Introduction to
Sequence Analysis

Chaochun Wei
Fall 2013
Contents

1. Overview
2. Complexity analysis
3. Sequence alignment
4. Sequence mapping
5. sequence assembly
6. Gene prediction
Research areas for sequence analysis

NGS applications

- Total Reads
- Rm Adaptor
- Rm redundancy
- Assembly/freq
- DNA sequencing
- RNA sequencing

Gene Prediction
- SNV Detection
- CNV Detection

Comparison analysis
- Exon
- Intron
- UTR
- Intergenic

Compare to non-coding RNA db
- Rfam
- Mibase
- ncRNA DB

Prediction software
- tRNA: tRNAscan
- rRNA: RNAmmer
- snoRNA: snoGPS, snoSeekert
- tmRNA: ARAGORN, BRUCE
- scaRNA: SCARNA, miRNA
Regulatory network analysis pipeline

- Expression Data
- Direct evidence
- Indirect evidence
- Coexpressed genes
- Data-Driven
- Knowledge-Driven
- TFBS
- Regulatory Network
- Hypothesis and Validation
- Genome-Wide location analysis
- Functional Category
- Text Mining
- Metabolic networks & signaling
- Other
- TRANSFAC
- Promoter
Basic problems in sequence analysis

1. Sequence alignment
2. Genome assembly
3. Functional element finding
4. Regulatory network reconstruction
Contents

1. Overview
2. Complexity analysis
3. Sequence alignment
4. Sequence mapping
5. sequence assembly
6. Gene prediction
Algorithm Complexity Analysis

Chaochun Wei
Fall 2013
Contents

• Reading materials
• Why do we need to analyze the complexity of an algorithm?
  • Examples
• Introduction
  – Algorithm complexity
  – “Big O” notation: O( )
Reading

Cormen book:
Thomas, H. , Cormen, Charles, E., Leiserson, and Ronald, L., Rivest .
Introduction to Algorithms, The MIT Press.

(read Chapter 1 and 2, page 1-44).
A real example: Exon-capture data analysis

There are ~60 millions of short reads sequenced from exon regions of a human genome. We need to figure out the how many exons were covered with at least 10 reads.

Steps:
1. Reads are aligned to the genome;
2. Each alignment is checked to see the exon it covers;
3. For each exon, check the number of reads cover the exon;
4. For all exons, filter out those with read number < 10.
A real example: Exon-capture data analysis
A real example: Exon-capture data analysis

1 days later

Student: I have created a program to do the analysis. It’s running.
Teacher: Cool. Let me know when your analysis finishes.
A real example: Exon-capture data analysis

6 days later...

Student: My program has been running for 5 days, and it keeps on running. I have no idea about what is happening and what to do with it.

Teacher: Its core is a sorting algorithm with a complexity of at most $O(N \times \lg N)$. It should be done within a few minutes!

Student: What?.....
Algorithm and its complexity

An **algorithm** is any well-defined computational procedure that takes in some **inputs** and produces some **outputs**.

Example: Sort an array of numbers

3, 2, 4, 5, 7, 1, 6 → 1, 2, 3, 4, 5, 6, 7
Algorithm and its complexity

An algorithm is any well-defined computational procedure that takes in some inputs and produces some outputs.

**Complexity:** a function of **input size**
- Time complexity: the running time
- Space complexity: the memory size required
Algorithm and its complexity

Input size

- Number of items in the input
  - Sorting problem
  - FFT
- Total number of bits needed to represent the input
  - Arithmetic operation (+,-,x,/) 
- The value of input
  - Factorial (N!)

Multiple input sizes

- Need to specify which input size is used
  - Graph operation (number of Vertices, and edges)
Algorithm and its complexity

Before we start

- we use a generic one-processor, random-access machine.

No parallel
Algorithm and its complexity

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 → 1, 2, 3, 4, 5, 6

Insertion sort (A)
for j = 2 to length(A)
do key = A[j]
    /*insert A[j] into the sorted sequence A[1...j-1]*/
i=j-1
    while i>0 and A[i]>key
do A[i+1]=A[i];
i=i-1;
A[i+1]=key;
Algorithm and its complexity

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 → 1, 2, 3, 4, 5, 6
Algorithm and its complexity

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 \rightarrow 1, 2, 3, 4, 5, 6

Insertion sort (A)

for j = 2 to length(A)
do key = A[j]
/*insert A[j] into the sorted sequence A[1...j-1]
i=j-1
while i>0 and A[i]>key
do A[i+1]=A[i];
i=i-1;
A[i+1]=key;

Algorithm time complexity: O(N^2)
Worst-case and average-case analysis

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 \to 1, 2, 3, 4, 5, 6

Insertion sort (A)
for j = 2 to length(A)
do key = A[j]
  /*insert A[j] into the sorted sequence A[1...j-1]
  i=j-1
  while i>0 and A[i]>key
    do A[i+1]=A[i];
    i=i-1;
  A[i+1]=key;

Algorithm time complexity: O(N^2)
Order of growth

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 → 1, 2, 3, 4, 5, 6

Insertion sort:
Algorithm run time complexity: $O(N^2)$
Order of growth: 2
O-notation (big-O notation): Asymptotic upper bound

$$O(g(n)) = \{ f(n): \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \leq f(n) \leq c \cdot g(n) \text{ for all } n \geq n_0 \}$$

Note about O-notation operations:
$$O(k_1 \cdot N^2 + k_2 \cdot N^3) = O(N^3) \text{ for constants } k_1, k_2$$
O-notation (big-O notation): Asymptotic upper bound

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 → 1, 2, 3, 4, 5, 6

Insertion sort:
algorithm time complexity: O(N^2)
Sorting with time complexity of O(N*logN)

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 → 1, 2, 3, 4, 5, 6

Sort (A)
for j = 2 to length(A)
    do key = A[j]
        /*Use binary search to insert A[j]
        /*into the sorted sequence A[1…j-1]
        i=j-1
        Binary_search(A[j], A[1…j-1],)
Sorting

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 → 1, 2, 3, 4, 5, 6

There are a lot of sorting algorithms:
- Heap sort (O(N*logN))
- Merge sort (O(N*logN))
- *Quick sort (worst-case O(N^2), average O(N*logN))
Merge sort

Merge-Sort (A, p, r)
    if p<r
        then q=[(p+r)/2]
        Merge-Sort(A, p, q)
        Merge-Sort(A,q+1,r)
        Merge(A, p, q, r)

Time Complexity: \[ T(N) = \begin{cases} 
    O(1); & \text{if } N = 1 \\
    2T(N/2) + O(N); & \text{if } N > 1 
\end{cases} \]

Solve it: T(N) = O(N*\log N)
Space complexity

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 → 1, 2, 3, 4, 5, 6

Need an array of size N: A[1…N], and 3 temporary variables
O(N)

Example: Sequence alignment

Need a two-dimension array of size N*M, and a constant number of temporary variables
O(N*M) or O(max(N, M))
Other issues

- **Output size**
  - **Blast**: output can be a problem
- **Input/Output method/place/mode**
  - **Speed**
    - screen $< <$ hard disk $< <$ memory
- **Programming language**
  - **Speed**
    - Perl $< <$ java $< <$ C++ $< <$ C
Contents

1. Overview
2. Complexity analysis
3. Sequence alignment
4. Sequence mapping
5. sequence assembly
6. Gene prediction
The simple but powerful dot plot

A DNA dot plot of a human zinc finger transcription factor (GenBank ID NM_002383), showing regional self-similarity.
Sequence comparison algorithms

• Simple identity (as in C’s strcmp())
• Hashing
• Longest common substring
### Longest common substring

<table>
<thead>
<tr>
<th>$\Delta$</th>
<th>C</th>
<th>A</th>
<th>G</th>
<th>C</th>
<th>C</th>
<th>U</th>
<th>C</th>
<th>G</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>A</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>A</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>U</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
<td>1.3</td>
<td>0.3</td>
<td>1.0</td>
<td>0.3</td>
<td>2.0</td>
<td>0.7</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>1.0</td>
<td>0.3</td>
<td>0.0</td>
<td>3.0</td>
<td>1.7</td>
<td>1.3</td>
<td>1.0</td>
<td>0.7</td>
<td>1.0</td>
<td>1.3</td>
<td>1.3</td>
<td>0.0</td>
</tr>
<tr>
<td>A</td>
<td>0.0</td>
<td>2.0</td>
<td>0.7</td>
<td>0.3</td>
<td>1.7</td>
<td>2.7</td>
<td>1.3</td>
<td>1.0</td>
<td>0.7</td>
<td>1.7</td>
<td>2.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>A</td>
<td>0.0</td>
<td>0.7</td>
<td>1.7</td>
<td>0.3</td>
<td>1.3</td>
<td>2.7</td>
<td>2.3</td>
<td>1.0</td>
<td>0.7</td>
<td>1.7</td>
<td>2.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>U</td>
<td>0.0</td>
<td>0.0</td>
<td>0.3</td>
<td>1.3</td>
<td>1.0</td>
<td>2.3</td>
<td>2.3</td>
<td>2.0</td>
<td>0.7</td>
<td>1.7</td>
<td>2.7</td>
<td>1.7</td>
<td>1.0</td>
</tr>
<tr>
<td>U</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.3</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>2.0</td>
<td>3.3</td>
<td>2.0</td>
<td>1.7</td>
<td>1.3</td>
<td>2.7</td>
</tr>
<tr>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.3</td>
<td>0.7</td>
<td>0.7</td>
<td>2.0</td>
<td>3.0</td>
<td>1.7</td>
<td>1.3</td>
<td>2.3</td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.7</td>
<td>1.0</td>
<td>2.0</td>
<td>0.7</td>
<td>1.7</td>
<td>1.7</td>
<td>3.0</td>
<td>2.7</td>
<td>1.3</td>
<td>1.0</td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>1.0</td>
<td>0.3</td>
<td>0.7</td>
<td>1.7</td>
<td>0.3</td>
<td>2.7</td>
<td>1.7</td>
<td>2.7</td>
<td>2.3</td>
<td>1.0</td>
</tr>
<tr>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.7</td>
<td>0.7</td>
<td>0.3</td>
<td>0.3</td>
<td>1.3</td>
<td>1.3</td>
<td>2.3</td>
<td>1.3</td>
<td>2.3</td>
<td>2.0</td>
</tr>
<tr>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.7</td>
<td>0.7</td>
<td>0.3</td>
<td>0.3</td>
<td>1.3</td>
<td>1.3</td>
<td>2.3</td>
<td>1.3</td>
<td>2.3</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Fig. 1. $H_{xy}$ matrix generated from the application of eqn (1) to the sequences A-A-U-G-C-C-A-U-U-G-A-C-G-G and C-A-G-C-C-U-C-G-C-U-U-A-G. The underlined elements indicate the trackback path from the maximal element 3.30.
Analysis of algorithms and big-O notation

Measure the Complexity of an algorithm: $O()$

- `strcmp`: $O(n)$
- `longest common substring`: $O(nm)$
Pattern matching algorithms

- Brute force
- Knuth/Morris/Pratt: a finite state automata solution
- Regular expressions and nondeterministic finite state automata
Dynamic programming sequence alignment algorithms

- Needleman/Wunsch global alignment
- Smith/Waterman local alignment
- Linear and affine gap penalties
Needleman/Wunsch global alignment (1970)

- Two sequences \( X = x_1 \ldots x_n \) and \( Y = y_1 \ldots y_m \)
- Let \( F(i, j) \) be the optimal alignment score of \( X_{1 \ldots i} \) of \( X \) up to \( x_i \) and \( Y_{1 \ldots j} \) of \( Y \) up to \( Y_j \) \((0 \leq i \leq n, 0 \leq j \leq m)\), then we have

\[
F(0,0) = 0
\]

\[
F(i, j) = \max \left\{ \begin{array}{l}
F(i-1, j-1) + s(x_i, y_j) \\
F(i-1, j) - d \\
F(i, j-1) - d
\end{array} \right. 
\]
Needleman/Wunsch global alignment (1970)

\[
F(i-1, j-1) \quad F(i, j-1)
\]

\[
S(x_i, y_j) \quad -d
\]

\[
F(i-1, j) \quad F(i, j)
\]

\[
-d
\]

\[
F(0,0) = 0
\]

\[
F(i, j) = \max \left\{ \begin{array}{l}
F(i-1, j-1) + s(x_i, y_j) \\
F(i-1, j) - d \\
F(i, j-1) - d
\end{array} \right. 
\]
Smith/Waterman local alignment (1981)

- Two sequences $X = x_1...x_n$ and $Y = y_1...y_m$
- Let $F(i, j)$ be the optimal alignment score of $X_1...i$ of $X$ up to $x_i$ and $Y_1...j$ of $Y$ up to $Y_j$ ($0 \leq i \leq n$, $0 \leq j \leq m$), then we have

$$F(0,0) = 0$$

$$F(i, j) = \max \begin{cases} 
0 \\
F(i - 1, j - 1) + s(x_i, y_j) \\
F(i - 1, j) - d \\
F(i, j - 1) - d
\end{cases}$$
Linear and affine gap penalties

- **Linear**: \( w(k) = k \cdot d \)
- **Affine**: \( w(k) = d + (k-1) \cdot e \)

Let \( M(i,j), I_x(i,j), I_y(i,j) \) be the best scores up to \((i,j)\):

- \( M(i,j) \): \( x_i \) is aligned to \( y_j \);
- \( I_x(i,j) \): \( x_i \) is aligned to a gap;
- \( I_y(i,j) \): \( y_j \) is aligned to a gap

then we have

\[
M(i, j) = \max \begin{cases} 
M(i-1, j-1) + s(x_i, y_j), \\
I_x(i-1, j-1) + s(x_i, y_j), \\
I_y(i-1, j-1) + s(x_i, y_j);
\end{cases}
\]

\[
I_x(i, j) = \max \begin{cases} 
M(i-1, j) - d, \\
I_x(i-1, j) - e;
\end{cases}
\]

\[
I_y(i, j) = \max \begin{cases} 
M(i, j-1) - d, \\
I_y(i, j-1) - e.
\end{cases}
\]
Reading materials

**Required**

   The Smith/Waterman algorithm

**Other recommended background:**

   The efficient form of the Needleman/Wunsch and Smith/Waterman algorithms.
   More advanced reading: a divide and conquer method to reduce the memory cost from O(n^2) to O(n)
sequence alignment (2)

Blast
Blast

• Contents
  – Reading materials
  – Introduction to BLAST
  – Inside BLAST
    • Algorithm
    • Karlin-Altschul Statistics
Karlin, S, and SF Altschul (1990), “Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes”, PNAS 87:2264-68

Supporting materials
Introduction to BLAST

- What is BLAST
  - Basic Local Alignment Search Tool
- Why BLAST
  - Quickly search a sequence database
Alignment in Real Life (20 years ago)

• One of the major uses of alignments is to find sequences in a database
• The current protein database contains about $10^8$ residues!
  – Searching a $10^3$ long target sequence requires to evaluate about $10^{11}$ matrix cells…
  – … which will take about **three hours** in the rate of $10^6$ evaluations per second.
  – Quite annoying when, say, $10^3$ sequences are waiting to be searched. About **four months** will be required for completing the analysis!
Introduction to BLAST

- Different versions of BLAST
  - NCBI-BLAST
  - WU-BLAST (now AB-BLAST)
Different BLAST programs: according to the query and database

<table>
<thead>
<tr>
<th>Program</th>
<th>Query</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>blastp</td>
<td>protein</td>
<td>protein</td>
</tr>
<tr>
<td>blastn</td>
<td>nucleotide</td>
<td>nucleotide</td>
</tr>
<tr>
<td>blastx</td>
<td>nucleotide</td>
<td>protein</td>
</tr>
<tr>
<td></td>
<td></td>
<td>protein</td>
</tr>
<tr>
<td>tblastn</td>
<td>protein</td>
<td>nucleotide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>protein</td>
</tr>
<tr>
<td>tblastx</td>
<td>nucleotide</td>
<td>nucleotide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>protein</td>
</tr>
</tbody>
</table>
Query= RU1A_HUMAN
(282 letters)

Database: /home/ccwei/courses/g_and_p/C.elegans/Proteome/ws_215.protein
24,705 sequences; 10,879,267 total letters.
Searching....10....20....30....40....50....60....70....80....90....100% done

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Accession</th>
<th>Description</th>
<th>Score</th>
<th>P(N)</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>K08D10.3</td>
<td>CE07355</td>
<td>WBGene00004386 locus:rnp-3 U1 small nucl...</td>
<td>378</td>
<td>3.2e-53</td>
<td>2</td>
</tr>
<tr>
<td>K08D10.4</td>
<td>CE28597</td>
<td>WBGene00004385 locus:rnp-2 U1 small nucl...</td>
<td>332</td>
<td>1.5e-51</td>
<td>2</td>
</tr>
<tr>
<td>C50D2.5</td>
<td>CE38492</td>
<td>WBGene00016808 status:Confirmed UniProt:Q...</td>
<td>113</td>
<td>7.4e-08</td>
<td>1</td>
</tr>
<tr>
<td>F46A9.6</td>
<td>CE08260</td>
<td>WBGene00003172 locus:mec-8 mecanosensory ...</td>
<td>111</td>
<td>5.8e-07</td>
<td>2</td>
</tr>
<tr>
<td>R09B3.2</td>
<td>CE16307</td>
<td>WBGene00011155 RNA recognition motif. (ak...</td>
<td>91</td>
<td>2.6e-05</td>
<td>1</td>
</tr>
<tr>
<td>D2089.4b</td>
<td>CE30509</td>
<td>WBGene00004207 locus:ptb-1 status:Partia...</td>
<td>86</td>
<td>5.4e-05</td>
<td>2</td>
</tr>
<tr>
<td>T01D1.2g</td>
<td>CE41586</td>
<td>WBGene00001340 locus:etr-1 status:Confir...</td>
<td>95</td>
<td>6.5e-05</td>
<td>2</td>
</tr>
<tr>
<td>T23F6.4</td>
<td>CE18963</td>
<td>WBGene00004315 locus:rbd-1 RNA recognitio...</td>
<td>85</td>
<td>8.1e-05</td>
<td>2</td>
</tr>
<tr>
<td>T01D1.2a</td>
<td>CE12942</td>
<td>WBGene00001340 locus:etr-1 RNA-binding p...</td>
<td>95</td>
<td>9.0e-05</td>
<td>2</td>
</tr>
</tbody>
</table>
>K08D10.3   CE07355 WBGene00004386  locus:rnp-3       U1 small nuclear ribonucleoprotein
A     status:Confirmed        UniProt:Q21323  protein_id:AAA98033.1
Length = 217

Score = 378 (138.1 bits), Expect = 3.2e-53, Sum P(2) = 3.2e-53
Identities = 69/116 (59%), Positives = 89/116 (76%)

Query:      5 ETRPNHTIYINNLNEKIKKDELKKSLYAIFSQFGQILDILVSRSLKMNRGQAFVIFKEVSS 64
  +               +               +               ++         + R KMRGQA ++FKEVSS
Sbjct:      3 DINPNHTIYVNNLNEKVKKDELKRSKLHMYFTQFGEIIQILMSFRKEKMNRQAHIVFKEVSS 62

Query:     65 ATNALRSMQGFPYDMPKMRQYAYKTDSDIIAKMKGTFVX X X X X X X X X X X XS QETPA 120
  A+NALR+++QGFPY KPMRIQYA+ DSD+I++ KGTFV E PA
Sbjct:     63 ASNALRALQGFPYGYKPMRIQYAREDSVISRAGTFVEKQRQKSTKIAKPYEKP 118

Score = 179 (68.1 bits), Expect = 3.2e-53, Sum P(2) = 3.2e-53
Identities = 33/77 (42%), Positives = 49/77 (63%)

Query:    206 PNHILFLTNLPEETNEMLMLSMFnQFPFGKEVRLVPGRHDIAFVEFDNEVQAGAAARDALQ 265
  PN+ILF +N+PET + +F+QFPG +EVR +P D AF+E+++E + AR AL
Sbjct:    141 PNNILFCSNIPEQTEPQITIFSQFPGLREVRWMPNTKDFAFIEYSEDLSEPARQALD 200

Query:    266 GFKITQNNAMKISFAKK 282
  F+IT     + + FA K
Sbjct:    201 NFRITPTQQITVKFASK 217
Heuristic Search

• Rather than struggling to find the optimal alignment we may save a lot of time by employing heuristic algorithms
  – Execution time is much faster
  – May completely miss the optimal alignment

• Two important algorithms
  – BLAST
  – FASTA
Basic Intuition 1: Seeds

• **Observation:** Real-life matches often contain long strings with gap-less matches

• **Action:** Try to find significant gap-less matches and then extend them
**Observation:** If the optimal alignment of $s$ and $t$ has few gaps, then path of the alignment will be close to diagonal.

**Action:** To find such a path, it suffices to search in a diagonal band of the matrix.

- If the diagonal band consists of $k$ diagonals (width $k$), then dynamic programming takes $O(kn)$.
- Much faster than $O(n^2)$ of standard DP.

Basic Intuition 2: Banded DP
Banded DP for Local Alignment

• **Problem:** The banded diagonal needs not be the main diagonal when looking for a good local alignment
  – Also the case when the lengths of $s$ and $t$ are different

• **Solution:** Heuristically find potential diagonals and evaluate them using Banded DP
Basic Local Alignment Search Tool (BLAST)

- **Publications:**
  - **Ungapped BLAST** – Altschul et al., 1990
  - **Gapped BLAST, PSI-BLAST** - Altschul et al., 1997

- **Input:**
  - **Query (target) sequence** – either DNA, RNA or Protein
  - **Scoring Scheme** – gap penalties, substitution matrix for proteins, identity/mismatch scores for DNA/RNA
  - **Word length** $w$ – typical is $w=3$ for proteins and $w=11$ for DNA/RNA

- **Output:**
  - Statistically significant matches
Running BLAST
NCBI BLAST – result summary

**Query ID**: T00048, human
**Database Name**: nr, Non-redundant GenBank CDS translations+PDB+SwissProt+PIR+PRF excluding environmental samples from WGS projects

**Putative conserved domains** have been detected, click on the image below for detailed results.

Distribution of 608 Blast Hits on the Query Sequence

*Color key for alignment scores*

- **Query**
- **<40**
- **40-50**
- **50-60**
- **60-70**
- **70-80**
- **80-200**
- **>200**

Mouse-over to show define and scores, click to show alignments.
NCBI BLAST use – predict function
NCBI BLAST use – infer evolutionary tree

This tree was produced using BLAST pairwise alignments.
NCBI BLAST use – construct families

Local query sequence

Conserved domains on [i=1][2][7][2][3][8]

Graphical summary

List of domain hits

Blast search parameters

Options: Database: CDD  Low complexity filter: yes  E-value threshold: 0.01  Max. hits: 100
Data Source: Live blast search  RID = 9W9J3SV601N
System: Search creator: newblast  Software: blastp 2.2.20+  Service: rpsblast

References:
NCBI BLAST batch jobs

Batch BLAST jobs

(1) input "batches" of sequences into one form and retrieve the results

Select a BLAST search page form the main BLAST home page. Next you can either cut and paste multiple FASTA sequences from a text file into the main input box.

Or alternatively, you can use the browse button to import a local file from your computer.
Stand-alone BLAST

(1) NCBI standalone BLAST

(2) The WU-BLAST
BLAST use – command line

BLAST use

(1) Make a formatted database to use

execute command: `formatdb (xdformat for WU-BLAST)`

input: fasta format sequences (database sequences)

output: formatted database, used by BLAST program
xdformat: create a WU-BLAST database

Purpose: produce databases for BLAST in XDF (eXtended Database Format) from one or more input files in FASTA format; or report XDF databases to standard output in FASTA format.

Create a database:
   xdformat [-p|n] [options] fadb
   xdformat [-p|n] -o xdbname [options] fadb...

Append sequences to an existing database:
   xdformat [-p|n] -a xdbname [options] fadb...

Report the contents of existing database(s) to stdout in FASTA format:
   xdformat [-p|n] -r [options] xdbname...

Describe the contents of existing database(s):
   xdformat [-p|n] -i xdbname...

Verify the integrity of existing database(s):
   xdformat [-p|n] -V xdbname...
BLAST use – command line

BLAST use

Carry out BLAST program

execute command: blastn, blastp

input: fasta sequences (query sequences), database, parameters

output: resulted alignment file
Blastn parameters

BLASTN 3.0PE-AB [2009-10-30] [linux26-x64-I32LPF64 2009-11-17T18:52:53]

Copyright (C) 2009 Warren R. Gish. All rights reserved. Unlicensed use, reproduction or distribution are prohibited. Advanced Biocomputing, LLC, licenses this software only for personal use on a personally owned computer.


Notice: this program and its default parameter settings are optimized to find nearly identical sequences rapidly. To identify weak protein similarities encoded in nucleic acid, use BLASTX, TBLASTN or TBLASTX.

Usage:

    BLASTN database queryfile [options]


-matrix <matrix-name> use the specified scoring matrix (default matrix is computed from M=+5 N=-4); be sure to consider changing the default gap penalties when using a non-default scoring system
Blastn parameters

- Q <s> penalty score for a gap of length 1
- R <s> penalty score for extending a gap by each letter after the first
- top search only the top strand of the query
- bottom search only the bottom strand of the query
- mformat <n>[, outfile] specify alternate output format(s) (default 1)
- msgstyle <n> specify alternate informatory message style (default 0)
- filter <method> hard mask the query using the specified method (e.g., "seg", "xnu", "ccp", "dust" or "none")
- lcfilter hard mask lower case letters in the query sequence
- lcmask soft mask lower case letters in the query sequence
- topcomboN <n> report this number of consistent (colinear) groups of HSPs
PART II  inside into BLAST (omitted)
Contents

1. Overview
2. Complexity analysis
3. Sequence alignment
4. Sequence mapping
5. Sequence assembly
6. Gene prediction
Sequence alignment or mapping

Chaochun Wei
Fall 2013
Sequence mapping

• Contents
  – Reading materials
  – Sequence alignment
    • BLAT
  – Latest progress of sequence alignment (Sequence mapping)
    • Bowtie
    • SOAP
    • Tophat
Reading

- Blat FAQ: [http://genome.ucsc.edu/FAQ/FAQblat.html](http://genome.ucsc.edu/FAQ/FAQblat.html)


BLAT: Blast-Like Alignment Tool

- Not BLAST
- Indexed on database (BLAST indexed on the query)
  - Need ~1G memory for human genome
- Need some extra time for database initialization (index)
- Can be 500 times faster than BLAST
- Can display results in the UCSC genome browser

Blat FAQ: http://genome.ucsc.edu/FAQ/FAQblat.html
BLAT

• Designed to quickly find
  – DNA sequences of 95% and greater similarity of length 25 bases or more.
  – Protein sequences of 80% and greater similarity of length 20 amino acids or more.

• In practice
  – DNA BLAT works well on primates, and
  – protein blat on land vertebrates
# BLAT—The BLAST-Like Alignment Tool

Timing of BLAT vs. WU-TBLASTX on a Data Set of 1000 Mouse Reads against a RepeatMasked Human Chromosome 22

<table>
<thead>
<tr>
<th>Method</th>
<th>K</th>
<th>N</th>
<th>Matrix</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>WU-TBLASTX</td>
<td>5</td>
<td>1</td>
<td>+15/−12</td>
<td>2736 s</td>
</tr>
<tr>
<td>WU-TBLASTX</td>
<td>5</td>
<td>1</td>
<td>BLOSUM62</td>
<td>2714 s</td>
</tr>
<tr>
<td>BLAT</td>
<td>5</td>
<td>1</td>
<td>+2/−1</td>
<td>61 s</td>
</tr>
<tr>
<td>BLAT</td>
<td>4</td>
<td>2</td>
<td>+2/−1</td>
<td>37 s</td>
</tr>
</tbody>
</table>

**K**: the size of the perfectly matching as a seed for an alignment  
**N**: the number of hits in a gapless 100-aa window required to trigger a detailed alignment.

**Matrix**: column describes the match/mismatch scores or the substitution score matrix used.
## Comparison of NGSs vs. traditional technology

<table>
<thead>
<tr>
<th></th>
<th>Sanger</th>
<th>454</th>
<th>Solexa</th>
<th>SOLiD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Length (bps)</td>
<td>650–1100</td>
<td>150–250</td>
<td>35–150</td>
<td>25–50</td>
</tr>
<tr>
<td>Capacity (reads/run)</td>
<td>96</td>
<td>400,000</td>
<td>200,000,000</td>
<td>2,000,000,000</td>
</tr>
<tr>
<td>Error Rate</td>
<td>10^−3</td>
<td>&lt;10^−2</td>
<td>~10^−2</td>
<td>~10^−2</td>
</tr>
<tr>
<td>Cost ($/Mbp)</td>
<td>5000</td>
<td>~5</td>
<td>~0.6</td>
<td>~0.2</td>
</tr>
<tr>
<td>Time/run</td>
<td>~3h</td>
<td>~7h</td>
<td>2–10d</td>
<td>3–14d</td>
</tr>
<tr>
<td>Throughput</td>
<td>100Kb</td>
<td>~1Gb</td>
<td>~600Gb</td>
<td>100–300Gb</td>
</tr>
</tbody>
</table>
http://genome.ucsc.edu/cgi-bin/hgBlat?command=start

BLAT Search Genome

Genome: Cow  Assembly: Oct. 2007 (Baylor 4.0/bosTau4)  Query type: BLAT's guess  Sort output: query.score  Output type: psl

Paste in a query sequence to find its location in the genome. Multiple sequences may be searched if separated by lines starting with '>' followed by the sequence name.

File Upload: Rather than pasting a sequence, you can choose to upload a text file containing the sequence.

Only DNA sequences of 25,000 or fewer bases and protein or translated sequence of 10000 or fewer letters will be processed. Up to 25 sequences can be submitted at the same time. The total limit for multiple sequence submissions is 50,000 bases or 25,000 letters.

For locating PCR primers, use In-Silico PCR for best results instead of BLAT.
How to map billions of short reads onto genomes

Cole Trapnell & Steven L Salzberg

Mapping the vast quantities of short sequence fragments produced by next-generation sequencing platforms is a challenge. What programs are available and how do they work?

A new generation of DNA sequencers that can rapidly and inexpensively sequence billions of bases is transforming genomic science. These new machines are quickly becoming the technology of choice for whole-genome sequencing and for a variety of sequencing-based assays, including gene expression, DNA-protein interaction, human resequencing and RNA splicing studies\(^1\)--\(^3\). For example, the RNA-Seq protocol, in which processed mRNA is converted to cDNA and sequenced, is enabling the identification of previously unknown genes and alternative splice variants; the ChIP-Seq approach, which sequences immunoprecipitated DNA fragments bound to proteins, is revealing networks of interactions between transcription factors and DNA regulatory elements\(^4\); and the whole-genome sequencing of tumor cells is uncovering previously unidentified cancer-related genes.

In this case, to make sense of the reads, their positions within the reference sequence must be determined. This process is known as aligning or ‘mapping’ the read to the reference. In one version of the mapping problem, reads must be aligned without allowing large gaps in the reference.

Challenges of mapping short reads

<table>
<thead>
<tr>
<th>Program</th>
<th>Website</th>
<th>Open source?</th>
<th>Handles ABI color space?</th>
<th>Maximum read length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bowtie</td>
<td><a href="http://bowtie.cbc.b.umd.edu">http://bowtie.cbc.b.umd.edu</a></td>
<td>Yes</td>
<td>No</td>
<td>None</td>
</tr>
<tr>
<td>BWA</td>
<td><a href="http://maq.sourceforge.net/bwa-man.shtml">http://maq.sourceforge.net/bwa-man.shtml</a></td>
<td>Yes</td>
<td>Yes</td>
<td>None</td>
</tr>
<tr>
<td>Maq</td>
<td><a href="http://maq.sourceforge.net">http://maq.sourceforge.net</a></td>
<td>Yes</td>
<td>Yes</td>
<td>127</td>
</tr>
<tr>
<td>Mosaik</td>
<td><a href="http://bioinformatics.bc.edu/marthlab/Mosaik">http://bioinformatics.bc.edu/marthlab/Mosaik</a></td>
<td>No</td>
<td>Yes</td>
<td>None</td>
</tr>
<tr>
<td>Novoalign</td>
<td><a href="http://www.novocraft.com">http://www.novocraft.com</a></td>
<td>No</td>
<td>No</td>
<td>None</td>
</tr>
<tr>
<td>SOAP2</td>
<td><a href="http://soap.genomics.org.cn">http://soap.genomics.org.cn</a></td>
<td>No</td>
<td>No</td>
<td>60</td>
</tr>
<tr>
<td>ZOOM</td>
<td><a href="http://www.bioinfor.com">http://www.bioinfor.com</a></td>
<td>No</td>
<td>Yes</td>
<td>240</td>
</tr>
</tbody>
</table>
Latest progress of sequence alignment/mapping

• Aligning (mapping) billions of short reads
  – Bowtie
  – SOAP
  – BWA
  – Tophat
Algorithms (a) based on spaced-seed indexing; (b) based on Burrows-Wheeler transform
Bowtie (Burrows-Wheeler transform)

(a) The Burrows-Wheeler matrix and transformation for 'acaacg'.

(b) Steps taken by EXACTMATCH to identify the range of rows, and thus the set of reference suffixes, prefixed by 'aac'.

(c) UNPERMUTE repeatedly applies the last first (LF) mapping to recover the original text (in red on the top line) from the Burrows-Wheeler transform (in black in the rightmost column).

## Bowtie versus SOAP v1.10 and Maq v0.6.6

<table>
<thead>
<tr>
<th>Platform</th>
<th>CPU time</th>
<th>Wall clock time</th>
<th>Reads mapped per hour (millions)</th>
<th>Peak virtual memory footprint (megabytes)</th>
<th>Bowtie speed-up</th>
<th>Reads aligned (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bowtie -v 2 Server</td>
<td>15 m 7 s</td>
<td>15 m 41 s</td>
<td>33.8</td>
<td>1,149</td>
<td>-</td>
<td>67.4</td>
</tr>
<tr>
<td>SOAP</td>
<td>91 h 57 m 35 s</td>
<td>91 h 47 m 46 s</td>
<td>0.10</td>
<td>13,619</td>
<td>351×</td>
<td>67.3</td>
</tr>
<tr>
<td>Bowtie PC</td>
<td>16 m 41 s</td>
<td>17 m 57 s</td>
<td>29.5</td>
<td>1,353</td>
<td>-</td>
<td>71.9</td>
</tr>
<tr>
<td>Maq</td>
<td>17 h 46 m 35 s</td>
<td>17 h 53 m 7 s</td>
<td>0.49</td>
<td>804</td>
<td>59.8×</td>
<td>74.7</td>
</tr>
<tr>
<td>Bowtie Server</td>
<td>17 m 58 s</td>
<td>18 m 26 s</td>
<td>28.8</td>
<td>1,353</td>
<td>-</td>
<td>71.9</td>
</tr>
<tr>
<td>Maq</td>
<td>32 h 56 m 53 s</td>
<td>32 h 58 m 39 s</td>
<td>0.27</td>
<td>804</td>
<td>107×</td>
<td>74.7</td>
</tr>
</tbody>
</table>

The performance and sensitivity when aligning 8.84 M reads from the 1,000 Genome project (NCBI Short Read Archive: SRR001115) trimmed to 35 base pairs.

## Comparison of short-read mapping methods

<table>
<thead>
<tr>
<th>Software</th>
<th>database size</th>
<th>Index time</th>
<th>Index size</th>
<th>Reads</th>
<th>time</th>
<th>Result size</th>
<th>Mapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAQ</td>
<td>70M</td>
<td>--</td>
<td>34.3M</td>
<td>20,000</td>
<td>112s</td>
<td>270.8K</td>
<td>19,576</td>
</tr>
<tr>
<td>SOAP</td>
<td>70M</td>
<td>91.46s</td>
<td>792.8M</td>
<td>20,000</td>
<td>0.46s</td>
<td>1.9M</td>
<td>12,564/12,954</td>
</tr>
<tr>
<td>SOAP</td>
<td>70M</td>
<td>91.46s</td>
<td>792.8M</td>
<td>3,251,337</td>
<td>83.97s</td>
<td>316.1M</td>
<td>2,055,104/2,120,025</td>
</tr>
<tr>
<td>bowtie</td>
<td>70M</td>
<td>220s</td>
<td>81.9M</td>
<td>20,000</td>
<td>&lt;1s</td>
<td>2.7M</td>
<td>18,958/19,573</td>
</tr>
<tr>
<td>bowtie</td>
<td>70M</td>
<td>220s</td>
<td>81.9M</td>
<td>3,251,337</td>
<td>77.2s</td>
<td>431.7M</td>
<td>3,086,705/3,184,978</td>
</tr>
</tbody>
</table>

Reads: simulated by MetaSim version 0.9.1
Index: converted from virus sequences from NCBI
RNA-seq data analysis


Figure 2  RNA-Seq assays produce short reads sequenced from processed mRNAs. Aligning these reads to the genome with Bowtie or Maq will produce the alignments shown in black but will fail to align the blue reads. A spliced-read mapper such as TopHat or ERANGE will also report the (blue) alignments spanning intron boundaries.
Contents

1. Overview
2. Complexity analysis
3. Sequence alignment
4. Sequence mapping
5. sequence assembly
6. Gene prediction
Popular assembly software

- Phrap
  - Memory intensive
  - Use quality values

http://www.phrap.org/
Assembly software for NGS data

- ABySS
- ALLPATHS
- SOAPdenovo
- VCAKE
- Velvet
- …
Contents

1. Overview
2. Complexity analysis
3. Sequence alignment
4. Sequence mapping
5. sequence assembly
6. Gene prediction
Gene prediction

Contents

1. Reading materials
2. Introduction

3. Gene prediction method review
   • Gene prediction methods
   • HMM
   • TWINSAN and N-SCAN
   • Using ESTs for gene prediction
   • Latest progress

4. Gene prediction FAQ
Reading materials


Experimental Validation of Predictions

- Tenney, A. E. et al., “Gene prediction and verification in a compact genome with numerous small introns”, *Genome Research*, 2004
Reading materials

Latest Progress in Gene Prediction

• Conrad: gene prediction using conditional random fields. Decaprio et al., *Genome Res.* 2007 Sep;17(9):1389-98.
  – Not working for vertebrate genomes
  – SVM for splice site
• CONTRAST: Gross et al., *Genome Biology* 2007, 8:R269
  • Best de novo gene predictor for human (gene level accuracy ~50%)
  • Used SVM and conditional random field
1. Introduction: DNA

- DNA contains genetic information.
- DNA can be expressed as a sequence of letters A, C, G and T.
  
  Eg: ACGTTTCGAGGT
1. Introduction

DNA $\rightarrow$ RNA $\rightarrow$ Protein

DNA

Transcription & processing

RNA

Translation

Protein
1. Introduction

RNA Processing

Primary mRNA 5’

\[
\begin{align*}
\text{UTR} & \quad (A)_n \\
(3' \text{ poly}(A) \text{ tail}) & \quad (A)_n \\
\beta\text{-Globin mRNA} & \\
1 & 31 & 105 & 147 (A)_n
\end{align*}
\]

Splicing
A gene is a highly structured region of DNA, it is a functional unit of inheritance.
Patterns in Splice Sites

Donor Sites

Acceptor Sites

Sequence data from RefSeq of human, mouse, rat and chicken.
1. Introduction

A Typical Human Gene Structure
1. Introduction

Genes in a Genome
In a Mammalian Genome

• Finding all the genes is hard
  – Mammalian genomes are large
    • 8,000 km of 10pt type
  – Only about 1% protein coding
• EST:
  • Short (~650bps)
  • High error rate (~1-5%)
  • Contains only UTRs or coding regions
1. Introduction

The Challenge and Opportunity

• ~3000 genomes
  – 222 animals
  – 93 plants

Numbers from
# 2. Gene Prediction Methods

<table>
<thead>
<tr>
<th>Generation</th>
<th>Date</th>
<th>Feature</th>
<th>Systems</th>
<th>Information or Methods used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(^{st})</td>
<td>Early 1980s</td>
<td>Approximate ends of protein coding regions and non-coding regions</td>
<td>TestCode, Fickett 1982</td>
<td>splice sites&lt;br&gt;promoters&lt;br&gt;Codon usage bias&lt;br&gt;Neuro-network methods</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GRAIL, Uberbacher and Mural 1991</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>⮖splice sites&lt;br&gt;promoters&lt;br&gt;Codon usage bias&lt;br&gt;Neuro-network methods</td>
<td></td>
</tr>
<tr>
<td>2(^{nd})</td>
<td>Early 1990s</td>
<td>A complete single gene in a short sequence</td>
<td>GeneID, Guigo et al. 1992</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GeneParser, Snyder and Stormo 1993</td>
<td>Translation start sites&lt;br&gt;Stop sites&lt;br&gt;Method: HMM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FGENSH, Solovyev et al. 1994</td>
<td></td>
</tr>
<tr>
<td>3(^{rd})</td>
<td>Mid-1990s</td>
<td>Multiple complete or partial genes in a long sequence</td>
<td>Genscan, Burge and Karlin, 1997</td>
<td>UTR&lt;br&gt;Method: Generalized HMM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4(^{th})</td>
<td>2000s</td>
<td>Complete gene structures in whole genomes</td>
<td>Twinscan, Korf, et al. 2001</td>
<td>Multiple-genomes&lt;br&gt;Transcript products&lt;br&gt;Method: Generalized HMM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>N-SCAN, Brown, et al. 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Twinscan_EST, N-SCAN_EST, Wei and Brent, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Gene Prediction Methods

Gene Prediction Methods (1)

• Categorization: by input information

1. Ab initio methods
   • Only need genomic sequences as input
     – GENSCAN (Burge 1997; Burge and Karlin 1997)
     – GeneFinder (Green, unpublished)
     – Fgenesh (Solovyev and Salamov 1997)
   • Can predict novel genes

2. Transcript-alignment-based methods
   • Use cDNA, mRNA or Protein similarity as major clues
     – ENSEMBL (Birney, Clamp, et. al. 2004)
   • Highly accurate
   • Can only find genes with transcript evidences
     – cDNA coverage 50-60%
     – + EST coverage up to 85%
Categorization: by input information

3. Hybrid Methods

• Integrate cDNA, mRNA, protein and EST alignments into ab initio methods
  – Genie (Kulp, Haussler et al. 1996)
  – Fgenesh+ (Solovyev and Salamov 1997)
  – Genomescan (Yeh, Lim et al. 2001)
  – GAZE (Howe, Chothiea et al. 2002)
  – AUGUSTUS+ (Stanke, Schoffmann et al. 2006)
2. Gene Prediction Methods

Gene Prediction Methods (3)

• Comparative-Genomics-Based Methods
  – TWINSCAN and N-SCAN
    • De novo
    • Assumption:
      – Coding regions are more conserved.
    • No transcript similarity information (like EST, cDNA, mRNA, or protein) is used
  – TWINSCAN-EST and N-SCAN_EST
    • Hybrid
    • Use EST to improve prediction accuracy
Conservation sequences represent the conservation patterns between two genomes.

No transcript similarity information (like EST, cDNA, mRNA, and protein) is used.
2. Gene Prediction Methods

Hidden Markov Model:
Model behind gene predictors

HMM for two biased coins flipping

\[
e_1(H) = 0.8, e_1(T) = 0.2, e_2(H) = 0.3, e_2(T) = 0.7
\]

\begin{align*}
\text{Observed sequence } x &= 11221111112222111111112222 \\
\text{Hidden state sequence } &= \pi
\end{align*}

\[
\pi^* = \arg\max_{\pi} P(x, \pi)
\]
2. Gene Prediction Methods

Most Probable Path and Viterbi Algorithm

Let \( f_l(i) = \max_{\{\pi_0, \ldots, \pi_{i-1}\}} \left( \Pr(x_0, \ldots, x_i, \pi_0, \ldots, \pi_{i-1}, \pi_i = l) \right) \)

Recursion (\(i=1\ldots L\))

\[
f_l(i) = e_l(x_i) \max_k \left( f_k(i-1) a_{kl} \right) ;
\]

\[
ptr_i(l) = \arg \max_k \left( f_k(i-1) a_{kl} \right).
\]

Time complexity \( O(N^2L) \)  
Space complexity \( O(NL) \)
2. Gene Prediction Methods

Probability of All the Possible Paths and Forward Algorithm

Let \( f_l(i) = \Pr(x_0, \ldots, x_i, \pi_i = l) \)

Recursion (\( i=1\ldots L \))

\[
f_l(i) = e_l(x_i) \sum_k (f_k(i-1) a_{kl})
\]

Probability of all the probable paths

\[
P(x) = \sum_{\pi} P(x, \pi) = \sum_k f_k(L)
\]
Posterior Probability and Forward and Backward Algorithm

Posterior Probability

\[ P(\pi_i = k \mid x) = \frac{P(\pi_i = k, x)}{P(x)} \]
2. Gene Prediction Methods

Posterior Probability and Forward and Backward Algorithm

Let

\[ b_k(L) = \Pr(x_{i+1}, \ldots, x_L \mid \pi_i = k) \]

Recursion \((i=L-1, \ldots, 1)\)

\[ b_k(i) = \sum_l a_{kl} e_l(x_{i+1})b_l(i+1) \]

Posterior Probability

\[ P(\pi_i = k \mid x) = \frac{P(\pi_i = k, x)}{P(x)} = \frac{f_k(i)b_k(i)}{P(x)} \]

\[ P(x) = \sum_{\pi} P(x, \pi) = \sum_l a_{0l} e_l(x_1)b_l(1) \]
2. Gene Prediction Methods

TWINSCAN Model

- Generalized HMM
- Each feature in a gene structure corresponds to one state.
- State-specific length models.
- State-specific sequence models
- Use Conservation information

Diagram of TWINSCAN Model with exons and other gene structure elements.

114
Conservation Sequence

Generated by projecting local alignments to the target sequence

human  CTAGAGATGCAAAAGAAACAGGTACCGCAGTC\-\-CCC
mouse  CTAGAG\-\-\-\-\-\-\-AGACAGGTACC\-\-\-\-\-\-AGGGCTCTCCT

- Pair each nucleotide of the target with
  - “|” if it is aligned and identical
  - “:” if it is aligned to mismatch
  - “.” if it is unaligned
N-SCAN: A Novel Gene Prediction System Using Multiple Genomes

- Uses Bayesian model to include phylogenetic tree information
- Predicts introns in 5’UTR
- Has Conserved non-coding regions

(Brown, Gross and Brent, *Genome Res.* 2005)
Using ESTs for Gene Prediction: TWINSCAN_EST

- Integrating EST alignment information into TWINSCAN to improve its accuracy where EST evidence exits and not to compromise its ability to predict novel genes.
Sequence Representation of EST Alignments

1. Use EST-to-genome alignment programs
   - BLAT (Kent 2002)

2. Project the top alignment for each EST to the target genomic sequence
Contents

1. Overview
2. Complexity analysis
3. Sequence alignment
4. Sequence mapping
5. sequence assembly
6. Gene prediction
2. Gene Prediction Methods

Accuracy Measurement

• Annotated data sets for training/testing

• Accuracy in different levels
  – Nucleotide level
  – Exon level
  – Gene level
  – Transcript level

• Sensitivity and specificity
Accuracy Measurement (continue)

2. Gene Prediction Methods

Annotation

Prediction

Correct Prediction

Sensitivity = \frac{Correct \ Prediction}{Total \ Annotation}

Specificity = \frac{Correct \ Prediction}{Total \ Prediction}

Gene Prediction

Methods
2. Gene Prediction Methods

TWINS\textsc{can\_est} and N-SCAN\textsc{\_est} on the Whole Human Genome

![Bar chart showing gene prediction methods]

- **exact gene sensitivity**
  - TWINSCAN\textsc{2.03}:
    - 24
  - TWINSCAN\textsc{\_est}:
    - 34
  - N-SCAN:
    - 38
  - N-SCAN\textsc{\_est}:
    - 44

- **exact gene specificity**
  - TWINSCAN\textsc{2.03}:
    - 14
  - TWINSCAN\textsc{\_est}:
    - 17
  - N-SCAN:
    - 22
  - N-SCAN\textsc{\_est}:
    - 23

- **exact exon sensitivity**
  - TWINSCAN\textsc{2.03}:
    - 67
  - TWINSCAN\textsc{\_est}:
    - 81
  - N-SCAN:
    - 85
  - N-SCAN\textsc{\_est}:
    - 88

- **exact exon specificity**
  - TWINSCAN\textsc{2.03}:
    - 56
  - TWINSCAN\textsc{\_est}:
    - 58
  - N-SCAN:
    - 59
  - N-SCAN\textsc{\_est}:
    - 60
2. Gene Prediction Methods

An Example of N-SCAN_EST Prediction

(Hg17, chr21:33,459,500-33,465,411)
2. Gene Prediction Methods

An Example of N-SCAN_EST Prediction
2. Gene Prediction Methods

Experimental Validation of Predictions

Siepel, *Genome Research, 2007*
Experimental Validation of Predictions

- See
  - The MGC Project Team, “The Completion of the Mammalian Gene Collection (MGC)”, Genome Research, 2009, 19:2324-2333
  - Tenney, A. E. et al., “Gene prediction and verification in a compact genome with numerous small introns”, Genome Research, 2004
2. Gene Prediction Methods

Latest Progress in Gene Prediction

New Methods

- **Conrad**: gene prediction using conditional random fields. Decaprio et al., *Genome Res.* 2007 Sep;17(9):1389-98.
  - Not working for vertebrate genomes
- **SVM** for splice site
- **CONTRAST**: Gross et al., *Genome Biology* 2007, 8:R269
  - Best *de novo* gene predictor for human (gene level accuracy ~50%)
  - Used SVM and conditional random field
3. Gene Prediction FAQs (from Dr. Ian Korf)

- **Algorithms vs. experts**
  - Q: are expert biologists better than computer programs?
  - A: Yes and no.

- **Next-generation sequencing**
  - Q: Will next-gen transcript sequencing replace gene prediction?
3. Gene Prediction FAQs

Automated High Throughput Sequencing

AGAAGCACCATCAACTAATCAAAATGCCTTTCAAAC
CAGCAGACAACCCAAAATGCCCCAAAATGCAGCAGAT
CCGTATACGCCGNAGAAGAAAAAGTAGCTGGAGGAT
ACAAATACCAAAATCCTGCTCCAAATACCACAAATCCTGCT
CAATAAAATGCTCGACTCCACCAACGTAACTGAACAC
GAAGCTGAATTGTACTGCAAAAATTGCCATGGACGTA
AATACGGACCTAAAGGATACGGATTCGGTGGTGGAG
CTGGGTGCTTAAGTATGGACGATGGAGCCCAATTCAA
GGGAACACAATAATTTTAAGAAGGAATCAATGTGAAG
ATGGCGGCCAAAACCACACCAACTGTCAGCGGTCGT
CAGTTCTACCCTTTTCCATCCCCATATACACTAATG
TAATATTTTTAGATCTTAAATTACAGACTTAGTTTTTATT
TATAAAATTTCGTATGACACGTTATAAATAAGAATTCGG
TTATTTGTATTAATTTGAATTAAATATTCTTTATTAAGA
CCAAAAAA
Next-generation sequencing technology

Sanger method:
huge lab, numerous machines and staffs

Next-generation:
one staff, one machine
3. Gene Prediction FAQs

Next-generation sequencing platforms

Applied Biosystems ABI 3730XL

Roche / 454 Genome Sequencer FLX

HeliScope™ Single Molecule Sequencer

Illumina / Solexa Genetic Analyzer

Applied Biosystems SOLiD
Comparison of NGS and traditional sequencing platform

<table>
<thead>
<tr>
<th>Platform</th>
<th>Sanger</th>
<th>454</th>
<th>Solexa</th>
<th>SOLiD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read length (bp)</td>
<td>650–1100</td>
<td>150–1000</td>
<td>35–150</td>
<td>25–50</td>
</tr>
<tr>
<td># of reads/run</td>
<td>96</td>
<td>400,000</td>
<td>2,000,000,000</td>
<td>2,000,000,000</td>
</tr>
<tr>
<td>Error rate</td>
<td>10^-3</td>
<td>&lt;10^-2</td>
<td>~10^-2</td>
<td>~10^-2</td>
</tr>
<tr>
<td>Cost ($/Mbp)</td>
<td>5000</td>
<td>~5</td>
<td>~0.2</td>
<td>~0.2</td>
</tr>
<tr>
<td>Time/run</td>
<td>~3 h</td>
<td>~7 hours</td>
<td>2–14 days</td>
<td>3–14 days</td>
</tr>
<tr>
<td>Throughput</td>
<td>100Kb</td>
<td>~1Gb</td>
<td>~600Gb</td>
<td>100–300Gb</td>
</tr>
</tbody>
</table>
Latest sequencing technologies

- Pacific Biosciences
  - Human genome: $100, 15 Minutes (2013)
- Complete Genomics
  - Finish 10,000 genomes in year 2010
- Ion Torrent
- Oxford Nanopore
- Visigen
- more…
Gene Prediction FAQs (continue)

• Algorithms vs. experts
  – Q: are expert biologists better than computer programs?
  – A: Yes and no.

• Next-generation sequencing
  – Q: Will next-gen transcript sequencing replace gene prediction?
  – A: No. Rare transcripts may require directed experiments to validate.
3. Gene Prediction FAQs

Gene Prediction FAQs (continue)

• Algorithms vs. experts
  – Q: are expert biologists better than computer programs?
  – A: Yes and no.

• Next-generation sequencing
  – Q: Will next-gen transcript sequencing replace gene prediction?
  – A: No. Rare transcripts may require directed experiments to validate.

• Prediction accuracy
  – Q: Why are gene prediction programs inaccurate?
  – A: We don’t always know why.
3. Gene Prediction FAQs

Gene Prediction FAQs (continue)

• Genes in my favorite genome…
  – Q: There is no gene predictor for it, what should I do?
  – A: Training a gene predictor or use one that is for another organism that is close to this genome. But it may be inaccurate.

• Difficult genes
  – Q: why some genes are not predicted by any program?
  – A: They are statistical outliers.
3. Gene Prediction FAQs

Gene Prediction FAQs (continue)

• Just coding exons…
  – Q: why other parts are not predicted, such as non-coding exons, alternative isoforms, non-canonical splice sites, gene within genes?
  – A: There are trade offs.

• Pseudogenes
  – Q: why do some gene predictions have tiny introns?
  – A: Retro-pseudo genes often have very strong coding signals, because they are derived from highly expressed genes.
3. Gene Prediction FAQs

Gene Prediction FAQs (end)

• How can I tell a good gene prediction from a bad one?
  – Scores have been assigned to every exon and intron of a gene. People can tell if a gene prediction is good or not by the scores of exons and introns of this gene.
  – You may have to run the program on your own computer to figure them out!