Biological Sequence Alignment

Ryan Thompson

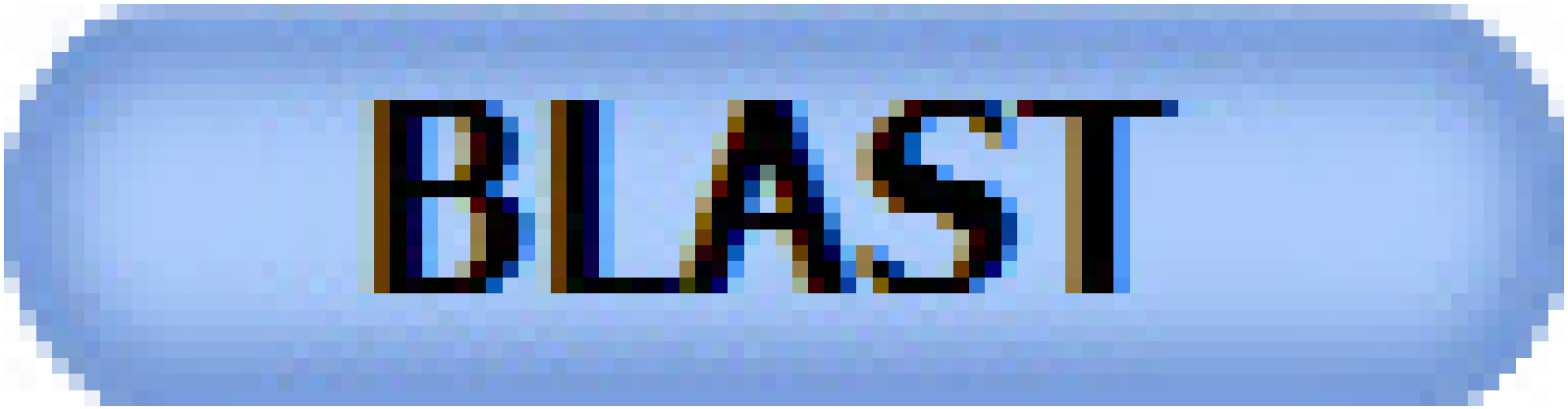
November 13, 2008

Outline

Outline

Outline

What does this button do?



The Biological Question

How can we tell if two genes/proteins are related?

- Without Sequencing:
  - chemical & physical properties (size, *pI*, hydrophobicity, etc.)
  - biological activity
  - localization
- But this is all circumstantial evidence. Can we do better?
- With Sequencing:
  - Compare the primary sequences!
    (Ok, it's still circumstantial, but at least we can do statistics now.)

The Biological Question

How can we tell if two genes/proteins are related?

- Without Sequencing:
  - chemical & physical properties (size, *pI*, hydrophobicity, etc.)
  - biological activity
  - localization
- But this is all circumstantial evidence. Can we do better?
- With Sequencing:
  - Compare the primary sequences!
    (Ok, it's still circumstantial, but at least we can do statistics now.)

The Biological Question

How can we tell if two genes/proteins are related?

- Without Sequencing:
  - chemical & physical properties (size, *pI*, hydrophobicity, etc.)
  - biological activity
  - localization
- But this is all circumstantial evidence. Can we do better?
- With Sequencing:
  - Compare the primary sequences!
    (Ok, it's still circumstantial, but at least we can do statistics now.)

The Biological Question

How can we tell if two genes/proteins are related?

- Without Sequencing:
  - chemical & physical properties (size, *pI*, hydrophobicity, etc.)
  - biological activity
  - localization
- But this is all circumstantial evidence. Can we do better?
- With Sequencing:
  - Compare the primary sequences!
    (Ok, it's still circumstantial, but at least we can do statistics now.)

Outline

But Before We Can Compare Sequences...

We need sequences!

But Before We Can Compare Sequences...

We need sequences!

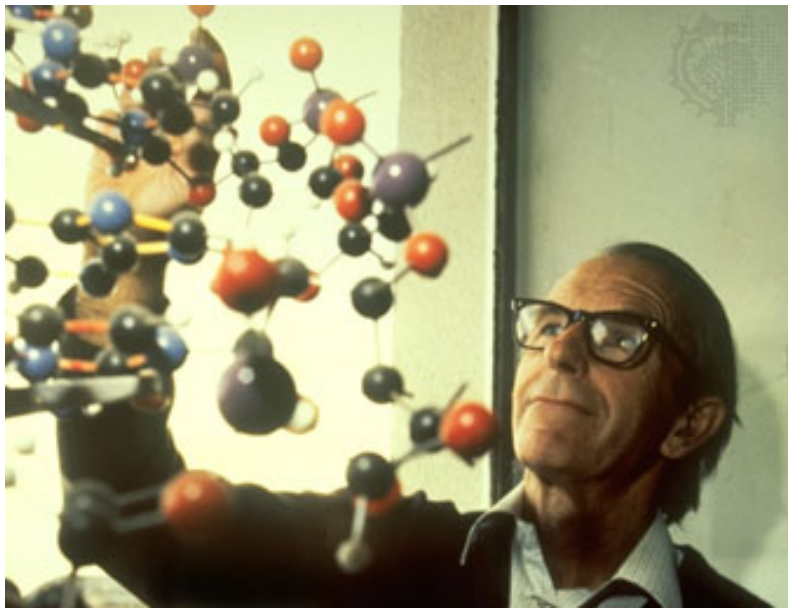
```
Score =  ??? bits (???),  Expect = ?e-??
Identities = ??/?? (??%), Positives = ??/?? (??%), Gaps = ?/?? (?%)
Query  25   ??????????????????????????????????????-----????????????????? 79
            ?????????????????????????? ??? ?  ??     ?    ??      ???
Sbjct  25   ??????????????????????????????????????????????????????????? 84
Query  80   ??????????????????????  105
              ?????????? ????????????
Sbjct  85   ??????????????????????  110
```

Protein Sequencing

- Pehr Edman
  - 1950: Developed a method for N-terminal polypeptide sequencing
- Frederick Sanger
  - 1955: Determined the *complete* sequence of insulin
    - Trypsin digestion, chromatography, and inference
  - Showed that proteins have precise primary sequences
- In practice, most protein sequences are predicted from DNA sequences

Protein Sequencing

- Pehr Edman
  - 1950: Developed a method for N-terminal polypeptide sequencing
- Frederick Sanger
  - 1955: Determined the *complete* sequence of insulin
    - Trypsin digestion, chromatography, and inference
  - Showed that proteins have precise primary sequences
- In practice, most protein sequences are predicted from DNA sequences

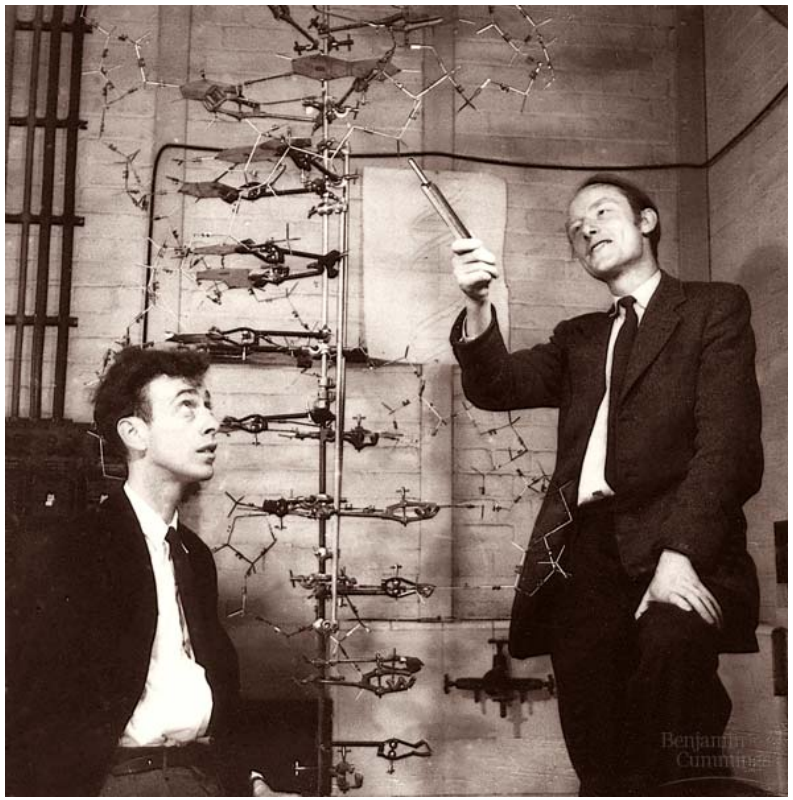


Protein Sequencing

- Pehr Edman
  - 1950: Developed a method for N-terminal polypeptide sequencing
- Frederick Sanger
  - 1955: Determined the *complete* sequence of insulin
    - Trypsin digestion, chromatography, and inference
  - Showed that proteins have precise primary sequences
- In practice, most protein sequences are predicted from DNA sequences

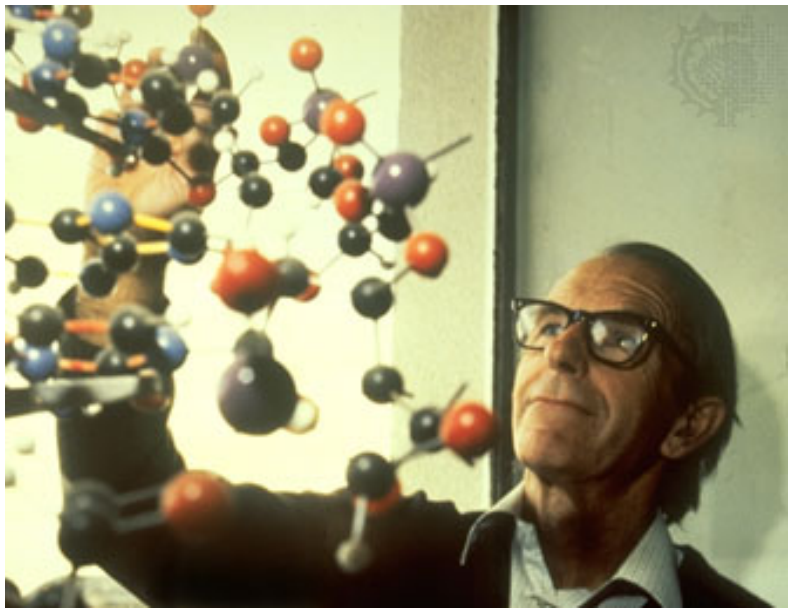
DNA Sequencing

- James D. Watson & Francis Crick
  - 1953: Proposed base-pairing as a model for DNA replication
  - But their model also implied a primary sequence for DNA
- Fredrick Sanger
  - 1975: Dideoxy chain termination sequencing method
  - 1977: Sequenced and manually assembled an entire phage genome



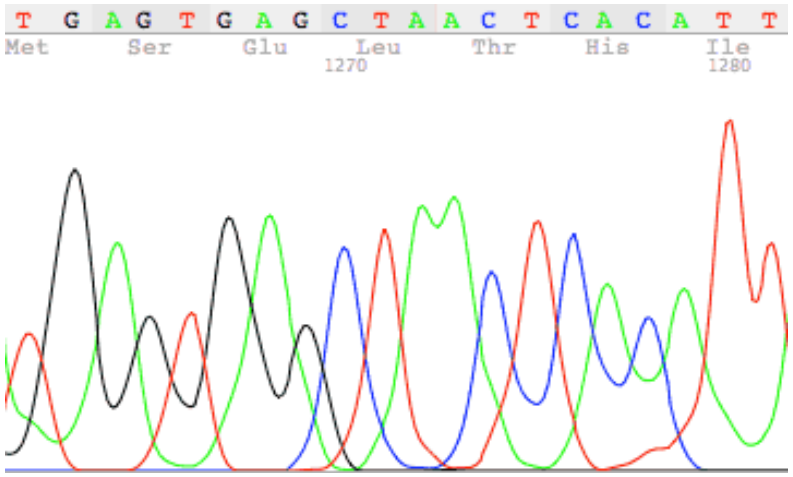
DNA Sequencing

- James D. Watson & Francis Crick
  - 1953: Proposed base-pairing as a model for DNA replication
  - But their model also implied a primary sequence for DNA
- Fredrick Sanger
  - 1975: Dideoxy chain termination sequencing method
  - 1977: Sequenced and manually assembled an entire phage genome



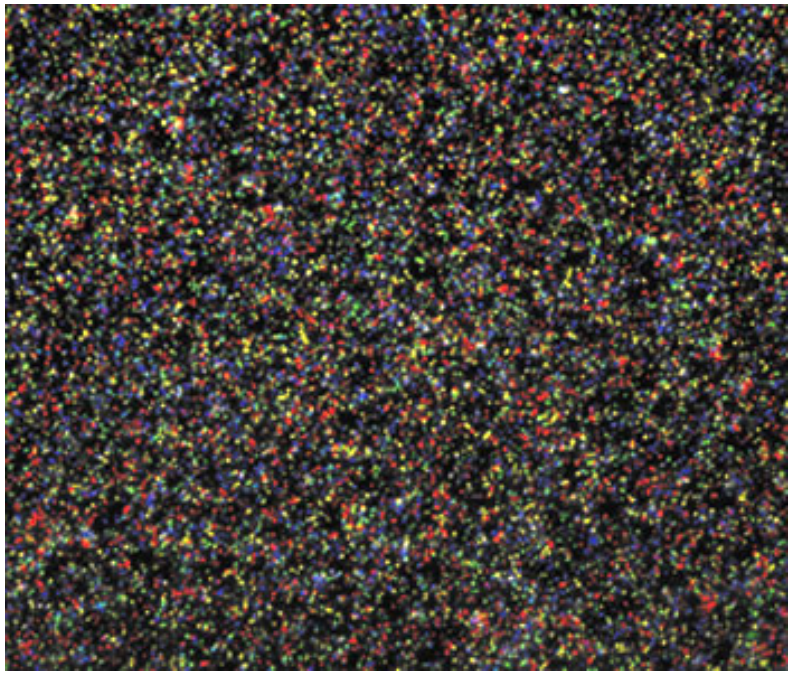
Genome Sequencing

- Later embellishments
  - 4-color fluorescent chain-terminators
  - pyrosequencing
  - high-throughput parallel sequencing



Genome Sequencing

- Later embellishments
  - 4-color fluorescent chain-terminators
  - pyrosequencing
  - high-throughput parallel sequencing



Ok, we've got some sequences

What was the question again?

Oh yeah, how do we tell if they're related?

Ok, we've got some sequences

What was the question again?

Oh yeah, how do we tell if they're related?

Outline

Theory of Molecular Evolution

- Our plan was to compare primary sequences
  - But how can we relate the primary sequence to evolutionary history?
- Evolution happens one mutation at a time
  - frequently only changing a singe base/amino acid
  - diverging genes accumulate divergent mutations
  - therefore, closely related genes should have similar primary sequences

Theory of Molecular Evolution

- Our plan was to compare primary sequences
  - But how can we relate the primary sequence to evolutionary history?
- Evolution happens one mutation at a time
  - frequently only changing a singe base/amino acid
  - diverging genes accumulate divergent mutations
  - therefore, closely related genes should have similar primary sequences

Theory of Molecular Evolution

- Our plan was to compare primary sequences
  - But how can we relate the primary sequence to evolutionary history?
- Evolution happens one mutation at a time
  - frequently only changing a singe base/amino acid
  - diverging genes accumulate divergent mutations
  - therefore, closely related genes should have similar primary sequences

Theory of Molecular Evolution

- Our plan was to compare primary sequences
  - But how can we relate the primary sequence to evolutionary history?
- Evolution happens one mutation at a time
  - frequently only changing a singe base/amino acid
  - diverging genes accumulate divergent mutations
  - therefore, closely related genes should have similar primary sequences

Theory of Molecular Evolution

- Our plan was to compare primary sequences
  - But how can we relate the primary sequence to evolutionary history?
- Evolution happens one mutation at a time
  - frequently only changing a singe base/amino acid
  - diverging genes accumulate divergent mutations
  - therefore, closely related genes should have similar primary sequences

Theory of Molecular Evolution

- Our plan was to compare primary sequences
  - But how can we relate the primary sequence to evolutionary history?
- Evolution happens one mutation at a time
  - frequently only changing a singe base/amino acid
  - diverging genes accumulate divergent mutations
  - therefore, closely related genes should have similar primary sequences

The Molecular Basis of Evolution (1959)

A short excerpt

TABLE 9

Variations in Amino Acid Sequences Among Different Preparations of ACTH

Preparation	Species	Residue No. 25 26 27 28 29 30	31 32 33
β-Corticotropin	sheep, beef	Ala.Gly.Glu.Asp.Asp.Glu	Ala.Ser.Glu.NH_2
Corticotropin A	pig	Asp.Gly.Ala.Glu.Asp.Glu	Leu.Ala.Glu

Two points are of particular interest in regard to the sequences shown. First, the corticotropins of sheep and beef are identical and differ from that of the pig. This finding is consonant with the closer phylogenetic relationship of sheep and cows to each other than of either to pigs. Second, chemical differences are found only in that portion of the ACTH molecule which has been shown to be unessential for hormonal activity. Genetic mutations leading to such differences might, therefore, not be expected to impose significant disadvantages in terms of survival, and these genes could become established in the gene pools of the species.

Outline

Outline

"Optimal" Alignment

What does that mean?

- Always finds the best possible alignment between any two sequences
- Naïve optimal algorithm: try every possible alignment
  - Remeber, that includes all possible gaps
  - How long would this take? A very long time.
- Can we find a faster way? Hint: Yes

"Optimal" Alignment

What does that mean?

- Always finds the best possible alignment between any two sequences
- Naïve optimal algorithm: try every possible alignment
  - Remeber, that includes all possible gaps
    - How long would this take? A very long time.
- Can we find a faster way? Hint: Yes

"Optimal" Alignment

What does that mean?

- Always finds the best possible alignment between any two sequences
- Naïve optimal algorithm: try every possible alignment
  - Remeber, that includes all possible gaps
  - How long would this take? A very long time.
- Can we find a faster way? Hint: Yes

"Optimal" Alignment

What does that mean?

- Always finds the best possible alignment between any two sequences
- Naïve optimal algorithm: try every possible alignment
  - Remeber, that includes all possible gaps
  - How long would this take? A very long time.
- Can we find a faster way? Hint: Yes

"Optimal" Alignment

What does that mean?

- Always finds the best possible alignment between any two sequences
- Naïve optimal algorithm: try every possible alignment
  - Remeber, that includes all possible gaps
  - How long would this take? A very long time.
- Can we find a faster way? Hint: Yes

Dynamic Programming

- Don't worry, we're not talking about computer programming here
- "Programming" in this case means "optimization" or "planning ahead"
- Like checking Google Maps *before* you try to find your way on your own.
  - You use some extra time checking the map and printing it out
  - But then you save time because you don't get lost and backtrack
- In our case, we'll map the whole "search space" and *then* find the best alignment

Dynamic Programming

- Don't worry, we're not talking about computer programming here
- "Programming" in this case means "optimization" or "planning ahead"
- Like checking Google Maps *before* you try to find your way on your own.
  - You use some extra time checking the map and printing it out
  - But then you save time because you don't get lost and backtrack
- In our case, we'll map the whole "search space" and *then* find the best alignment

Dynamic Programming

- Don't worry, we're not talking about computer programming here
- "Programming" in this case means "optimization" or "planning ahead"
- Like checking Google Maps *before* you try to find your way on your own.
  - You use some extra time checking the map and printing it out
  - But then you save time because you don't get lost and backtrack
- In our case, we'll map the whole "search space" and *then* find the best alignment

Dynamic Programming

- Don't worry, we're not talking about computer programming here
- "Programming" in this case means "optimization" or "planning ahead"
- Like checking Google Maps *before* you try to find your way on your own.
  - You use some extra time checking the map and printing it out
    - But then you save time because you don't get lost and backtrack
- In our case, we'll map the whole "search space" and *then* find the best alignment

Dynamic Programming

- Don't worry, we're not talking about computer programming here
- "Programming" in this case means "optimization" or "planning ahead"
- Like checking Google Maps *before* you try to find your way on your own.
  - You use some extra time checking the map and printing it out
  - But then you save time because you don't get lost and backtrack
- In our case, we'll map the whole "search space" and *then* find the best alignment

Dynamic Programming

- Don't worry, we're not talking about computer programming here
- "Programming" in this case means "optimization" or "planning ahead"
- Like checking Google Maps *before* you try to find your way on your own.
  - You use some extra time checking the map and printing it out
  - But then you save time because you don't get lost and backtrack
- In our case, we'll map the whole "search space" and *then* find the best alignment

Smith-Waterman Algorithm (1981)

Local Alignment of Subsequences

Algorithm

The two molecular sequences will be $\underset{\sim}{\mathrm{A}} = \mathrm{a}_1\mathrm{a}_2 \ldots \mathrm{a}_n$ and $\underset{\sim}{\mathrm{B}} = \mathrm{b}_1\mathrm{b}_2 \ldots \mathrm{b}_m$. A similarity $s(\mathrm{a},\mathrm{b})$ is given between sequence elements a and b. Deletions of length k are given weight W_k. To find pairs of segments with high degrees of similarity, we set up a matrix H. First set

$$H_{k0} = H_{0l} = 0 \text{ for } 0 \leq k \leq n \text{ and } 0 \leq l \leq m.$$

Preliminary values of H have the interpretation that H_{ij} is the maximum similarity of two segments *ending* in a_i and b_j, respectively. These values are obtained from the relationship

$$H_{ij} = \max\{H_{i-1,j-1} + s(\mathrm{a}_i,\mathrm{b}_j), \max_{k \geq 1}\{H_{i-k,j} - W_k\}, \max_{l \geq 1}\{H_{i,j-l} - W_l\}, 0\}, \quad (1)$$

$1 \leq i \leq n$ and $1 \leq j \leq m$.

Smith-Waterman Algorithm (1981)

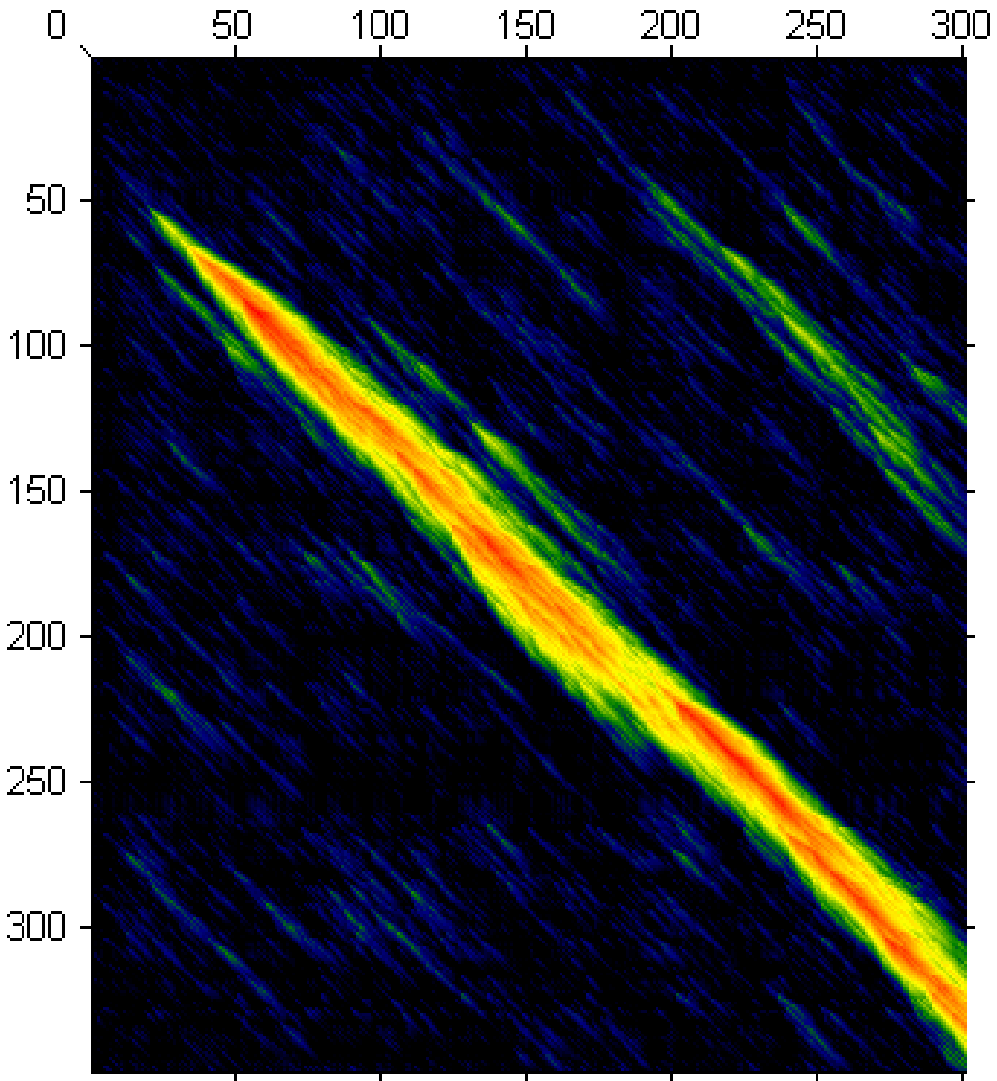
Local Alignment of Subsequences

Just kidding! I have an interactive demo instead.

http://zucker.limbio-paris13.org/COURS/M1S1-SMBH/Cours2baba.html

Homologous Ridges

What a filled-in SW matrix looks like



Scoring Matrices

A Dash of Biological Significance; or, Not All Mutations are Created Equal

- Not all mutations are equally likely
  - Some mutations are acceptable
    - Ser $\leftrightarrow$ Thr, Trp $\leftrightarrow$ Phe, Val $\leftrightarrow$ Ile
  - Some mutations are disruptive
    - Leu $\leftrightarrow$ Asp, Val $\leftrightarrow$ Arg, Tyr $\leftrightarrow$ Leu
- We can quantify the likelihood of all possible mutations using a scoring matrix

Scoring Matrices

A Dash of Biological Significance; or, Not All Mutations are Created Equal

- Not all mutations are equally likely
  - Some mutations are acceptable
    - Ser $\leftrightarrow$ Thr, Trp $\leftrightarrow$ Phe, Val $\leftrightarrow$ Ile
  - Some mutations are disruptive
    - Leu $\leftrightarrow$ Asp, Val $\leftrightarrow$ Arg, Tyr $\leftrightarrow$ Leu
- We can quantify the likelihood of all possible mutations using a scoring matrix

Scoring Matrices

A Dash of Biological Significance; or, Not All Mutations are Created Equal

- Not all mutations are equally likely
  - Some mutations are acceptable
    - Ser $\leftrightarrow$ Thr, Trp $\leftrightarrow$ Phe, Val $\leftrightarrow$ Ile
  - Some mutations are disruptive
    - Leu $\leftrightarrow$ Asp, Val $\leftrightarrow$ Arg, Tyr $\leftrightarrow$ Leu
- We can quantify the likelihood of all possible mutations using a scoring matrix

Scoring Matrices

A Dash of Biological Significance; or, Not All Mutations are Created Equal

- Not all mutations are equally likely
  - Some mutations are acceptable
    - Ser $\leftrightarrow$ Thr, Trp $\leftrightarrow$ Phe, Val $\leftrightarrow$ Ile
  - Some mutations are disruptive
    - Leu $\leftrightarrow$ Asp, Val $\leftrightarrow$ Arg, Tyr $\leftrightarrow$ Leu
- We can quantify the likelihood of all possible mutations using a scoring matrix

Blosum62
Scoring Matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	
A	5	-2	-2	-2	0	0	0	0	-2	-2	-3	-2	-1	-2	0	0	0	-2	-3	0	A
R		5	-2	-3	-3	0	-1	-2	0	-3	-4	1	-3	-3	-2	-2	0	0	-3	-4	R
N			5	0	0	0	-2	0	0	-4	-5	-2	-3	-3	-2	0	0	-2	-2	-5	N
D				5	-4	0	1	-1	0	-5	-6	-3	-4	-4	0	-2	-2	-2	-2	-5	D
C					8	-2	-3	-1	-1	0	-2	-3	0	-1	-1	1	0	0	-2	0	C
Q						5	2	0	0	-2	-4	0	-2	-3	0	0	0	0	-2	-3	Q
E							5	0	0	-3	-4	0	-3	-3	0	0	0	-2	-3	-3	E
G								6	0	-4	-5	-2	-3	-2	-2	0	0	0	-2	-3	G
H									6	-3	-4	0	-2	0	0	0	0	0	2	-2	H
I										4	0	-3	2	0	-2	-3	0	0	-3	2	I
L											4	-4	0	0	-3	-4	-3	0	-4	0	L
K												4	-2	-4	-1	-2	0	0	-3	-4	K
M													6	0	-3	-3	-2	0	-3	2	M
F														6	-3	-2	-2	2	2	0	F
P															7	0	0	-2	-3	0	P
S																4	2	-2	-2	-3	S
T																	5	-1	-3	0	T
W																		9	2	-1	W
Y																			7	-3	Y
V																				4	V

Outline

“Heuristic” Alignment

What does that mean?

- Unlike optimal algorithms, heuristic algorithms don’t guarantee anything
  - No mathematical proof that says “this algorithm always works”
- You “usually” get “pretty good” results
- in practice, this is good enough

“Heuristic” Alignment

What does that mean?

- Unlike optimal algorithms, heuristic algorithms don’t guarantee anything
  - No mathematical proof that says “this algorithm always works”
- You “usually” get “pretty good” results
- in practice, this is good enough

Why settle?

Why settle for “pretty good” when we can have optimal?

- Because it’s faster and requires less memory!
- Example: aligning two entire genomes (e.g. mouse & human)
  - With Smith-Waterman, this would take about 40 exabytes
    - (That’s 40 billion gigabytes)
- For Comparison:
  - You’re lucky to have 4 GB of memory in your PC/laptop
  - Large servers might have 400 GB
  - You’d still need several million servers to do the full alignment
  - It would probably take years to complete
- Using BLAST, this could probably be done on an average PC in a few days.

Why settle?

Why settle for “pretty good” when we can have optimal?

- Because it’s faster and requires less memory!
- Example: aligning two entire genomes (e.g. mouse & human)
  - With Smith-Waterman, this would take about 40 exabytes
    - (That’s 40 billion gigabytes)
- For Comparison:
  - You’re lucky to have 4 GB of memory in your PC/laptop
  - Large servers might have 400 GB
  - You’d still need several million servers to do the full alignment
  - It would probably take years to complete
- Using BLAST, this could probably be done on an average PC in a few days.

Why settle?

Why settle for “pretty good” when we can have optimal?

- Because it’s faster and requires less memory!
- Example: aligning two entire genomes (e.g. mouse & human)
  - With Smith-Waterman, this would take about 40 exabytes
    - (That’s 40 billion gigabytes)
- For Comparison:
  - You’re lucky to have 4 GB of memory in your PC/laptop
  - Large servers might have 400 GB
  - You’d still need several million servers to do the full alignment
  - It would probably take years to complete
- Using BLAST, this could probably be done on an average PC in a few days.

Why settle?

Why settle for “pretty good” when we can have optimal?

- Because it’s faster and requires less memory!
- Example: aligning two entire genomes (e.g. mouse & human)
  - With Smith-Waterman, this would take about 40 exabytes
    - (That’s 40 billion gigabytes)
- For Comparison:
  - You’re lucky to have 4 GB of memory in your PC/laptop
  - Large servers might have 400 GB
  - You’d still need several million servers to do the full alignment
  - It would probably take years to complete
- Using BLAST, this could probably be done on an average PC in a few days.

Why settle?

Why settle for “pretty good” when we can have optimal?

- Because it’s faster and requires less memory!
- Example: aligning two entire genomes (e.g. mouse & human)
  - With Smith-Waterman, this would take about 40 exabytes
    - (That’s 40 billion gigabytes)
- For Comparison:
  - You’re lucky to have 4 GB of memory in your PC/laptop
  - Large servers might have 400 GB
  - You’d still need several million servers to do the full alignment
  - It would probably take years to complete
- Using BLAST, this could probably be done on an average PC in a few days.

Why so slow?

- Smith-Waterman is too slow because it computes the entire $n \times m$ "search space"
- If you double the length of each sequence, the search space is quadrupled
- To go faster, we want to efficiently narrow our search and only do full Smith-Waterman alignments in small areas

Why so slow?

- Smith-Waterman is too slow because it computes the entire $n \times m$ "search space"
- If you double the length of each sequence, the search space is quadrupled
- To go faster, we want to efficiently narrow our search and only do full Smith-Waterman alignments in small areas

Why so slow?

- Smith-Waterman is too slow because it computes the entire $n \times m$ "search space"
- If you double the length of each sequence, the search space is quadrupled
- To go faster, we want to efficiently narrow our search and only do full Smith-Waterman alignments in small areas

FASTA (1988)

William R. Pearson, Dept. of Biochemistry, U. Va. and David J. Lipman, NIH

- Faster than Smith-Waterman, slower than BLAST
- Pearson's goal is not pure speed, but the best tradeoff of sensitivity, selectivity, and speed
- designed to find distantly divergent but related sequences

FASTA (1988)

William R. Pearson, Dept. of Biochemistry, U. Va. and David J. Lipman, NIH

- Faster than Smith-Waterman, slower than BLAST
- Pearson's goal is not pure speed, but the best tradeoff of sensitivity, selectivity, and speed
- designed to find distantly divergent but related sequences

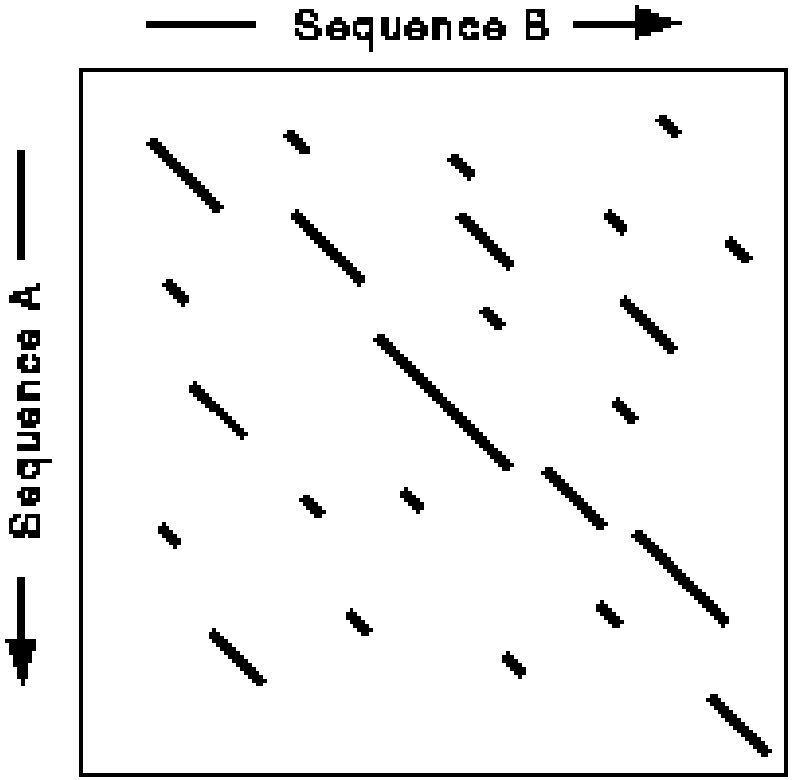
FASTA (1988)

William R. Pearson, Dept. of Biochemistry, U. Va. and David J. Lipman, NIH

- Faster than Smith-Waterman, slower than BLAST
- Pearson's goal is not pure speed, but the best tradeoff of sensitivity, selectivity, and speed
- designed to find distantly divergent but related sequences

FASTA
The Algorithm

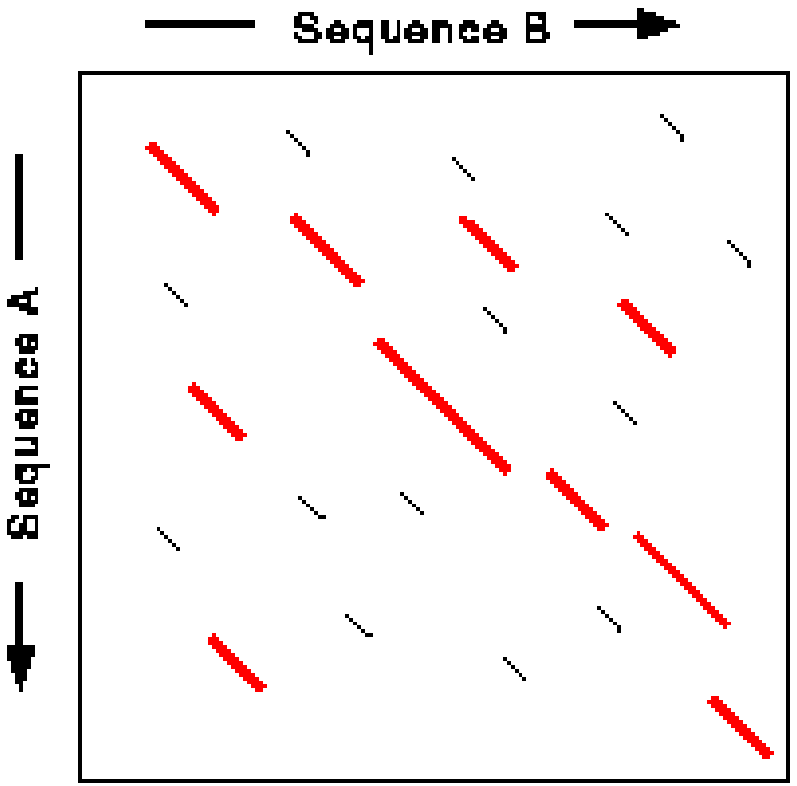
- Quickly scan for identical stretches
- Rescore each stretch using a scoring matrix
- Keep only the top ten
- Join nearby segments with appropriate gap penalties
- Keep only segments that join with the top scoring segment
- Optimize alignment with banded Smith-Waterman



Find runs of identitical words

FASTA
The Algorithm

- Quickly scan for identical stretches
- Rescore each stretch using a scoring matrix
- Keep only the top ten
- Join nearby segments with appropriate gap penalties
- Keep only segments that join with the top scoring segment
- Optimize alignment with banded Smith-Waterman

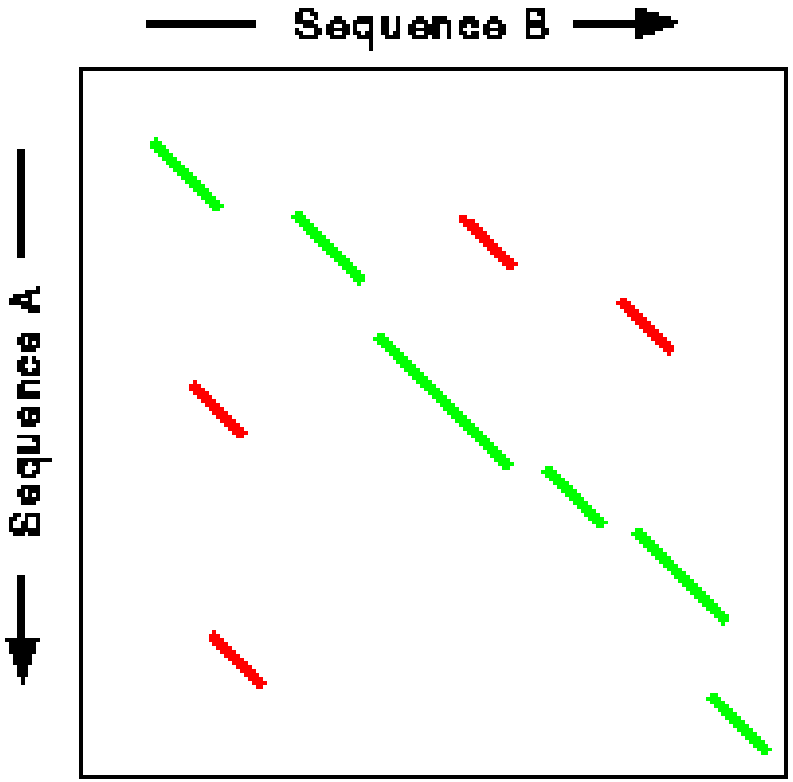


Re-score using PAM matrix

Keep top scoring segments

FASTA
The Algorithm

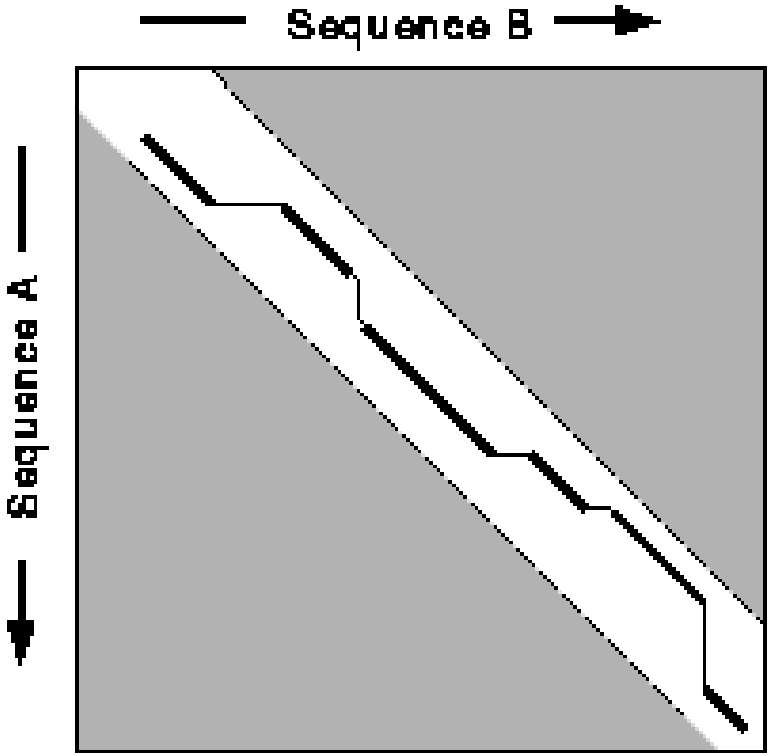
- Quickly scan for identical stretches
- Rescore each stretch using a scoring matrix
- Keep only the top ten
- Join nearby segments with appropriate gap penalties
- Keep only segments that join with the top scoring segment
- Optimize alignment with banded Smith-Waterman



Join segments using gaps, eliminate other segments

FASTA
The Algorithm

- Quickly scan for identical stretches
- Rescore each stretch using a scoring matrix
- Keep only the top ten
- Join nearby segments with appropriate gap penalties
- Keep only segments that join with the top scoring segment
- Optimize alignment with banded Smith-Waterman



Use dynamic programming to create an optimal alignment

BLAST!

Definitely a backronym

BLAST!

Definitely a backronym

BLAST!

Definitely a backronym

Wait, no. That's not it.

BLAST!

Definitely a backronym

BLAST!

Definitely a backronym

That's not it either.

BLAST!

Definitely a backronym

BLAST!

Definitely a backronym

What? No!

BLAST!

Definitely a backronym

BLAST!

Definitely a backronym

Nope. Wrong year.

That *was* an awesome show, though.

BLAST!

Definitely a backronym

Nope. Wrong year.

That *was* an awesome show, though.

BLAST (1990)

Altschul et. al.

- Compile a list of "words" from the query sequence
  - Example: PVAKEPIK...
  - Words: PVA, VAK, AKE, KEP, EPI, PIK, ...
- Search for all these small words in the database
  - words are short and have equal length
  - no gaps allowed
  - this allows major optimization
- Throw out any matches below a threshold score
- "Venus flytrap" selection: only consider pairs of nearby matches
- Finally, use Smith-Waterman to locally extend selected matches only
- Search space is reduced to $\sim n + m$ instead of $n \times m$

BLAST (1990)

Altschul et. al.

- Compile a list of "words" from the query sequence
  - Example: PVAKEPIK...
  - Words: PVA, VAK, AKE, KEP, EPI, PIK, ...
- Search for all these small words in the database
  - words are short and have equal length
  - no gaps allowed
  - this allows major optimization
- Throw out any matches below a threshold score
- "Venus flytrap" selection: only consider pairs of nearby matches
- Finally, use Smith-Waterman to locally extend selected matches only
- Search space is reduced to $\sim n + m$ instead of $n \times m$

BLAST (1990)

Altschul et. al.

- Compile a list of “words” from the query sequence
  - Example: PVAKEPIK...
  - Words: PVA, VAK, AKE, KEP, EPI, PIK, ...
- Search for all these small words in the database
  - words are short and have equal length
  - no gaps allowed
  - this allows major optimization
- Throw out any matches below a threshold score
- “Venus flytrap” selection: only consider pairs of nearby matches
- Finally, use Smith-Waterman to locally extend selected matches only
- Search space is reduced to $\sim n + m$ instead of $n \times m$

BLAST (1990)

Altschul et. al.

- Compile a list of "words" from the query sequence
  - Example: PVAKEPIK...
  - Words: PVA, VAK, AKE, KEP, EPI, PIK, ...
- Search for all these small words in the database
  - words are short and have equal length
  - no gaps allowed
  - this allows major optimization
- Throw out any matches below a threshold score
- "Venus flytrap" selection: only consider pairs of nearby matches
- Finally, use Smith-Waterman to locally extend selected matches only
- Search space is reduced to $\sim n + m$ instead of $n \times m$

BLAST (1990)

Altschul et. al.

- Compile a list of "words" from the query sequence
  - Example: PVAKEPIK...
  - Words: PVA, VAK, AKE, KEP, EPI, PIK, ...
- Search for all these small words in the database
  - words are short and have equal length
  - no gaps allowed
  - this allows major optimization
- Throw out any matches below a threshold score
- "Venus flytrap" selection: only consider pairs of nearby matches
- Finally, use Smith-Waterman to locally extend selected matches only
- Search space is reduced to $\sim n + m$ instead of $n \times m$

BLAST (1990)

Altschul et. al.

- Compile a list of "words" from the query sequence
  - Example: PVAKEPIK...
  - Words: PVA, VAK, AKE, KEP, EPI, PIK, ...
- Search for all these small words in the database
  - words are short and have equal length
  - no gaps allowed
  - this allows major optimization
- Throw out any matches below a threshold score
- "Venus flytrap" selection: only consider pairs of nearby matches
- Finally, use Smith-Waterman to locally extend selected matches only
- Search space is reduced to $\sim n + m$ instead of $n \times m$

E-Values
Expecting the Unexpected

- BLAST reports an E-value for each alignment
  - stands for “expect”
- This score is effectively a false-discovery rate
  - “How often would a random alignment score as high as this alignment?”
  - If *false* discovery rate is very low, then this alignment is probably a *true* positive
- Example: suppose an alignment had an E-value of $E = 1 \times 10^{-10}$
  - Then we would expect one out of every $10,000,000,000$ random searches to yield a result as good as this alignment
- For those who prefer p-values: when $E < 0.01$, $P \approx E$

E-Values
Expecting the Unexpected

- BLAST reports an E-value for each alignment
  - stands for “expect”
- This score is effectively a false-discovery rate
  - “How often would a random alignment score as high as this alignment?”
  - If *false* discovery rate is very low, then this alignment is probably a *true* positive
- Example: suppose an alignment had an E-value of $E = 1 \times 10^{-10}$
  - Then we would expect one out of every $10,000,000,000$ random searches to yield a result as good as this alignment
- For those who prefer p-values: when $E < 0.01$, $P \approx E$

E-Values
Expecting the Unexpected

- BLAST reports an E-value for each alignment
  - stands for “expect”
- This score is effectively a false-discovery rate
  - “How often would a random alignment score as high as this alignment?”
  - If *false* discovery rate is very low, then this alignment is probably a *true* positive
- Example: suppose an alignment had an E-value of $E = 1 \times 10^{-10}$
  - Then we would expect one out of every $10,000,000,000$ random searches to yield a result as good as this alignment
- For those who prefer p-values: when $E < 0.01$, $P \approx E$

E-Values
Expecting the Unexpected

- BLAST reports an E-value for each alignment
  - stands for “expect”
- This score is effectively a false-discovery rate
  - “How often would a random alignment score as high as this alignment?”
  - If *false* discovery rate is very low, then this alignment is probably a *true* positive
- Example: suppose an alignment had an E-value of $E = 1 \times 10^{-10}$
  - Then we would expect one out of every $10,000,000,000$ random searches to yield a result as good as this alignment
- For those who prefer p-values: when $E < 0.01$, $P \approx E$

Outline

What Could Possibly Go Wrong?

Nothing, right?

- We are only comparing the *primary* sequence
- We can't easily predict 3D structure from primary sequence
  - That goes for both protein and RNA
- Some proteins with $< 25\%$ sequence identity still fold the same
- RNA folding depends primarily on presence, not identity, of specific base pairs
- Can't predict posttranslational modifications
- Ultimately, primary sequence homology is not a guarantee of actual relatedness
  - but it's pretty good most of the time

What Could Possibly Go Wrong?

Nothing, right?

- We are only comparing the *primary* sequence
- We can't easily predict 3D structure from primary sequence
  - That goes for both protein and RNA
- Some proteins with $< 25\%$ sequence identity still fold the same
- RNA folding depends primarily on presence, not identity, of specific base pairs
- Can't predict posttranslational modifications
- Ultimately, primary sequence homology is not a guarantee of actual relatedness
  - but it's pretty good most of the time

What Could Possibly Go Wrong?

Nothing, right?

- We are only comparing the *primary* sequence
- We can't easily predict 3D structure from primary sequence
  - That goes for both protein and RNA
- Some proteins with $< 25\%$ sequence identity still fold the same
- RNA folding depends primarily on presence, not identity, of specific base pairs
- Can't predict posttranslational modifications
- Ultimately, primary sequence homology is not a guarantee of actual relatedness
  - but it's pretty good most of the time

What Could Possibly Go Wrong?

Nothing, right?

- We are only comparing the *primary* sequence
- We can't easily predict 3D structure from primary sequence
  - That goes for both protein and RNA
- Some proteins with $< 25\%$ sequence identity still fold the same
- RNA folding depends primarily on presence, not identity, of specific base pairs
- Can't predict posttranslational modifications
- Ultimately, primary sequence homology is not a guarantee of actual relatedness
  - but it's pretty good most of the time

What Could Possibly Go Wrong?

Nothing, right?

- We are only comparing the *primary* sequence
- We can't easily predict 3D structure from primary sequence
  - That goes for both protein and RNA
- Some proteins with $< 25\%$ sequence identity still fold the same
- RNA folding depends primarily on presence, not identity, of specific base pairs
- Can't predict posttranslational modifications
- Ultimately, primary sequence homology is not a guarantee of actual relatedness
  - but it's pretty good most of the time

What Could Possibly Go Wrong?

Nothing, right?

- We are only comparing the *primary* sequence
- We can't easily predict 3D structure from primary sequence
  - That goes for both protein and RNA
- Some proteins with $< 25\%$ sequence identity still fold the same
- RNA folding depends primarily on presence, not identity, of specific base pairs
- Can't predict posttranslational modifications
- Ultimately, primary sequence homology is not a guarantee of actual relatedness
  - but it's pretty good most of the time

Outline

What does this button do?