/\epsilon^2 \log_1^\epsilon N\) shortest path computations

---

LP Based Tag Set Design

1. Run Garg-Konemann and store the minimum weight paths in a list
2. Traversing the list in reverse order, pick tags corresponding to paths if they are feasible and do not share c-tokens with already selected tags
3. Mark used c-tokens and run the alphabetic tree search algorithm to select additional tags

---

Experimental Results (h=15)
**Experimental Results (h=28)**

<table>
<thead>
<tr>
<th>h</th>
<th>ε</th>
<th>Selected Tags</th>
<th>Upper Bounds</th>
<th>CPU Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>3</td>
<td>3 5 5 5</td>
<td>5.38 5</td>
<td>0.00 0.84 1.10 117.70</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4 4</td>
<td>7.67 4</td>
<td>0.00 1.36 392.42 18989.06</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>5 5 5</td>
<td>11.41 5</td>
<td>0.00 1.36 392.42 18989.06</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6 6 6</td>
<td>11.41 6</td>
<td>0.00 1.36 392.42 18989.06</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>7 7 7</td>
<td>11.41 7</td>
<td>0.00 1.36 392.42 18989.06</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>8 8 8</td>
<td>11.41 8</td>
<td>0.00 1.36 392.42 18989.06</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>9 9 9</td>
<td>11.41 9</td>
<td>0.00 1.36 392.42 18989.06</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>10 10 10</td>
<td>11.41 10</td>
<td>0.00 1.36 392.42 18989.06</td>
</tr>
</tbody>
</table>

**Periodic Tags**

- c-token uniqueness constraint in c-h code formulation is too strong [MT05]
  - c-tokens should not appear in different tags, but is OK to repeat a c-token in the same tag

- A tag \( t \) is called *periodic* if it is the prefix of \((\alpha)^n\) for some “period” \( \alpha \)
  - Periodic strings make better use of c-tokens \( t \) uses at most \(|\alpha|\) c-tokens

**c-token factor graph, c=4 (incomplete)**

**Vertex-disjoint Cycle Packing Problem**

- Given directed graph \( G \), find maximum number of vertex disjoint directed cycles in \( G \)

  - [MT 05] APX-hard even for regular directed graphs with in-degree and out-degree 2
    - \( h/2+1 \) approximation factor for tag set design problem

  - [Salavatipour and Verstraete 05]
    - Quasi-NP-hard to approximate within \( \Omega(\log^{1/4} n) \)
    - \( O(\omega^{15}) \) approximation algorithm

**Tag Set Design Algorithm**

1. Construct c-token factor graph \( G \)
2. \( T \leftarrow \{\} \)
3. For all cycles \( C \) defining periodic tags, in increasing order of cycle length, do
   - Add to \( T \) the tag defined by \( C \)
   - Remove \( C \) from \( G \)
4. Perform an alphabetic tree search and add to \( T \) tags with no c-tokens in common with \( T \)
5. Return \( T \)

**Experimental Results**

<table>
<thead>
<tr>
<th>( h )</th>
<th>ε</th>
<th>One-copy ( T )</th>
<th>Multiple-copy ( T )</th>
<th>Cycle packing ( T )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>3 5 5 5 5 5 5</td>
<td>5.38 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4 4 4 4 4 4 4</td>
<td>7.67 4 4 4 4 4 4 4</td>
<td>4 4 4 4 4 4 4 4</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>5 5 5 5 5 5 5</td>
<td>11.41 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
</tr>
</tbody>
</table>

6
**Antitag-to-Antitag Hybridization**

- Additional practical constraint (ignored by Ben-Dor et al): antitags do not cross-hybridize, including self
- Formalization in c-token hybridization model:
  - (C3) No two (anti)tags contain complementary substrings of weight $\geq c$
- Cycle packing and tree search extend easily

**Results w/ Extended Constraints**

![Table showing results](image)

**Overview**

- Background on DNA Microarrays
- Universal DNA Tag Arrays
- Tag Set Design Problem
- Tag Assignment Problem
- Conclusions

**More Possible Mis-Hybridizations**

- What can be done:
  - Leave some tags unassigned
  - Distribute primers over multiple arrays
- Here we focus on avoiding case (a), primer-to-tag hybridization

**Constraints on tag assignment**

- If primer $p$ hybridizes with tag $t$, then either $p$ or $t$ must be left un-assigned, unless $p$ is assigned to $t$

**Characterization of Assignable Sets**

- [Ben-Dor 04] Set $P$ is assignable to $T$ iff $X+Y \geq |P|$, where, in the hybridization graph induced by $P+T$
  - $X =$ number of primers incident to a degree 1 tag
  - $Y =$ number of degree 0 tags

- Y=2
  - X=1

![Diagram showing tags and primers](image)
MAPS Problem

- Maximum Assignable Primer Set (MAPS) Problem:
given primer set $P$ and tag set $T$, find maximum size assignable subset of $P$

- [Ben-Dor 04] Greedy deletion heuristic: repeatedly delete primer of maximum weight from $P$ until it becomes assignable, where
  - Potential of tag $t$ is $2^{-|P(t)|}$
  - Potential of primer $p$ is sum of potentials of conflicting tags

Universal Array Multiplexing Problem

- Multiplexing Problem: given primer set $P$ and tag set $T$, find partition of $P$ into minimum number of assignable sets

- [Ben-Dor 04] Repeatedly find approximate MAPS

Integration with Probe Selection

- In practice, several primer candidates with equivalent functionality
  - In SNP genotyping, can pick primer from either forward and reverse strand
  - In gene expression/identification applications, many primers have desired length, $T_m$, etc.

Pooled Array Multiplexing Problem

- [MPT 05] Given set of primer pools $P$ and tag set $T$, find a primer from each pool and a partition of selected primers into minimum number of assignable sets

X+Y Characterization no Longer Holds

Pooled Multiplexing Algorithms

1. Primer-Del = greedy deletion for pools similar to [Ben-Dor et al 04]
2. Primer-Del+ = same, but never delete last primer from pool unless no other choice
3. Min-Pot = select primer with min potential from each pool, then run Primer-Del
4. Min-Deg = select primer with min conflict degree, then run Primer-Del
Results: GenFlex Tags, c=8

Overview

- Background on DNA Microarrays
- Universal DNA Tag Arrays
- Tag Set Design Problem
- Tag Assignment Problem
- Conclusions

Conclusions

- New techniques for tag set design and tag assignment lead to significantly improved multiplexing rates and more reliable assays
- Other applications of universal tags:
  - Lab-on-chip, DNA-driven assembly (e.g., carbon nanotubes), DNA computing [Brenneman & Condon 02]
- Other applications of partition/assignment techniques:
  - Genotyping by mass-spectroscopy [Aumann et al 05]
  - Genotyping using l-mer arrays

Ongoing Work

- More accurate hybridization models
  - Monotonic Tm \( \Rightarrow \) c-tokens \( \Rightarrow \) factor graph
- Integrating cross-hybridization constraints in primer selection tools (string barcoding, multiplex PCR)
- Special type of universal arrays: l-mer arrays
  - Initially introduced for sequencing by hybridization, but proved impractical
  - Currently investigated for use in resequencing by hybridization
  - More promising application: high-throughput SNP genotyping
  - “Isothermic” unavoidable string sets instead of all l-mers

More Open Problems

- Settle approximation complexity of (vertex) disjoint cycle packing
  - [Salavatipour and Verstraete 05] give approximation preserving reductions between edge and vertex disjoint versions
- Establish better approximation bounds for factor graphs arising in tag set design
- Improved approximation algorithms for tag assignment problems
Acknowledgments

- Claudia Prajescu and Dragos Trinca
- UCONN Research Foundation