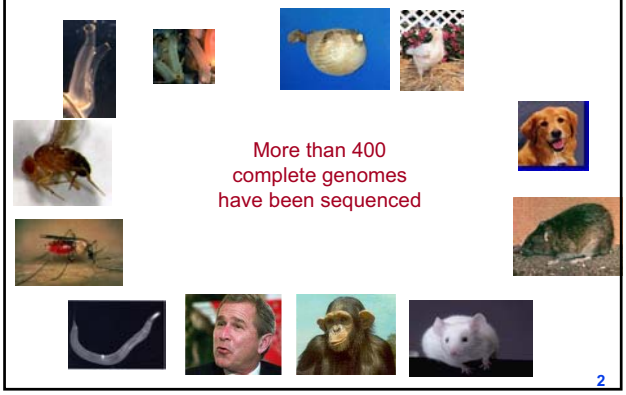
Lecture outline

- Sequence alignment
  - Why do we need to align sequences?
  - Evolutionary relationships
- Gaps and scoring matrices
- Dynamic programming
  - Global alignment (Needleman & Wunsch)
  - Local alignment (Smith & Waterman)
- Database searches
  - BLAST
  - FASTA

1

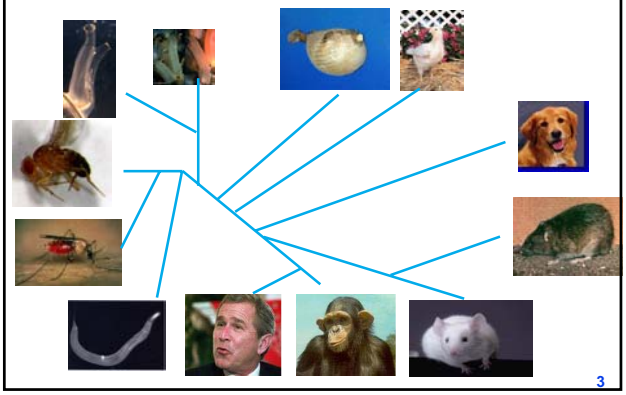
Complete DNA Sequences



More than 400
complete genomes
have been sequenced

2

Evolution



3

Sequence conservation implies function

Alignment is the key to
- Finding important regions
- Determining function
- Uncovering the evolutionary forces

Sequence alignment

- Comparing DNA/protein sequences for
  - Similarity
  - Homology
- Prediction of function
- Construction of phylogeny
- Shotgun assembly
  - End-space-free alignment / overlap alignment
- Finding motifs

5

Sequence Alignment

Procedure of comparing two (pairwise) or more (multiple) sequences by searching for a series of individual characters that are in the same order in the sequences

```
GCTAGTCAGATCTGACGCTA
|  ||||  |||||  |||
TGGTCACATCTGCCGC
```

6

Sequence Alignment

Procedure of comparing two (pairwise) or more (multiple) sequences by searching for a series of individual characters that are in the same order in the sequences

```
VLSPADKTNVKAAWGKVGAHAGYEG
|||   |     |    | ||  |      ||
VLSEGDWQLVLHVWAKVEADVAGEG
```

7

Sequence Alignment

```
AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC

-AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---
TAG-CTATCAC--GACCGC--GGTCGATTTGCCCGAC
```

Definition

Given two strings $x = x_1x_2...x_M$, $y = y_1y_2...y_N$,

an alignment is an assignment of gaps to positions 0,…, M in x, and 0,…, N in y, so as to line up each letter in one sequence with either a letter, or a gap in the other sequence

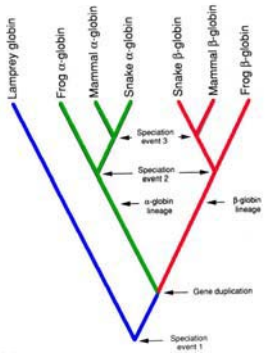
8

Homology

- Orthologs
  - Divergence follows speciation
  - Similarity can be used to construct phylogeny between species
- Paralogs
  - Divergence follows duplication
- Xenologs
- Article on terminology
- ISMB tutorial on protein sequence comparison

9

Orthologs and paralogs



10

Understanding evolutionary relationships

molecular molecular

Nothing in biology makes sense except in the light of evolution

Dobzhansky, 1973

11

Sources of variation

- Nucleotide substitution
  - Replication error
  - Chemical reaction
- Insertions or deletions (indels)
  - Unequal crossing over
  - Replication slippage
- Duplication
  - a single gene (complete gene duplication)
  - part of a gene (internal or partial gene duplication)
    - Domain duplication
    - Exon shuffling
  - part of a chromosome (partial polysomy)
  - an entire chromosome (aneuploidy or polysomy)
  - the whole genome (polyploidy)

12

Differing rates of DNA evolution

- Functional/selective constraints (particular features of coding regions, particular features in 5' untranslated regions)
- Variation among different gene regions with different functions (different parts of a protein may evolve at different rates).
- Within proteins, variations are observed between
  - surface and interior amino acids in proteins (order of magnitude difference in rates in haemoglobins)
  - charged and non-charged amino acids
  - protein domains with different functions
  - regions which are strongly constrained to preserve particular functions and regions which are not
  - different types of proteins -- those with constrained interaction surfaces and those without

13

Common assumptions

- All nucleotide sites change independently
- The substitution rate is constant over time and in different lineages
- The base composition is at equilibrium
- The conditional probabilities of nucleotide substitutions are the same for all sites, and do not change over time
- Most of these are not true in many cases…

14

Lecture outline

- Sequence alignment
  - Why do we need to align sequences?
  - Evolutionary relationships
- Gaps and scoring matrices
- Dynamic programming
  - Global alignment (Needleman & Wunsch)
  - Local alignment (Smith & Waterman)
- Database searches
  - BLAST
  - FASTA

15

A simple alignment

- Let us try to align two short nucleotide sequences:
  - AATCTATA and AAGATA
- Without considering any gaps (insertions/deletions) there are 3 possible ways to align these sequences

```
AATCTATA    AATCTATA    AATCTATA
AAGATA       AAGATA      AAGATA
```

- Which one is better?

16

What is a good alignment?

```
AGGCTAGTT,  AGCGAAGTTT

AGGCTAGTT-          6 matches, 3 mismatches, 1 gap
AGCGAAGTTT

AGGCTA-GTT-         7 matches, 1 mismatch, 3 gaps
AG-CGAAGTTT

AGGC-TA-GTT-        7 matches, 0 mismatches, 5 gaps
AG-CG-AAGTTT
```

17

Scoring the alignments

- We need to have a scoring mechanism to evaluate alignments
  - match score
  - mismatch score
- We can have the total score as:

$$\sum_{i=1}^{n} \text{match or mismatch score at position i}$$

- For the simple example, assume a match score of 1 and a mismatch score of 0:

```
AATCTATA    AATCTATA    AATCTATA
AAGATA       AAGATA      AAGATA
   4            1           3
```

18

3

Good alignments require gaps

- Maximal consecutive run of spaces in alignment
  - Matching mRNA (cDNA) to DNA
  - Shortening of DNA/protein sequences
  - Slippage during replication
  - Unequal crossing-over during meiosis
  - …
- We need to have a scoring function that considers gaps also

19

Simple alignment with gaps

- Considering gapped alignments vastly increases the number of possible alignments:

AATCTATA	AATCTATA	AATCTATA	more?
AAG-AT-A	AA-G-ATA	AA--GATA	
1	3	3	

- If gap penalty is -1 what will be the new scores?

20

Scoring Function

- Sequence edits:

AGGCCTC

  - Mutations — AGGACTC
  - Insertions — AGGGCCTC
  - Deletions — AGG . CTC

Alternative definition:

minimal edit distance

"Given two strings x, y, find minimum # of edits (insertions, deletions, mutations) to transform one string to the other"

Scoring Function:

Match: +m
Mismatch: -s
Gap: -d

Score F = (# matches) × m - (# mismatches) × s – (#gaps) × d

21

More complicated gap penalties

- Nature favors small number of long gaps compared to large number of short gaps.
- How do we adjust our scoring scheme to account for this fact above?

  By having different gap opening and gap extension penalties.

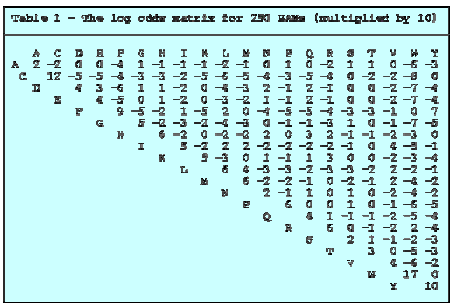
- Choices of gap penalties
  - Linear
  - Affine
    - Gap open penalty
    - Gap extension penalty
  - Arbitrary

22

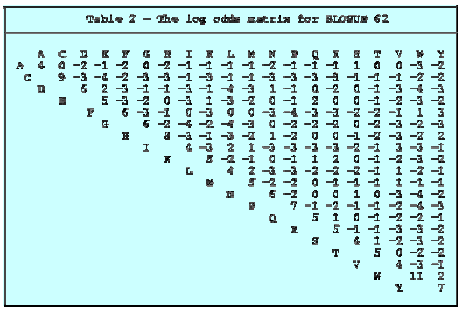
Score matrix

- Instead of having a single match/mismatch score for every pair of nucleotides or amino acids, consider chemical, physical, evolutionary relationships:
  - E.g.
    - alanine vs. valine or alanine vs. lysine? Alanine and valine are both small and hydrophobic, but lysine is large and charged.
    - which substitutions occur more in nature?
- Assign scores to each pair of symbol
  - Higher score means more similarity

23



24



25

Major Differences between PAM and BLOSUM

PAM	BLOSUM
Built from global alignments	Built from local alignments
Built from small amout of Data	Built from vast amout of Data
Counting is based on minimum replacement or maximum parsimony	Counting based on groups of related sequences counted as one
Perform better for finding global alignments and remote homologs	Better for finding local alignments
Higher PAM series means more divergence	Lower BLOSUM series means more divergence

26

Typical score matrix

- DNA
  - Match = +1
  - Mismatch = -3
  - Gap penalty = -5
  - Gap extension penalty = -2
- Protein sequences
  - Blossum62 matrix
  - Gap open penalty = -11
  - Gap extension = -1

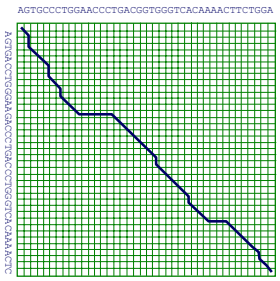
27

Lecture outline

- Sequence alignment
  - Why do we need to align sequences?
  - Evolutionary relationships
- Gaps and scoring matrices
- Dynamic programming
  - Global alignment (Needleman & Wunsch)
  - Local alignment (Smith & Waterman)
- Database searches
  - BLAST
  - FASTA

28

How do we compute the best alignment?



Too many possible alignments:

$>> 2^N$

(exercise)

29

Alignment is additive

Observation:

The score of aligning $x_1 \ldots x_M$ to $y_1 \ldots y_N$ is additive

Say that $x_1 \ldots x_i$ aligns to $y_1 \ldots y_j$, and $x_{i+1} \ldots x_M$ aligns to $y_{j+1} \ldots y_N$

The two scores add up:

$F(x[1:M], y[1:N]) = F(x[1:i], y[1:j]) + F(x[i+1:M], y[j+1:N])$

30

5

Errata for the textbook

- Dynamic Programming section in the textbook (pages 41-45) contains some errors:

1. Figure 2.5: caption should be "CACGA and CGA" instead of "CACGA and CCGA" and at row 1, column 3 "CGA" should be "GA".
2. page 44. line 13: "optimal alignment will be 1" should be "optimal alignment will be 2".

31

Types of alignment

- Global (Needleman & Wunsch)
  - Strings of similar size
    - Genes with a similar structure
    - Larger regions with a preserved order (syntenic regions)
- Local (Smith & Waterman)
  - Finding similar regions among
    - Dissimilar regions
    - Sequences of different lengths

32

Dynamic programming

- Instead of evaluating every possible alignment, we can create a table of partial scores by breaking the alignment problem into subproblems.
- Consider two sequences CACGA and CGA
  - we have three possibilities for the first position of the alignment

First position	Score	Remaining seqs.
C C	+1	ACGA GA
- C	-1	CACGA GA
C -	-1	ACGA CGA

33

Example

score(H,P) = -2, gap penalty=-8 (linear)

	-	H	E	A	G	A	W	G	H	E	E
-	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	-8	-2									
A	-16										
W	-24										
H	-32										
E	-40										
A	-48										
E	-56										

34

The DP recurrence relation

- s(a,b) = score of aligning a and b
- F(i,j) = optimal similarity of A(1:i) and B(1:j)
- Recurrence relation
  - $F(i,0) = \Sigma\, s(A(k),-),\ 0 <= k <= i$
  - $F(0,j) = \Sigma\, s(-,B(k)),\ 0 <= k <= j$
  - $F(i,j) = \max [F(i,j-1) + s(-,B(j)),$
    $F(i-1,j) + s(A(i),-),$
    $F(i-1,j-1) + s(A(i),B(j))]$
  - Assume linear gap penalty

35

Example contd.

score(E,P) = 0, score(E,A) = -1, score(H,A) = -2

	-	H	E	A	G	A	W	G	H	E	E
-	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	-8	-2	-8								
A	-16	-10	-3								
W	-24										
H	-32										
E	-40										
A	-48										
E	-56										

36

Optimal alignment: H E A G A W G H E - E
- P - - A W - H E A E

		H	E	A	G	A	W	G	H	E	E
	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	-8	-2	-8	-16	-24	-33	-42	-49	-57	-65	-73
A	-16	-10	-3	-4	-12	-19	-28	-36	-44	-52	-60
W	-24	-18	-11	-6	-7	-15	-4	-12	-21	-29	-37
H	-32	-14	-18	-13	-8	-9	-12	-6	-2	-11	-19
E	-40	-22	-8	-16	-16	-9	-12	-14	-6	-4	-5
A	-48	-30	-16	-3	-11	-11	-12	-12	-14	-4	2
E	-56	-38	-24	-11	-6	-12	-14	-15	-12	-8	2

The value in the final cell is the best score for the alignment

37

More examples

- Sequence alignment applet at:

  http://www.iro.umontreal.ca/~casagran/baba.html

38

Semi-global alignment

- In Needleman&Wunsch DP algorithm the gap penalty is assessed regardless of whether gaps are located internally or at the terminal ends.
- Terminal gaps may not be biologically significant

```
AATCTATA
--TCT---
```

- Treat terminal gaps differently than internal gaps → semi-global alignment
- What modifications should be made to the original DP?

39

Local sequence alignment

- Suppose, we have a long DNA sequence (e.g., 4000 bp) and we want to compare it with the complete yeast genome (12.5M bp).
- What if only a portion of our query, say 200 bp length, has strong similarity to a gene in yeast.
  - Can we find this 200 bp portion using (semi) global alignment?

    Probably not. Because, we are trying to align the complete 4000 bp sequence, thus a random alignment may get a better score than the one that aligns 200 bp portion to the similar gene in yeast.

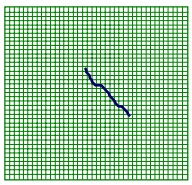
40

The local alignment problem

Given two strings $x = x_1 \ldots\ldots x_M$,
$y = y_1 \ldots\ldots y_N$

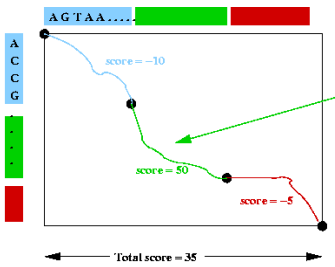
Find substrings x', y' whose similarity (optimal global alignment value) is maximum

x = aaaaccccccgggtta
y = ttcccgggaaccaacc



41

Local alignment



local alignment may have higher score than overall global alignment

42

7

Local sequence alignment (Smith-Waterman)

- F(i,j) = optimal local similarity among suffixes of A(1:i) and B(1:j)
- Recurrence relation
  - F(i,0) = 0
  - F(0,j) = 0
  - F(i,j) = max [0,
    F(i,j-1) + s(-,B(j)),
    F(i-1,j) + s(A(i),-),
    F(i-1,j-1) + s(A(i),B(j))]
  - Assume linear gap model

43

Example

Q: E Q L L K A L E F K L
P: K V L E F G Y

Linear gap model
Gap = -1
Match = 4
Mismatch = -2

	-	E	Q	L	L	K	A	L	E	F	K	L
-												
K												
V												
L												
E												
F												
G												
Y												

44

Example

Q: E Q L L K A L E F K L
P: K V L E F G Y

Linear gap model
Gap = -1
Match = 4
Mismatch = -2

	-	E	Q	L	L	K	A	L	E	F	K	L
-	0	0	0	0	0	0	0	0	0	0	0	0
K	0											
V	0											
L	0											
E	0											
F	0											
G	0											
Y	0											

45

Example

Q: E Q L L K A L E F K L
P: K V L E F G Y

Linear gap model
Gap = -1
Match = 4
Mismatch = -2

	-	E	Q	L	L	K	A	L	E	F	K	L
-	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	4	3	2	1	0	4	3
V	0	0	0	0	0	3	2	1	0	0	3	2
L	0	0	0	4	4	3	2	6	5	4	3	7
E	0	4	3	3	3	2	1	5	10	9	8	7
F	0	3	2	2	2	1	0	4	9	14	13	12
G	0	2	1	1	1	0	0	3	8	13	12	11
Y	0	1	0	0	0	0	0	2	7	12	11	10

46

Example

Q: E Q L L K A L E F K L
P: K V L E F G Y

Linear gap model
Gap = -1
Match = 4
Mismatch = -2

	-	E	Q	L	L	K	A	L	E	F	K	L
-	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	4	3	2	1	0	4	3
V	0	0	0	0	0	3	2	1	0	0	3	2
L	0	0	0	4	4	3	2	6	5	4	3	7
E	0	4	3	3	3	2	1	5	10	9	8	7
F	0	3	2	2	2	1	0	4	9	14	13	12
G	0	2	1	1	1	0	0	3	8	13	12	11
Y	0	1	0	0	0	0	0	2	7	12	11	10

47

Example

Q: E Q L L K A L E F K L
P: K V L E F G Y

Alignment

Q: K A - L E F
P: K - V L E F

Linear gap model
Gap = -1
Match = 4
Mismatch = -2

	-	E	Q	L	L	K	A	L	E	F	K	L
-	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	4	3	2	1	0	4	3
V	0	0	0	0	0	3	2	1	0	0	3	2
L	0	0	0	4	4	3	2	6	5	4	3	7
E	0	4	3	3	3	2	1	5	10	9	8	7
F	0	3	2	2	2	1	0	4	9	14	13	12
G	0	2	1	1	1	0	0	3	8	13	12	11
Y	0	1	0	0	0	0	0	2	7	12	11	10

48

8

Example

Q: E Q L L K A L E F K L
P: K V L E F G Y

Alignment

Q: K - A L E F
P: K V - L E F

	-	E	Q	L	L	K	A	L	E	F	K	L
-	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	4	3	2	1	0	4	3
V	0	0	0	0	0	3	2	1	0	0	3	2
L	0	0	0	4	4	3	2	6	5	4	3	7
E	0	4	3	3	3	2	1	5	10	9	8	7
F	0	3	2	2	2	1	0	4	9	14	13	12
G	0	2	1	1	1	0	0	3	8	13	12	11
Y	0	1	0	0	0	0	0	2	7	12	11	10

49

Example

Q: E Q L L K A L E F K L
P: K V L E F G Y

Alignment

Q: K A L E F
P: K V L E F

	-	E	Q	L	L	K	A	L	E	F	K	L
-	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	4	3	2	1	0	4	3
V	0	0	0	0	0	3	2	1	0	0	3	2
L	0	0	0	4	4	3	2	6	5	4	3	7
E	0	4	3	3	3	2	1	5	10	9	8	7
F	0	3	2	2	2	1	0	4	9	14	13	12
G	0	2	1	1	1	0	0	3	8	13	12	11
Y	0	1	0	0	0	0	0	2	7	12	11	10

50

Another Example

Find the local alignment between:

Q: G C T G G A A G G C A T
P: G C A G A G C A C G

Linear gap model
Gap = -4
Match = +5
Mismatch = -4

Q

P	--	G	C	T	G	G	A	A	G	G	C	A	T
--	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0												
C	0												
A	0												
G	0												
A	0												
G	0												
C	0												
A	0												
C	0												
G	0												

51

Another Example

Q's subsequence: G A A G - G C A
P's subsequence: G C A G A G C A

Q

P	--	G	C	T	G	G	A	A	G	G	C	A	T
--	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5	1	0	5	5	1	0	5	5	1	0	0
C	0	1	10	6	2	1	1	0	1	1	10	6	2
A	0	0	6	6	2	0	6	6	2	0	6	15	11
G	0	5	2	2	11	7	3	2	11	7	3	11	11
A	0	1	1	0	7	7	11	8	7	7	3	8	7
G	0	5	1	0	5	11	7	7	13	12	8	4	4
C	0	0	10	6	2	7	7	3	9	8	17	13	9
A	0	0	6	6	2	3	11	12	8	5	13	22	18
C	0	0	5	2	2	0	7	8	8	4	18	18	18
G	0	5	1	1	7	7	5	4	13	13	14	14	14

52

Local vs. Global alignment

PIR Entry				Similarity Score: Global (End Penalty)	Global (No End Penalty)	Local
HBHU	vs	HBHU	Hemoglobin beta-chain—human	725	725	725
		HAHU	Hemoglobin alpha-chain—human	314	320	322
		MYHU	Myoglobin—Human	121	164	166
		GPYL	Leghemoglobin—Yellow lupin	8	28	43
		LZCH	Lysozyme precursor—Chicken	−107	16	32
		NRBO	Pancreatic ribonuclease—Bovine	−124	16	31
		CCHU	Cytochrome c—Human	−160	10	26
MCHU	vs	MCHU	Calmodulin—Human	671	671	671
		TPHUCS	Troponin C, skeletal muscle	395	430	438
		PVPK2	Parvalbumin beta—Pike	−57	103	115
		CHHUH	Calpain heavy chain—Human	−2085	89	100
		AQJFNV	Aequorin precursor—Jelly fish	−65	48	76
		KLSWM	Calcium binding protein—Scallop	−89	45	52
QRHULD	vs	EGMSMG	Epidermal growth factor precursor	−591	475	655

53

Complexity

- O(mn) time
- O(mn) space
  - O(max(m,n)) if only distance value is needed
- More complicated "divide-and-conquer" algorithm that doubles time complexity and uses O(min(m,n)) space [Hirschberg, JACM 1977]

54

9

Time and space bottlenecks

- Comparing two one-megabase genomes.
- Space:
  An entry: 4 bytes;
  Table: $4 * 10^6 * 10^6 = 4$ T bytes memory.
- Time:
  1000 MHz CPU: 1M entries/second;
  10^{12} entries: 1M seconds = 10 days.

55

Lecture outline

- Sequence alignment
  - Why do we need to align sequences?
  - Evolutionary relationships
- Gaps and scoring matrices
- Dynamic programming
  - Global alignment (Needleman & Wunsch)
  - Local alignment (Smith & Waterman)
- Database searches
  - BLAST
  - FASTA

56

BLAST

- Basic Local Alignment Search Tool
  - Altschul *et al*. 1990,1994,1997
- Heuristic method for local alignment
- Designed specifically for database searches
- Idea: good alignments contain short lengths of exact matches

57

Steps of BLAST

1. Filter low complexity regions (optional)
2. *Query words* of length 3 (for proteins) or 11 (for DNA) are created from query sequence using a sliding window

```
MEFPGLGSLGTSEPLPQFVDPALVSS
MEF
 EFP
  FPG
   PGL
    GLG
```

58

Steps of BLAST

3. Scan each database sequence for an exact match to *query words**. Each match is a *seed* for an ungapped alignment.

*Blast actually uses a list of *high scoring words* created from words *similar* to query words.

59

Steps of BLAST

4. *(Original BLAST)* extend matching words to the left and right using ungapped alignments. Extension continues as long as score increases or stays same. This is a HSP (high scoring pair).

   (BLAST2) Matches along the same diagonal (think dot plot) within a distance A of each other are joined and then the longer sequence extended as before. Need at least two contiguous hits for extension.

60

Steps of BLAST

5. Using a cutoff score S, keep only the extended matches that have a score at least S.
6. Determine statistical significance of each remaining match.
7.
   - *(Original BLAST)* Only ungapped alignments; sometimes combined together
   - *(BLAST2)* Extend the HSPs using gapped alignment

61

Summarizing BLAST

- One of the few algorithms to make it as a verb
  - Blast(v): to run a BLAST search against a sequence database
- Extension is the most time-consuming step
- BLAST2 reported to be 3 times faster than the original version at same quality

62

Example BLAST run

- BLAST website

63

Steps of FASTA

1. Find k-tups in the two sequences (k=1-2 for proteins, 4-6 for DNA sequences)
2. Select top 10 scoring "local diagonals" with matches and mismatches but no gaps.
   a. For proteins, each k-tup found is scored using the PAM250 matrix
   b. For DNA, use the number of k-tups found
   c. Penalize intervening regions of mismatches

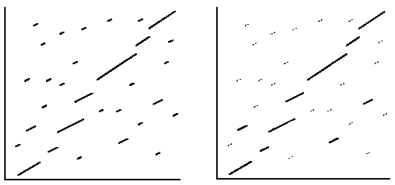
64

Finding k-tups

```
position 1 2 3 4 5 6 7 8 9 10 11
protein  1 n c s p t a . . . . .
protein  2 . . . . . a c s p r k
                    position in          offset
amino acid       protein A protein B   pos A - posB
-----------------------------------------------------
a                    6         6             0
c                    2         7            -5
k                    -        11
n                    1         -
p                    4         9            -5
r                    -        10
s                    3         8            -5
t                    5         -
-----------------------------------------------------
Note the common offset for the 3 amino acids c,s and p
A possible alignment is thus quickly found -
protein 1 n c s p t a
            | | |
protein 2 a c s p r k
```

65

FASTA



Finding 10 best diagonal runs

66

FASTA

3. Rescan top 10 diagonals (representing alignments), score with PAM250 (proteins) or DNA scoring matrix. Trim off the ends of the regions to achieve highest scores.
4. Join regions that are consistent with gapped alignments. (maximal weighted paths in a graph).

67

FASTA

5. After finding the best initial region (step 3), FASTA performs a DP global alignment in a gap centered on the best initial region.

68

Summarizing FASTA

- Statistics based on histograms on values of intermediate and final scores.
- Begins with exact matches unlike BLAST
- Less of a statistical basis for comparison
- Quality and complexity similar to BLAST

69

History of sequence searching

- 1970: NW
- 1980: SW
- 1985: FASTA
- 1989: BLAST
- 1997: BLAST2

70