genetic / physical map

genetic: genes
physical: landmarks
Physical mapping

Cut the DNA in each YAC clone and clone into overlapping cosmid clones.

Select a subset of cosmid clones of minimum total length that covers the YAC DNA.

Duplicate the cosmid and then cut the copies randomly. Select and sequence short fragments and then reassemble them into a deduced cosmid string.

C: Full DNA

Cut C and clone into overlapping YAC clones.
physical mapping

- location of ‘markers’
  - restriction mapping
    - cutting sites enzymes
    - double digest problem (NP complete)
    - partial digest problem
  - hybridization mapping
    - ‘clones’ and ‘probes’
    - non-unique probes (NP hard)
    - unique probes (P time)

fragment assembly

- full sequence from fragments
  - shortest superstring
  - overlap graph
using a physical map

markers: short sequences
- restriction sites
- hybridization sites
landmarks on the genome

order or location of sequence landmarks

restriction mapping

BamH1 – GGATCC

hybridization mapping

Probes

Clones

y

x

z

w

Michael L. Raymer – Wright State University
RESTRICTION MAPPING

- double digest problem
- partial digest problem

( pictures only )
double digest problem

long segments: unknown sequences

enzyme A \{3,6,8,10\}

enzyme B \{4,5,7,11\}

A+B \{1,2,3,3,6,7\}
cassette exchange / reflection

solution not unique
characterization: interdependence of solutions
reduction from set partition
proving NP completeness (decision version)

- $X = \{1, 3, 5, 6, 9\}$
- $S = 24$
- $A = X$
- $B = \{12, 12\}$
- $A+B = X$

set partition (two parts)   restriction

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

is there a partition?   is there a restriction?
partial digest problem

(varying duration restriction experiments)

(multi)-set { 3, 5, 5, 8, 9, 14, 14, 17, 19, 22 }

backtrack algorithm
worst case exponential time
here: each probe *unique position* on genome
unique probe mapping

6 clones 1, 2, ..., 6
7 probes A, B, ..., G

probe

clone

E  B  F  C  A  G  D

1             3               5
2             4               6

6 clones 1, 2, ..., 6
7 probes A, B, ..., G

matrix representation

1 : { B, E }
2 : { B, F }
3 : { A, C, F, G }
4 : { A, C }
5 : { A, C, F }
6 : { D, G }

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Clones contain consecutive probes.
characterization using cliques

\{1,2\} \{2,3,4\} \{2,3,5\} \{3,6\}

no details in this course!
choosing a data structure

representation for permutations

\[
\{123, 132, 213, 231, 312, 321\}
\]

\[
\{123, 321\}
\]
PQ-trees

*data structure* to represent all possibilities

P permutation

Q linear order

PQ trees represent possible reorderings (permutations of probes)
clones
\{A, C, D\}  \{A, B, C, E\}
PQ-trees

equivalent representations

{ 123, 132, 213, 231, 312, 321 }

{ 123, 321 }
\(S = \{A, C, E\}\)
PQ-tree algorithm

\textit{reduce}(T,S)

- \(T\): PQ tree \sim set of permutations
- \(S\): new clone \sim set of (consecutive) probes

add requirement \(S\) to tree \(T\)

- \textit{keep }\(S\) \textit{together}

\begin{itemize}
  \item colour leaves in \(S\)
  \item apply \textit{transformations}
    \hspace{1cm}reorder to get consecutive leaves
  \item apply \textit{replacement rules} (\textit{bottom-up})
    \hspace{1cm}to add new restriction to tree
\end{itemize}

\(P\) \hspace{1cm} all leaves in \(S\)
\(Q\) \hspace{1cm} segment in \(S\)

(partially) coloured nodes
replacement rules

1. Original

2. Root

3. Non-Root

4. Root

5. Non-Root

6. Original

7. Root

8. Original

9. Root
replacement rules (2,3)

root

= lowest node having both coloured and non-coloured leaves
replacement rules (4,5)

root
⇒
(4)

non-root
⇒
(5)
S = \{A, C, E\}
probe clone

1: { B, E } 2: { B, F } 3: { A, C, F, G } 4: { A, C } 5: { A, C, F } 6: { D, G }
1: \{B, E\}
2: \{B, F\}
3: \{A, C, F, G\}

i) reorder
   ‘transformation’

ii) replacement rule
    (4)
replacement rule (2)

4 : \{ A, C \}
example

5 : \{ A, C, F \}

i) reorder & rule (1)

ii) rule (3)

iii) rule (8)

also 7th STOC, 1975.
• shortest superstring
• sequencing by hybridization
f1 = ATCCGTTGAAGCCGCGGGC
f2 = TTAACTCGAGG
f3 = TTAAGTACTGCCCCG
f4 = ATCTGTGTCCGGG
f5 = CGACTCCCCGACACA
f6 = CACAGATCCGTTGAAGCCGCGGG
f7 = CTCGAGTTAAGTA
f8 = CGCGGGCAGTACCTT

CCTCGAGTTAA------------------GCCCCGCGGCTTCAACGGAT---------------------
-------------------------TAAGTACTGCCCCG-----------------------------ATCTGTGTCTCGGG---------------------
------------------------AAGTACTGCCCCGCG--------------------------TGTGTCCGGGAGTCC
-CTCGAGTTAAGTA------------------CCCGCGGGCTTCAACGGATCTGTG--
CCTCGAGTTAAGTACTGCCCCGCGGCTTCAACGGATCTGTGTGGGAGTCC
model: shortest common superstring

shortest common superstring

given a set of fragments $F$, find the shortest string $s$ that contains every $f \in F$ as a substring

- is NP-hard
- is perhaps not what we want

“An elegant theoretical abstraction, but fundamentally flawed” – R. Karp
shortest common superstring

but: covering is not ‘uniform’

Saad Mneimneh -- http://engr.smu.edu/~saad/
repeats 😞

\[ aXbYcXdYe \implies aXdYcXbYe \]

also: \[ aXbXcXd \implies aXcXbXd \]
base errors 😞

experimental
substitutions / insertions / deletions
chimeras

ACCGT
CGTGC
TTAC
TGCCGT
TTACCGTGC

consensus

direction of strings …
tool: overlap graph

{ ACCC, CTAAAG, GACA, TACGA }

assumption:
no substrings (inclusion)

omit zero weight edges

compute overlaps: suffix tree [exact & fast]
or alignment! [error proof]
Hamilton $\sim$ visit every node (exactly) once

TACGA

ACCC

GACA

CTAAAG

length ‘superstring’ = total length strings – length path

look for *longest* Hamilton path

sad face NP complete $\Rightarrow$ heuristics
overlap graph: greedy algorithm

simple heuristic:
join strings with maximal overlap

approximation within factor ??
conjecture: factor 2 of optimal
(proofs for 4, 2.75 …)

+additional heuristics

general ‘bad’ example:

<table>
<thead>
<tr>
<th>greedy</th>
<th>optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATGC</td>
<td>C(AT)^k</td>
</tr>
<tr>
<td>TGCAT</td>
<td>(TA)^k</td>
</tr>
<tr>
<td>GCC</td>
<td>(AT)^kG</td>
</tr>
<tr>
<td>ATGCATGCC</td>
<td>greedy C(AT)^kG(TA)^k</td>
</tr>
<tr>
<td></td>
<td>best C(AT)^k+1G</td>
</tr>
<tr>
<td></td>
<td>4k+2</td>
</tr>
<tr>
<td></td>
<td>2k+4</td>
</tr>
</tbody>
</table>
overlap graph: problems

AGTATTGGCAATC  AATCGATG
TTGGCAATCACT  ATGCAAAACCT
AGTATTGGCAATC AATCGATGCAAAACCTTTTGG

length 36 weak link 0
‘greedy’

AGTATTGGCAATC  CCTTTTGG
AATCGATG  TTGGCAATCACT
ATGCAAAACCT
AGTATTGGCAATCGATGCAAAACCTTTTGGCAATC

length 37 weak link 3
‘topological sorting’
ideal world

consensus
probabilistic models

• how much of the genome is covered?

\[ E(\text{not covered}) = e^{-R} \]
\[ R = \frac{N \cdot L}{G} \quad \text{redundancy} \]
\[ L \quad \text{clone length} \]
\[ N \quad \text{number of clones} \]
\[ G \quad \text{genome length} \]

• probability of islands (contig’s)

expected number of islands \( Ne^{-R(1-\theta)} \)
\[ \theta \quad \text{overlap factor} \]
sequencing by hybridization

all possible probes of length \( \ell \)
hybridization: determine substrings
reconstruct from (multi-)set of substrings

\( \ell = 3 \)

ATTGAC
SBH example

as before: overlap graph (not a good choice)

‘characteristic triplets’

\[ \ell = 3 \]

\{ ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT \}

ATGGCGTGCA

Hamilton approach: all nodes
(overlap \( \ell - 1 \))
SBH example

as before: overlap graph (not a good choice)

‘characteristic triplets’

\( \ell = 3 \)

\{ ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT \}

\[ \text{triplet}=\text{node} \]

another solution

Hamilton approach: all nodes
(overlap \( \ell-1 \))
we can do better with same problem:

$$\ell = 3$$

$$\{ \text{ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT} \}$$

**Euler approach:** edges (overlap $\ell-1 = \text{node}$)

linear ☺

**triplet=edge**
\( \ell = 3 \)

\{ ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT \}

**Euler approach:**

- Even degree nodes (except start+finish)

**Graph representation:**

```
ATG → TG → GT → CG → GCA
```

```
ATG → TG → GTT → GG → GC → CAG
```

**Path examples:**

- \( \text{ATGGCGTGCA} \)
- \( \text{ATGCGTGACGA} \)

**Euler approach:**

Edges

**Graph:**

- Nodes: AT, TG, GT, CG, GCA, ATG, GG, GC, CA
- Edges:
  - AT → TG
  - TG → GT
  - GT → CG
  - CG → GCA
  - GCA → ATG
  - ATG → TG
  - TG → GT
  - GT → CG
  - CG → GCA
  - GCA → ATG
  - ATG → TG

**Note:** Even degree nodes (except start+finish)
real world

model: ‘abstraction’

is this what we want?
(can we handle errors?)

algorithm
NP complete: heuristics
characterization
how solutions relate
this slide intentionally left blank