Genomics & Computational Biology

Section 2

Lan Zhang

Sep. 30th, 2003

Outline

- **How Computers Store Information**
- **Sequence Alignment**
- **Dot Matrix Analysis**
- **Dynamic programming**
  - **Global: Needleman-Wunsch Algorithm**
  - **Local: Smith-Waterman Algorithm**
- **Scoring Matrices**
- **BLAST**

How Computers Store Information

Binary = Number system in base 2

Base 10: $1,234 = 1\times10^3 + 2\times10^2 + 3\times10^1 \times 4\times10^0$

Base 2: $10110 = 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$= 16 + 4 + 2 = 22$ (in decimal)

Each binary digit is referred to as a **"bit"**, eight bits equal one **"byte"**

A 10-digit binary number can hold $2^{10} = 1,024$ different values

Example: What is the minimum amount of memory that can hold the Boston phonebook white pages? (Assume: 700 pages, 400 names/page, 16 letters/name, 7 decimal digits/phone #)

Step 1: The 26 letters of the alphabet can be encoded in how many bits?

Step 2: How many bits to encode a 7 digit decimal number?

Step 3: How many numbers & letters are needed total?

Sequence Alignment

What is sequence alignment?

- The procedure of **comparing** two (pair-wise sequence alignment) or more (multiple sequence alignment) sequences by searching for a series of individual characters or character patterns that are in the same order in the set of sequences.
- In an **optimal alignment**, non-identical characters and gaps are placed to line up as many identical or similar characters as possible.

Why align sequences?

- Sequences that are very much alike, or "**similar**" in the parlance of sequence analysis, probably have the same **function/structure**.
- Understanding **evolutionary events**: mutations, insertions and deletions

Types of Sequence Alignment

- **Pair-wise** alignment vs. **multiple** alignment
- **DNA** seq. alignment vs. **protein** seq. alignment
- Global alignment vs. local alignment
  - **Global:** stretched over the entire sequence length to include as many matching characters as possible up to and including the sequence ends

  - **Local:** stops at the ends of regions of identity or strong similarity, and a much higher priority is given to finding these local regions

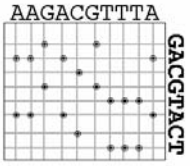
Methods of Pair-Wise Sequence Alignment

- Dot matrix analysis
- The dynamic programming (DP) algorithm
  - Global: Needleman-Wunsch Algorithm
  - Local: Smith-Waterman Algorithm
- Word or k-mer based methods (BLAST and FASTA) (next week)

Dot Matrix

Each position in the matrix D[i,j] is either

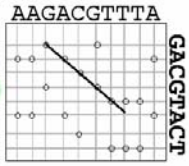
- has a dot, if A[i] == B[j]
- or blank, otherwise.



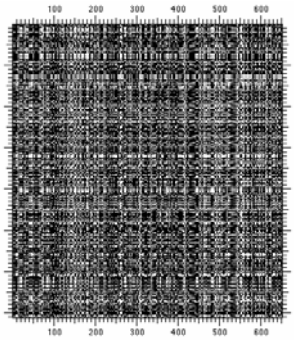
A more advanced dot matrix

With thousands of bases, it is impossible to plot all dots in the matrix. Instead we look for stretches of sequence with few mismatches. If the number of mismatches is less than the cutoff, plot a dot or line.

All diagonals with at least 4 out of 5 matches.



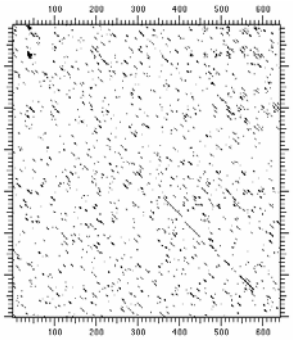
Phage λ and P22 Repressor DNA Seqs



Window size 1

stringency 1

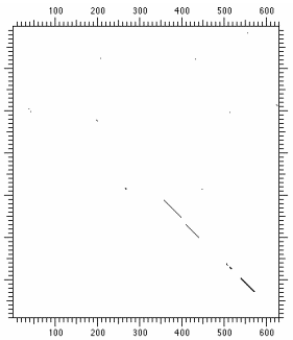
Phage λ and P22 Repressor DNA Seqs



Window size 11

stringency 7

Phage λ and P22 Repressor DNA Seqs



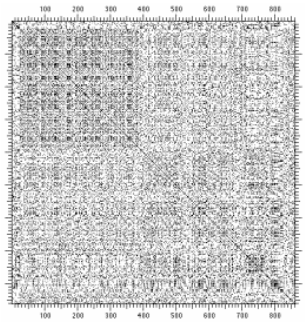
Window size 23

stringency 15

Dot Matrix Analysis

- **Advantage:**
  - Readily reveals the presence of **insertions/deletions**
  - Finds **direct and inverted repeats** that are more difficult to find by the other, more automated methods.
  - Finds regions of complementarity (RNA secondary structure)
  - Finds the location of genes between two genomes
- **Disadvantage/Limitation:**
  - Most dot matrix computer programs **do not show an actual alignment**. Does not return a **score** to indicate how 'optimal' a given alignment is.

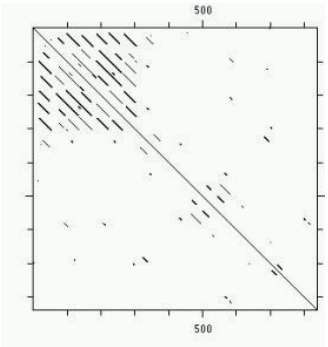
Human LDL Receptor Against Itself



Window size 1

stringency 1

Human LDL Receptor Against Itself



Window size 23

stringency 7

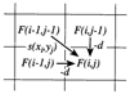
Dynamic Programming (DP)

- A general class of algorithms typically applied to **optimization problems**.
- For DP to be applicable, an optimization problem must have two key ingredients:
  - **Optimal substructure** – an optimal solution to the problem contains within it optimal solutions to sub-problems
  - **Overlapping sub-problems** – the pieces of larger problem have a sequential dependency.
  - *i.e, you can find the best solution by breaking it into pieces and solving each small piece first, and then merging the solutions of all pieces.*
- DP works by first **solving every sub-sub-problem** just once, and **saves its answer** in a table, thereby avoiding the work of re-computing the answer every time the sub-sub-problem is encountered. Each intermediate answer is stored with a score, and DP finally chooses the sequence of solution that yields the **highest score**.
- More details next week.

Global Alignment (Needleman Wunsch)

- General goal is to obtain **optimal global alignment** between two sequences, allowing gaps.
- Construct matrix *F* indexed by *i* and *j*, one index for each sequence, where the value *F*(*i,j*) is the score of the best alignment between the initial segment *x1…i* of *x* and the initial segment *y1…j* of *y*.
- Initializing *F*(0,0) = 0. Fill *F* from top left to bottom right. Calculate *F*(*i,j*) based on *F*(*i-1, j-1*), *F*(*i-1,j*) and *F*(i, *j-1*):

  $$F(i,j) = \max \{ F(i-1, j-1) + s(xi, yj); \; F(i-1,j) - d; \; F(i, j-1) - d. \; \}$$



  where *s*(*a*,*b*) is the score that residues *a* and *b* occur as an aligned pair, and *d* is the gap penalty.
- Once *F* is filled, trace back the path that leads to *F*(*n,m*), which is by definition the **best score for an alignment** of *x1…n* to *y1…m*.

Needleman-Wunsch Algorithm (DNA)

	-	A	C	A	C	T	A
-							
A							
G							
C							
A							
C							
A							
C							
A							

ϖ *(match) = 2*
ϖ *(mismatch) = -1*
ϖ *(gap) = -3*

Needleman-Wunsch Algorithm (DNA)

	-	A	C	A	C	T	A
-	0	-3	-6	-9	-12	-15	-18
A	-3	2	-1	-4	-7	-10	-13
G	-6	-1	1	-2	-5	-8	-11
C	-9	-4	1	0	0	-3	-6
A	-12	-7	-2	3	0	-1	-1
C	-15	-10	-5	0	5	2	-1
A	-18	-13	-8	-3	2	4	4
C	-21	-16	-11	-6	-1	1	3
A	-24	-19	-14	-9	-4	-2	3

ϖ *(match) = 2*
ϖ *(mismatch) = -1*
ϖ *(gap) = -3*

Needleman-Wunsch Algorithm (DNA)

	-	A	C	A	C	T	A
-	0	-3	-6	-9	-12	-15	-18
A	-3	2	-1	-4	-7	-10	-13
G	-6	-1	1	-2	-5	-8	-11
C	-9	-4	1	0	0	-3	-6
A	-12	-7	-2	3	0	-1	-1
C	-15	-10	-5	0	5	2	-1
A	-18	-13	-8	-3	2	4	4
C	-21	-16	-11	-6	-1	1	3
A	-24	-19	-14	-9	-4	-2	3

ω (match) = 2
ω (mismatch) = -1
ω (gap) = -3

Needleman-Wunsch Algorithm (A Bit More Advanced)

- Scoring matrices:
  - Specify different "weights" or "scores" for different matches or mismatches
  - Used in protein sequence alignment
  - More details to be given next week
- Gap penalties:
  - Different penalties for gap opening and gap extension

Needleman Wunsch Algorithm (Protein)

	GAP	M	N	A	L	S	D	R	T
GAP	0	-12	-16	-20	-24	-28	-32	-36	-40
M	-12								
G	-16								
S	-20								
D	-24								
R	-28								
T	-32								
T	-36								
E	-40								
T	-44								

PAM250
Gap opening: -12
Gap extension: -4

Needleman Wunsch Algorithm (Protein)

	GAP	M	N	A	L	S	D	R	T
GAP	0	-12	-16	-20	-24	-28	-32	-36	-40
M	-12	(6)	(-2)						
G	-16	(-3)	(0)						
S	-20								
D	-24								
R	-28								
T	-32								
T	-36								
E	-40								
T	-44								

PAM250
Gap opening: -12
Gap extension: -4

Needleman Wunsch Algorithm (Protein)

	GAP	M	N	A	L	S	D	R	T
GAP	0	-12	-16	-20	-24	-28	-32	-36	-40
M	-12	(6) 6	(-2) -6						
G	-16	(-3) -6	(0) 6						
S	-20								
D	-24								
R	-28								
T	-32								
T	-36								
E	-40								
T	-44								

PAM250
Gap opening: -12
Gap extension: -4

Needleman Wunsch Algorithm (Protein)

	GAP	M	N	A	L	S	D	R	T
GAP	0	-12	-16	-20	-24	-28	-32	-36	-40
M	-12	(6) 6	(-2) -6	-10	-14	-18	-22	-26	-30
G	-16	(-3) -6	(0) 6	-5	-10	-13	-17	-22	-26
S	-20	-10	-5	7	-5	-8	-13	-17	-21
D	-24	-14	-8	-5	3	-5	-4	-14	-17
R	-28	-18	-14	-9	-8	3	-6	2	-10
T	-32	-22	-18	-13	-11	-7	3	-7	5
T	-36	-26	-22	-17	-15	-10	-7	2	-4
E	-40	-30	-25	-21	-20	-15	-7	-8	2
T	-44	-34	-30	-24	-23	-19	-15	-8	-5

PAM250
Gap opening: -12
Gap extension: -4

Needleman Wunsch Algorithm (Protein)

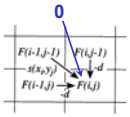
	GAP	M	N	A	L	S	D	R	T
GAP									
M									
G									
S									
D									
R									
T									
T									
E									
T									

PAM250
Gap opening: -12
Gap extension: -4

Local Alignment (Smith-Waterman)

- Two changes from global alignment:
- Possibility of taking the **value 0** if all other options have values less than 0. This corresponds to starting a new alignment.

$$F(i,j) = \max\{0;\ F(i-1, j-1) + s(xi, yj);\ F(i-1,j) - d;\ F(i, j-1) - d.\}$$



- Alignments can end anywhere in the matrix, so instead of taking the value in the bottom right corner, $F(n,m)$ for the best score, we look for the **highest value of $F(i,j)$** over the whole matrix and start the trace-back from there.

Smith-Waterman Algorithm (DNA)

	-	A	C	A	C	T	A
-							
A							
G							
C							
A							
C							
A							
C							
A							

ϖ (match) = 2
ϖ (mismatch) = -1
ϖ (gap) = -3

Smith-Waterman Algorithm (DNA)

	-	A	C	A	C	T	A
-	0	0	0	0	0	0	0
A	0	2	0	2	0	0	2
G	0	0	1	0	1	0	0
C	0	0	2	0	2	0	0
A	0	2	0	4	1	1	2
C	0	0	4	1	6	3	0
A	0	2	1	6	3	5	5
C	0	0	4	3	8	5	4
A	0	2	1	6	5	7	7

ϖ (match) = 2
ϖ (mismatch) = -1
ϖ (gap) = -3

Smith-Waterman Algorithm (DNA)

	-	A	C	A	C	T	A
-	0	0	0	0	0	0	0
A	0	2	0	2	0	0	2
G	0	0	1	0	1	0	0
C	0	0	2	0	2	0	0
A	0	**2**	0	4	1	1	2
C	0	0	**4**	1	6	3	0
A	0	2	1	**6**	3	5	5
C	0	0	4	3	**8**	5	4
A	0	2	1	6	5	7	7

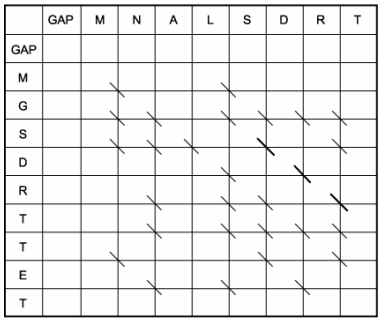
ϖ (match) = 2
ϖ (mismatch) = -1
ϖ (gap) = -3

Smith-Waterman Algorithm (Protein)

	GAP	M	N	A	L	S	D	R	T
GAP	0	0	0	0	0	0	0	0	0
M	0	6	0	0	4	0	0	0	0
G	0	0	6	1	0	5	1	0	0
S	0	0	1	7	0	2	5	1	1
D	0	0	2	1	3	0	6	4	1
R	0	0	0	0	0	3	0	12	3
T	0	0	0	1	0	1	3	0	15
T	0	0	0	1	0	1	1	2	3
E	0	0	1	0	0	0	4	0	2
T	0	0	0	2	0	1	0	3	3

PAM250
Gap opening: -12
Gap extension: -4

Smith-Waterman Algorithm (Protein)



PAM250
Gap opening: -12
Gap extension: -4

Next Week

- Dynamic Programming
- Scoring matrices
- BLAST

Acknowledgement / References

This handout includes material written by Suzanne Komili, Yonatan Grad, Doug Selinger, and Zhou Zhu.

Mount, Bioinformatics – Sequence and Genome Analysis
Durbin et al., Biological Sequence Analysis

http://www.medwiki.net/wiki/moin.cgi/Dot_20Matrix
http://www.bioinfo.rpi.edu/~bystrc/courses/biol4540/lecture3/sld006.htm