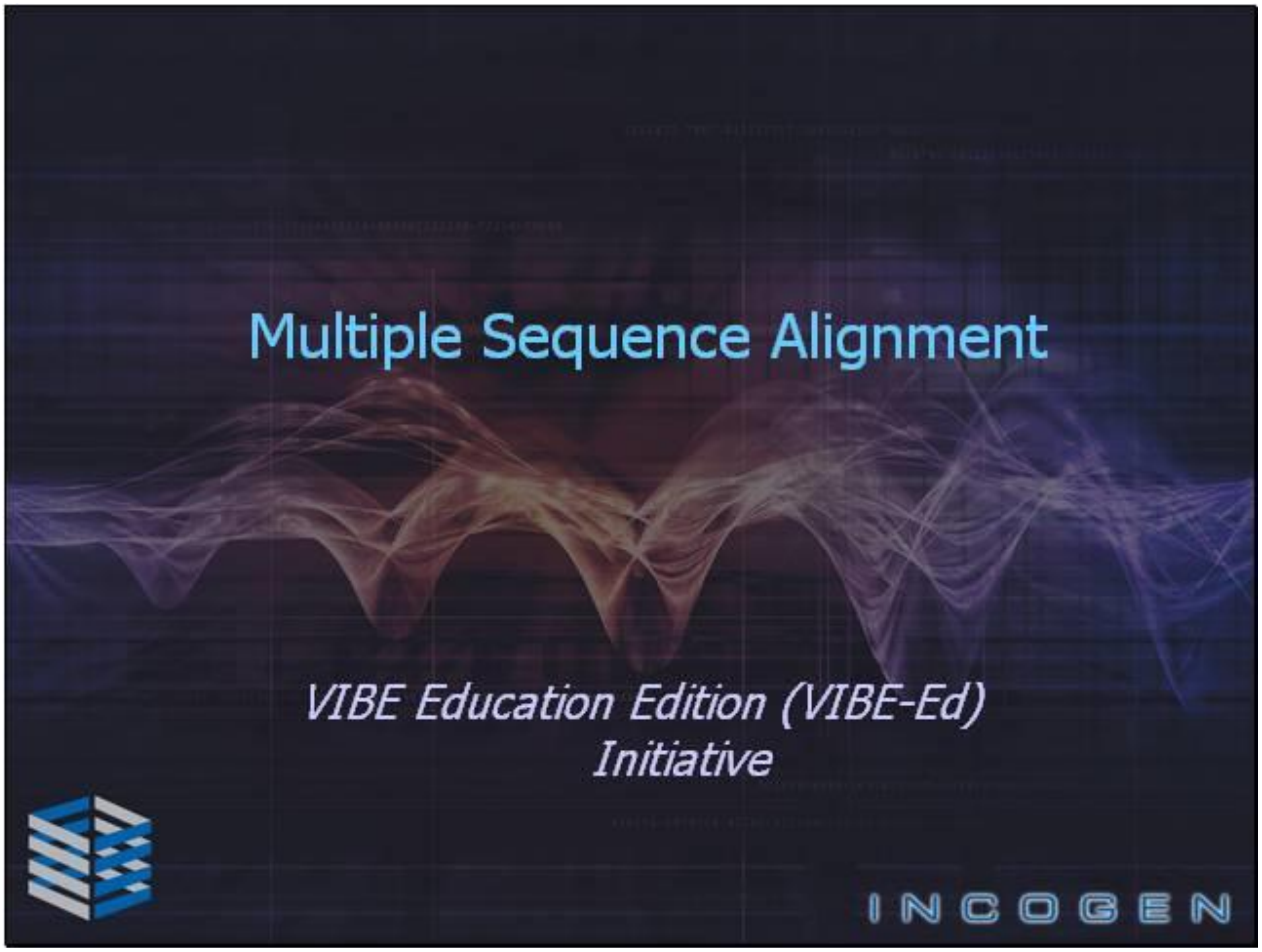
Slide 1 - Multiple Sequence Alignment



Multiple Sequence Alignment

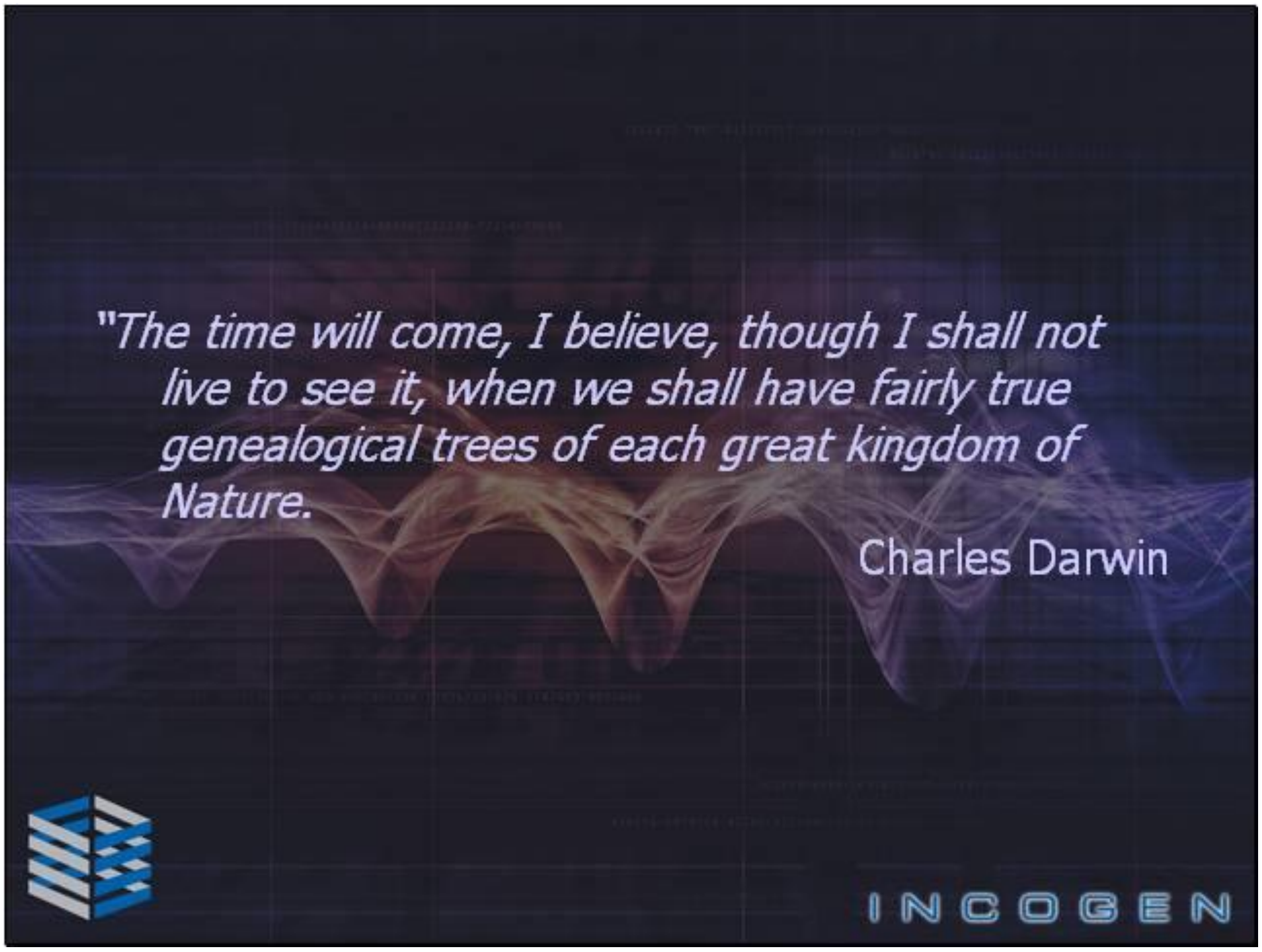
VIBE Education Edition (VIBE-Ed) Initiative

INCOGEN

Slide notes

This presentation will give you some information about multiple sequence alignments.

Slide 2 - Slide 2



"The time will come, I believe, though I shall not live to see it, when we shall have fairly true genealogical trees of each great kingdom of Nature.

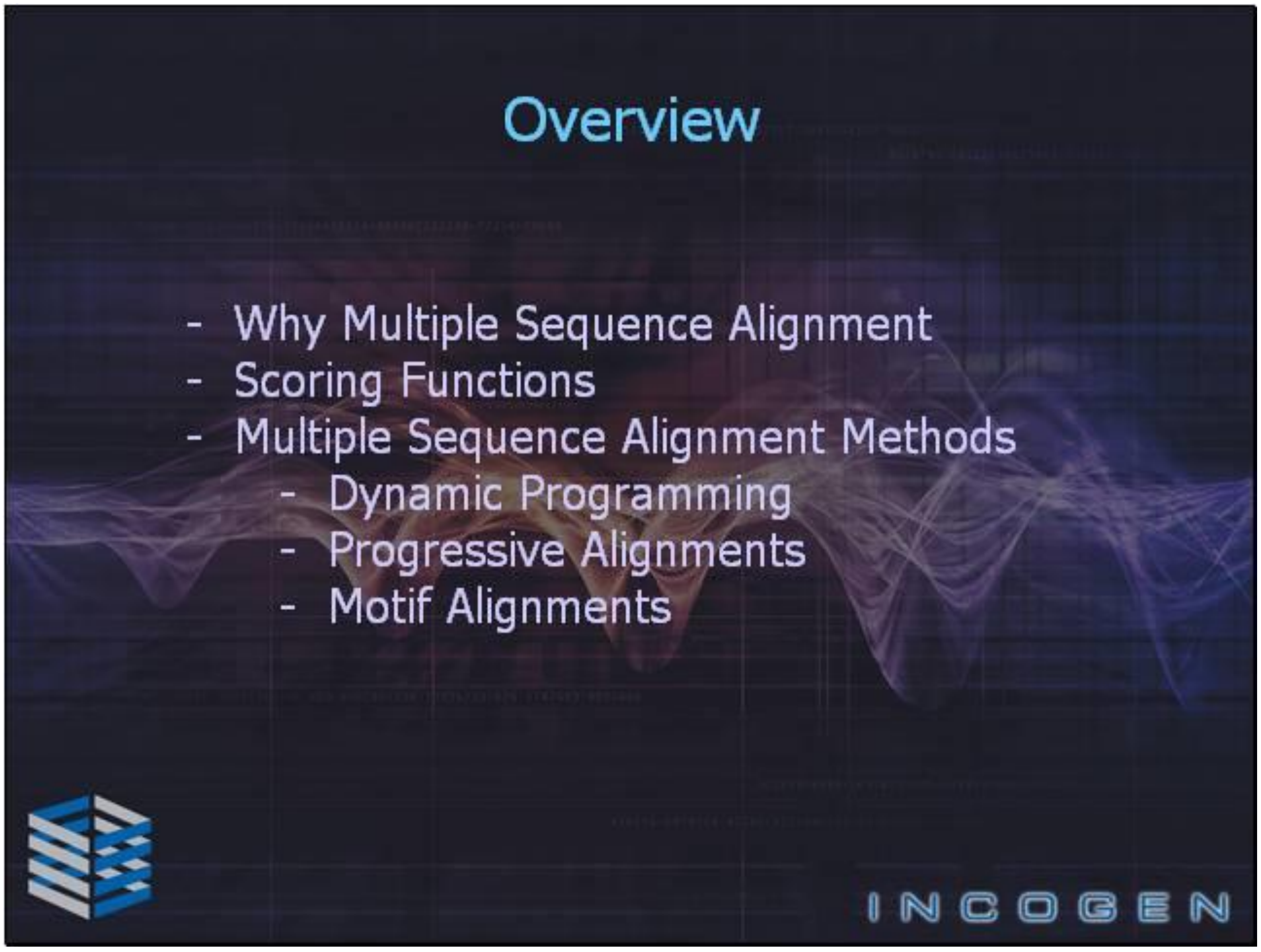
Charles Darwin

INCOGEN

Slide notes

Charles Darwin once said, "The time will come, I believe, though I shall not live to see it, when we shall have fairly true genealogical trees of each great kingdom of Nature."

Slide 3 - Overview



Overview

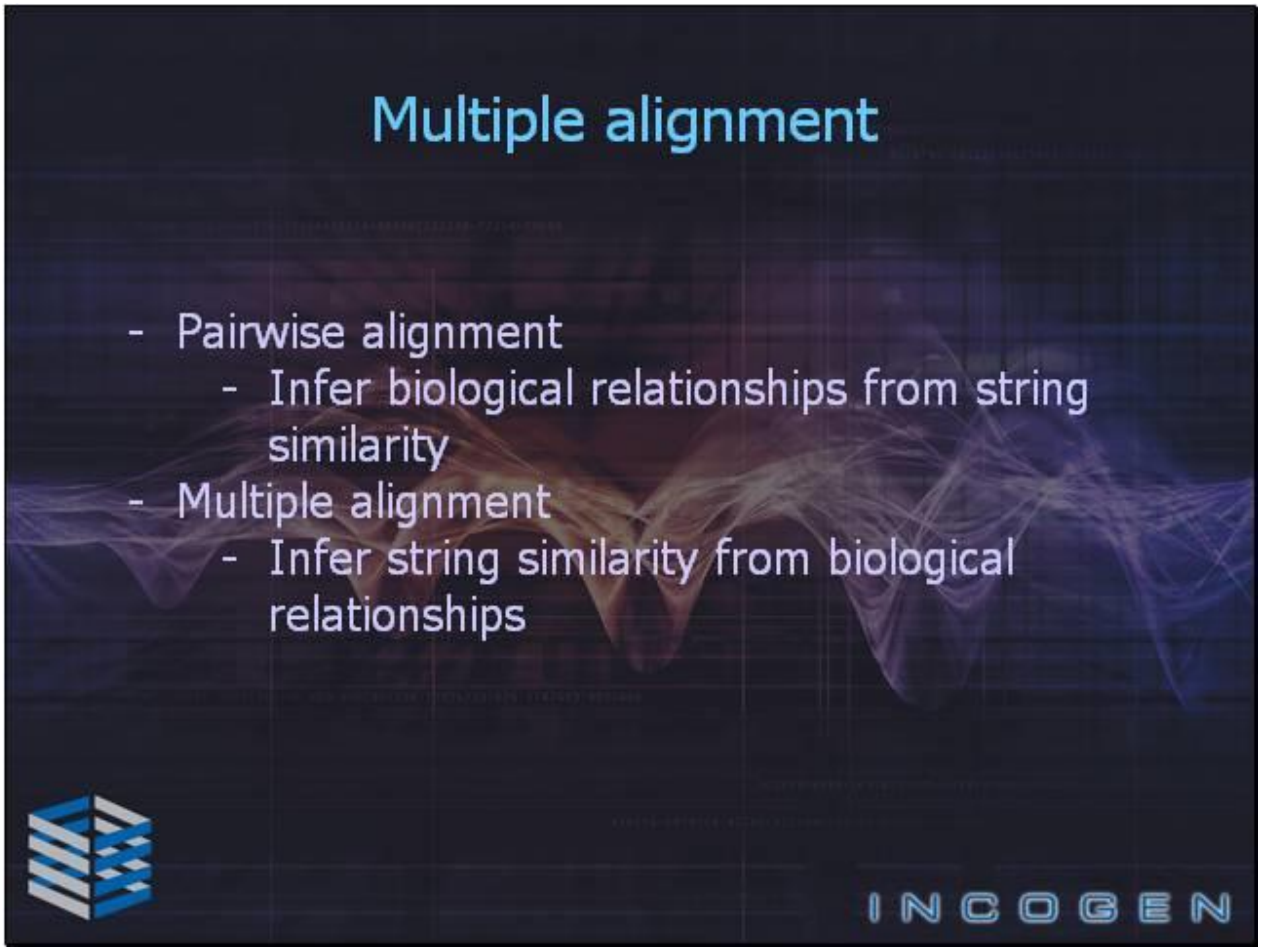
- Why Multiple Sequence Alignment
- Scoring Functions
- Multiple Sequence Alignment Methods
  - Dynamic Programming
  - Progressive Alignments
  - Motif Alignments

INCOGEN

Slide notes

Over the next few minutes, we'll discuss why we need multiple sequence alignments and then how we calculate and score them.

Slide 4 - Multiple alignment



Multiple alignment

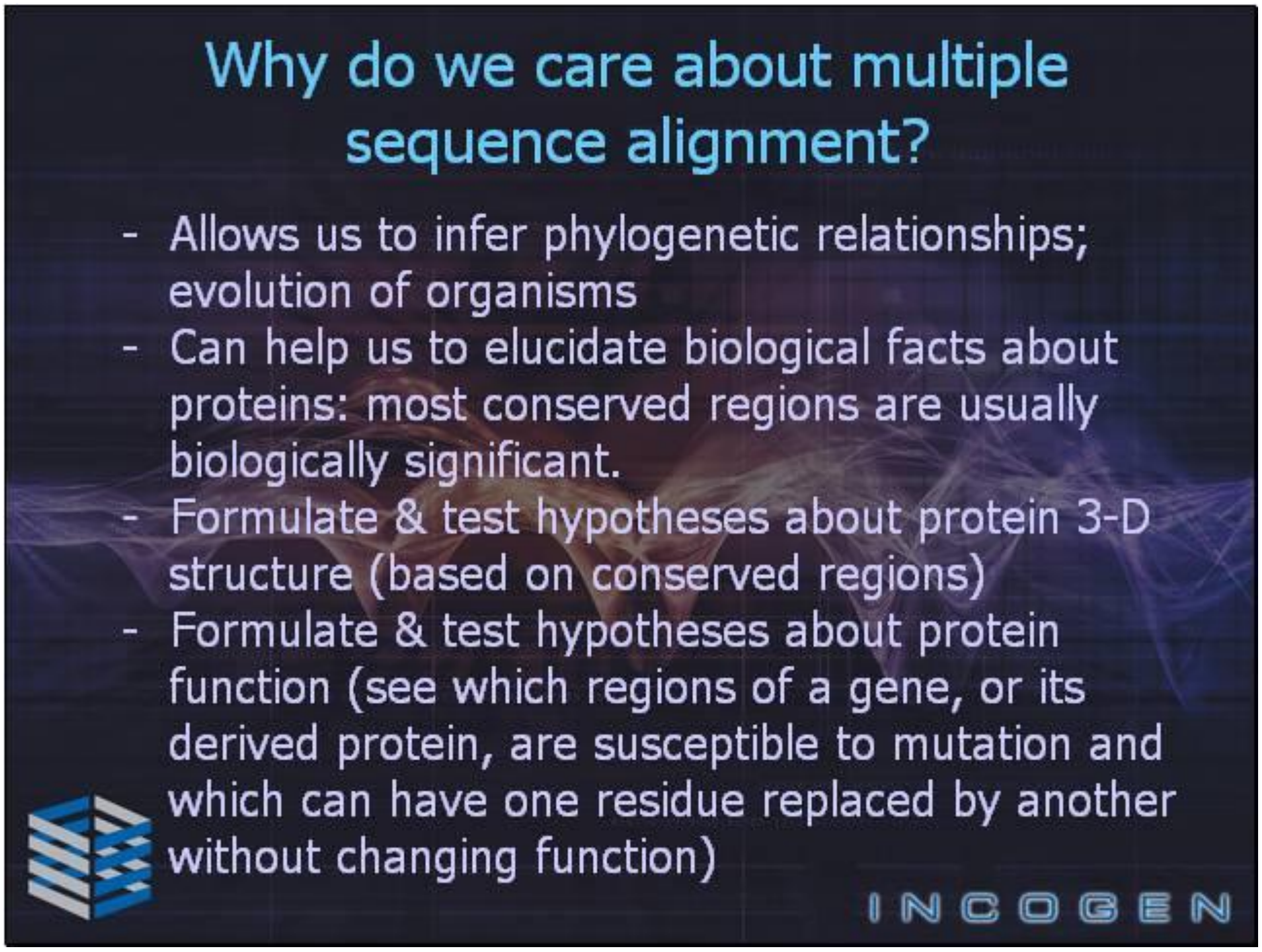
- Pairwise alignment
  - Infer biological relationships from string similarity
- Multiple alignment
  - Infer string similarity from biological relationships

INCOGEN

Slide notes

Why do we need multiple sequence alignments? What information can we learn from them that we can't learn from pairwise sequence alignments? Typically with a pairwise alignment, we infer biological relationships from the sequence similarity. With a multiple alignment, we know that the sequences are biologically related, and we use the multiple alignment to find the areas of sequence similarity that could point to the structure of an evolutionary ancestor or provide information about the evolutionary history of the sequences. Multiple sequence alignments, or MSAs, are also more sensitive to sequence similarities than a pairwise alignment because the conserved regions could be so dispersed that a pairwise alignment wouldn't find them.

Slide 5 - Why do we care about multiple sequence alignment?



Why do we care about multiple sequence alignment?

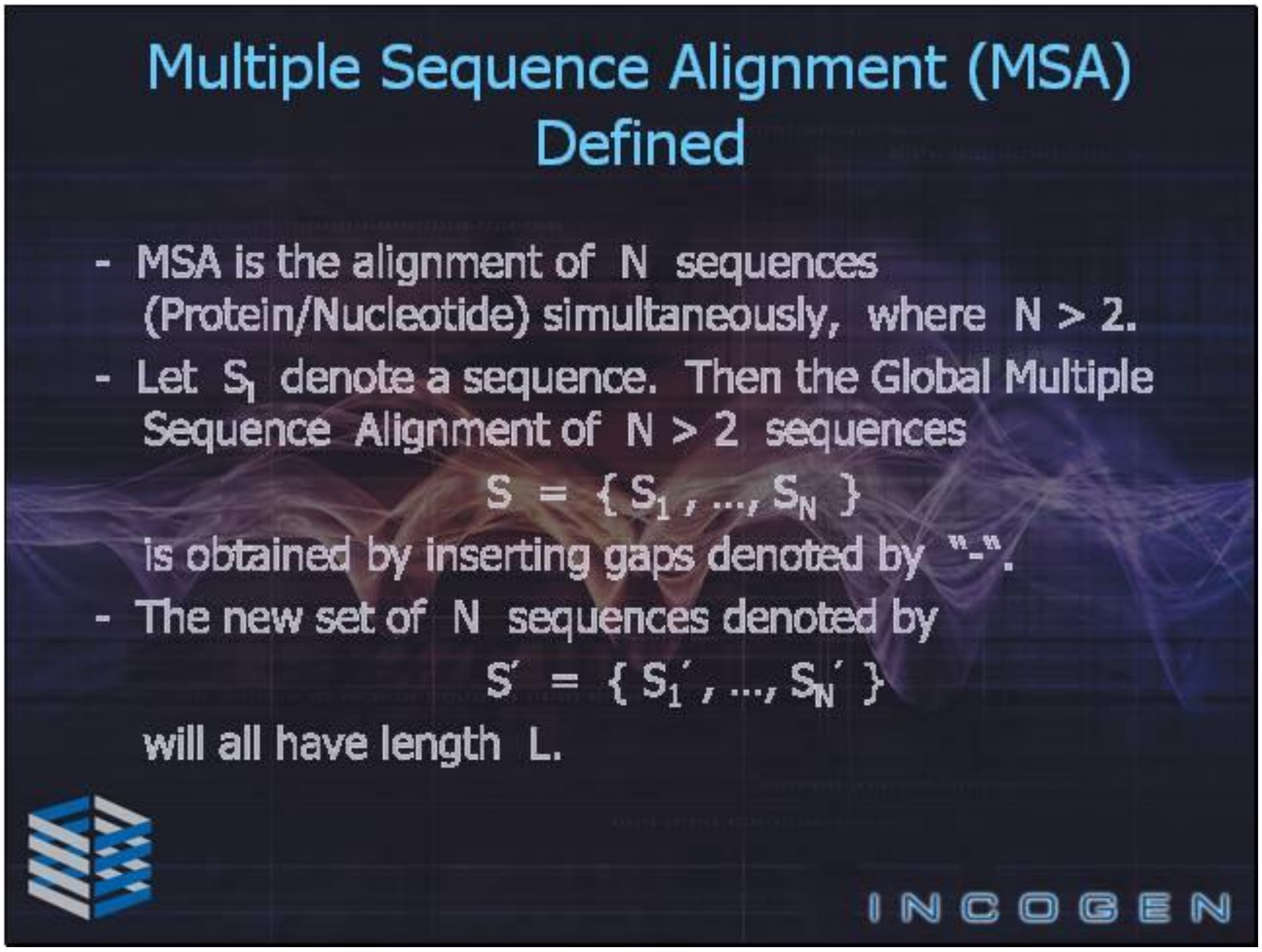
- Allows us to infer phylogenetic relationships; evolution of organisms
- Can help us to elucidate biological facts about proteins: most conserved regions are usually biologically significant.
- Formulate & test hypotheses about protein 3-D structure (based on conserved regions)
- Formulate & test hypotheses about protein function (see which regions of a gene, or its derived protein, are susceptible to mutation and which can have one residue replaced by another without changing function)

INCOGEN

Slide notes

So, MSAs allow us to infer phylogenetic relationships. They can also help us to elucidate biological facts about proteins since most conserved regions are biologically significant. MSAs can also help us to formulate and test hypotheses about protein 3-D structure and function.

Slide 6 - Multiple Sequence Alignment (MSA) Defined



Multiple Sequence Alignment (MSA) Defined

- MSA is the alignment of N sequences (Protein/Nucleotide) simultaneously, where $N > 2$.
- Let S_i denote a sequence. Then the Global Multiple Sequence Alignment of $N > 2$ sequences

$$S = \{ S_1, \ldots, S_N \}$$

is obtained by inserting gaps denoted by "-".

- The new set of N sequences denoted by

$$S' = \{ S_1', \ldots, S_N' \}$$

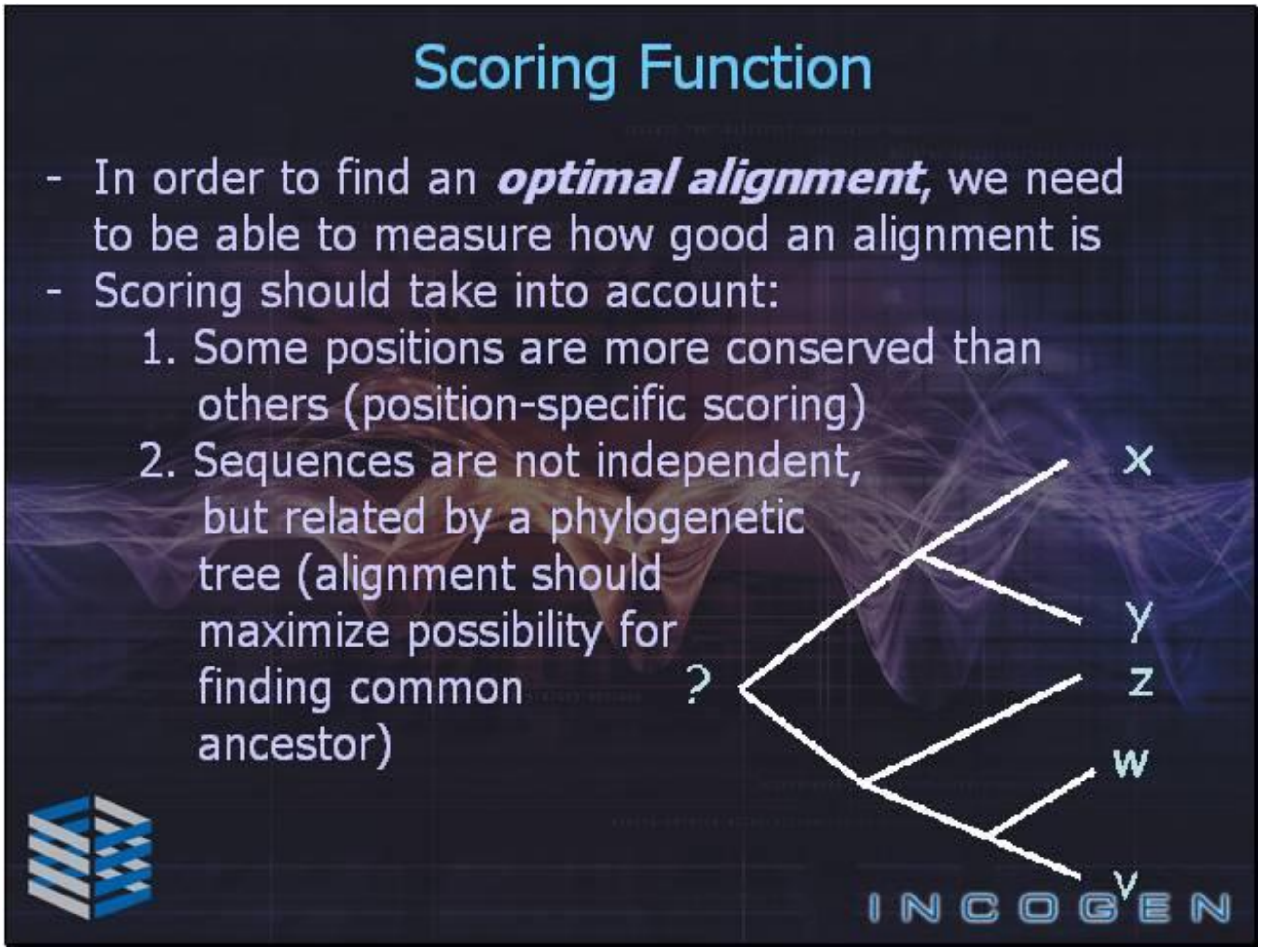
will all have length L.

INCOGEN

Slide notes

An MSA is the alignment of more than two protein or nucleotide sequences. The alignments in an MSAare global, and gaps are added as the sequences are aligned so that all of the sequences have the same length.

Slide 7 - Scoring Function



Scoring Function

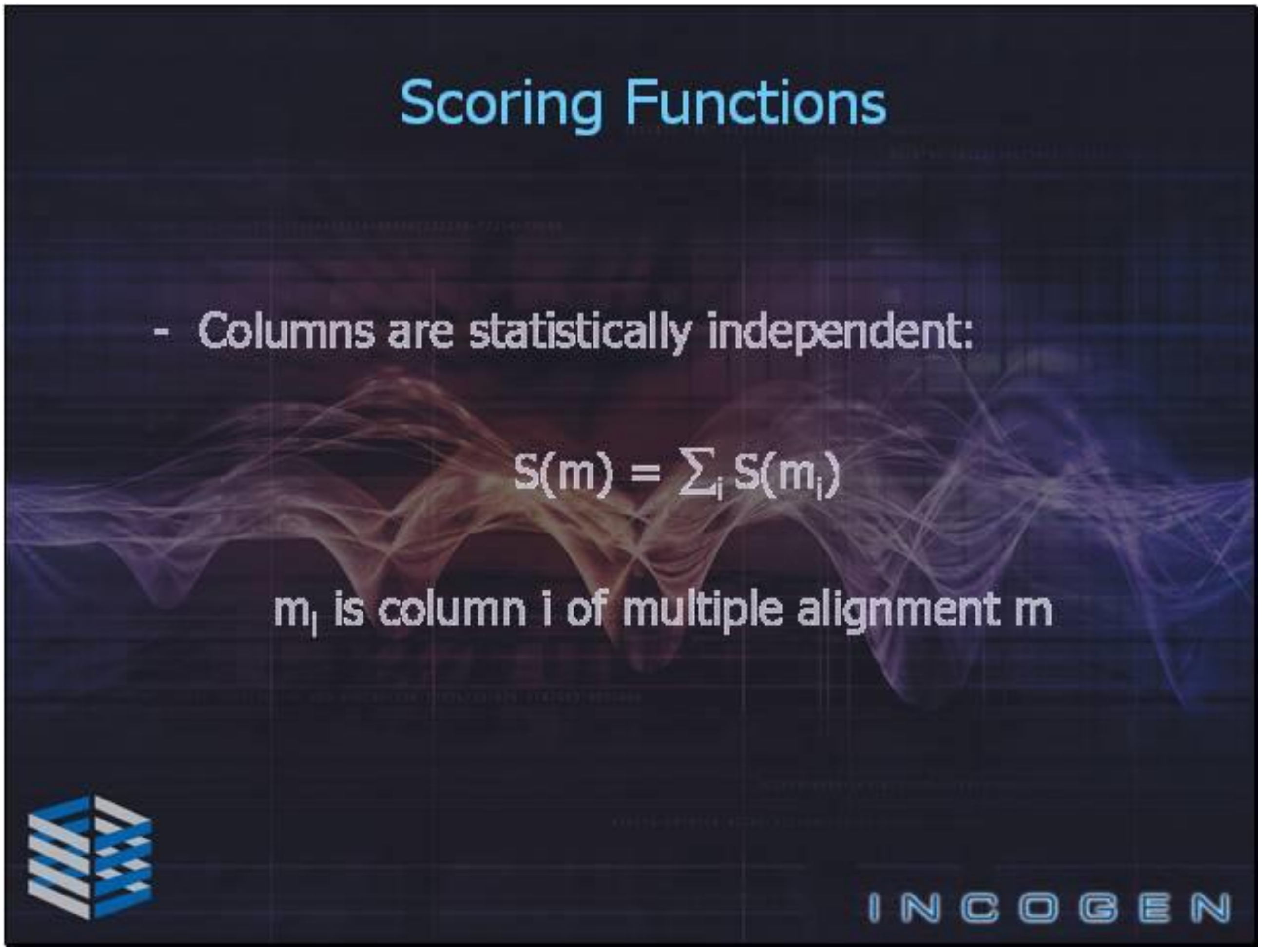
- In order to find an ***optimal alignment***, we need to be able to measure how good an alignment is
- Scoring should take into account:
    1. Some positions are more conserved than others (position-specific scoring)
    2. Sequences are not independent, but related by a phylogenetic tree (alignment should maximize possibility for finding common ancestor)

x
y
z
w
v
?
INCOGEN

Slide notes

In order to score an alignment, we have to be able to quantitatively calculate how good it is. Any scoring algorithm we use needs to take into account that some positions are more conserved than others, which is called position-specific scoring, and that the sequences are biologically related by a phylogenetic tree.

Slide 8 - Scoring Functions



Scoring Functions

- Columns are statistically independent:

$$S(m) = \sum_i S(m_i)$$

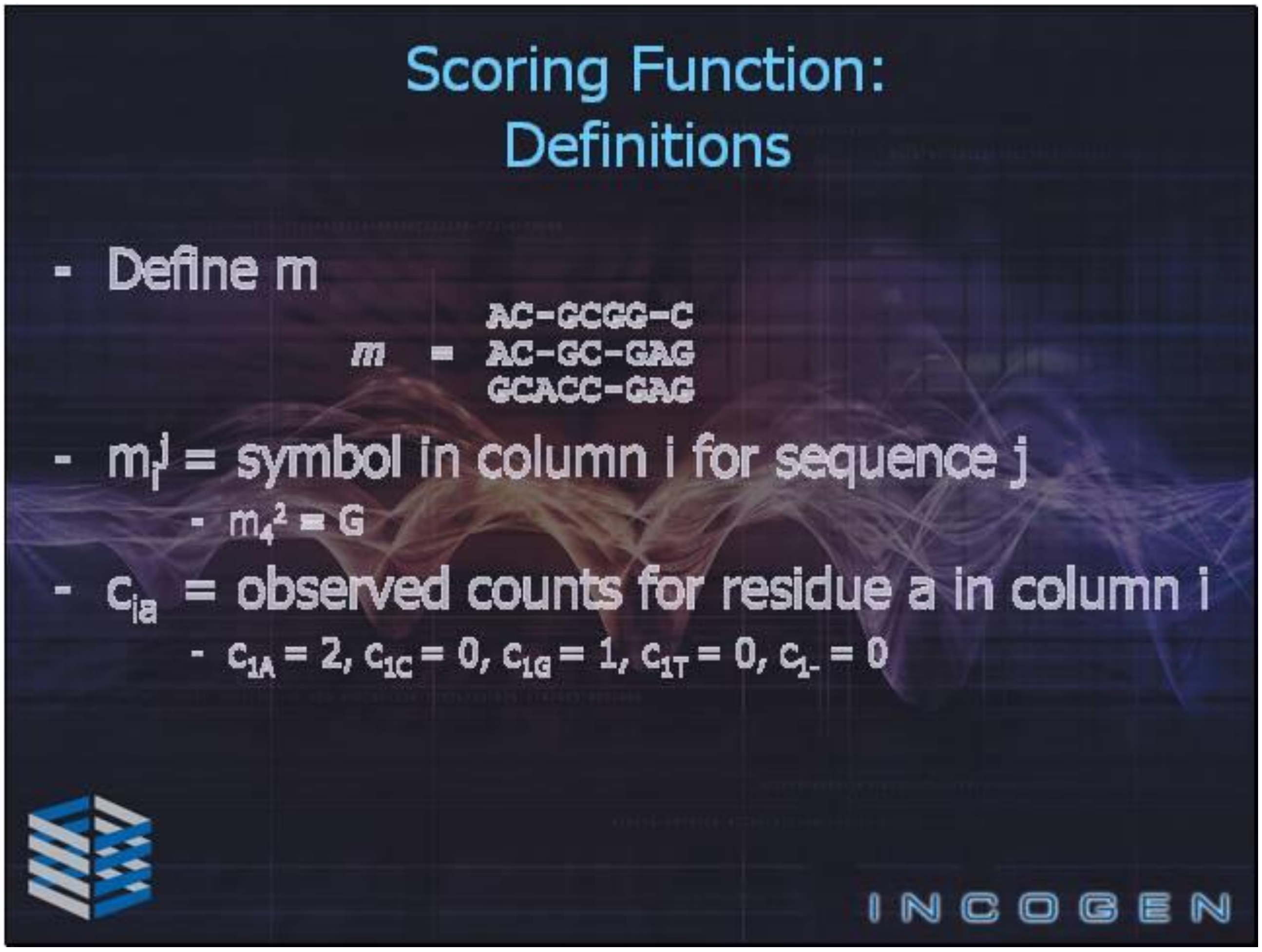
m_i is column i of multiple alignment m

INCOGEN

Slide notes

All columns in the alignment are treated as statistically independent.

Slide 9 - Scoring Function: Definitions



Scoring Function: Definitions

- Define m

$$m = \begin{array}{l}\texttt{AC-GCGG-C}\\ \texttt{AC-GC-GAG}\\ \texttt{GCACC-GAG}\end{array}$$

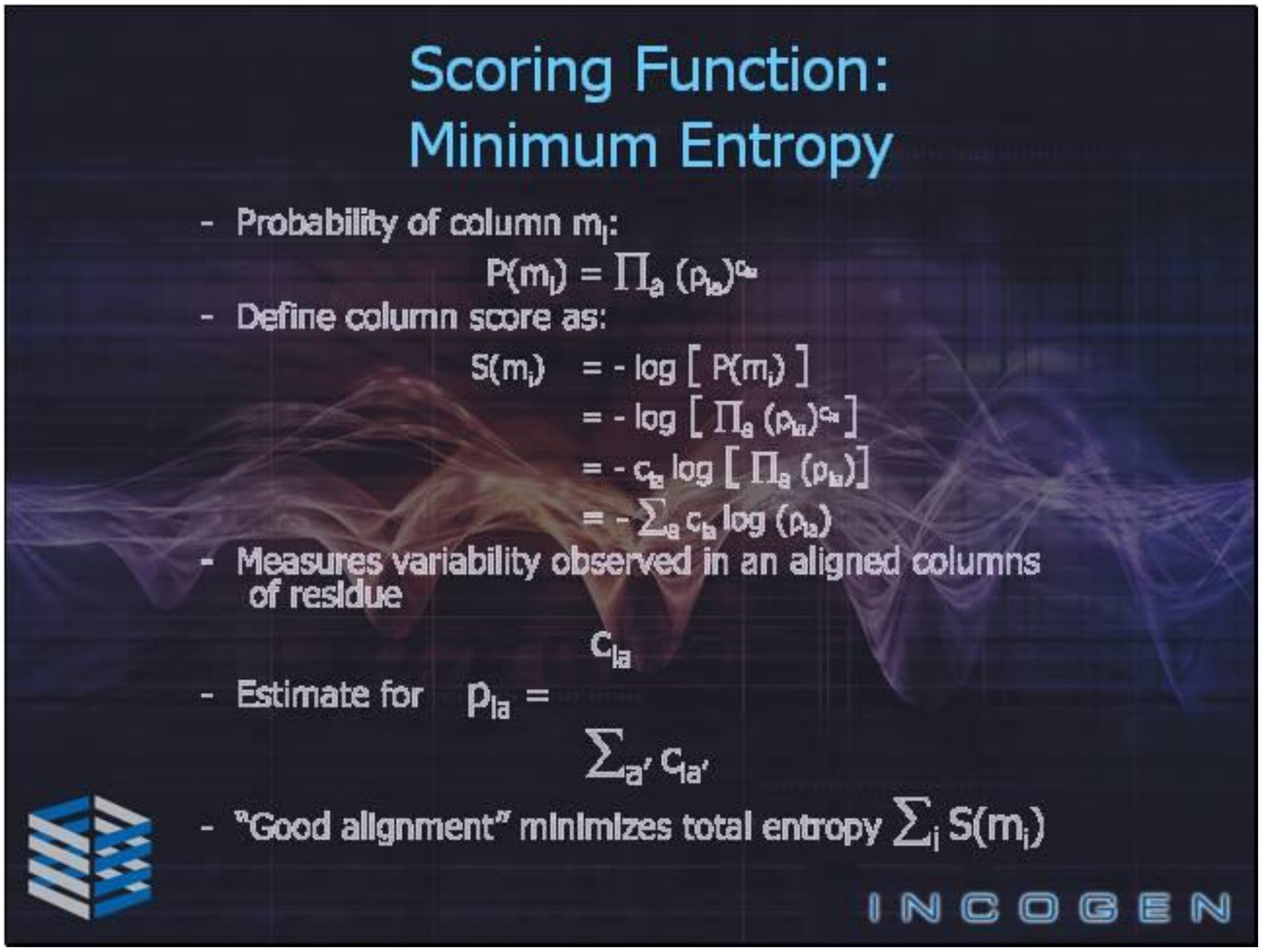
- m_i^j = symbol in column i for sequence j
  - $m_4^2 = G$
- c_{ia} = observed counts for residue a in column i
  - $c_{1A} = 2, c_{1C} = 0, c_{1G} = 1, c_{1T} = 0, c_{1-} = 0$

INCOGEN

Slide notes

Let's go through some terminology first. The alignment is referred to as m. The letter i typically refers to the column number, j refers to the sequence, and a refers to the specific residue. There are different scoring functions that can be used to calculate an MSA.

Slide 10 - Scoring Function: Minimum Entropy



Scoring Function: Minimum Entropy

- Probability of column m_i:

$$P(m_i) = \prod_a (p_{ia})^{c_{ia}}$$

- Define column score as:

$$\begin{aligned} S(m_i) &= -\log [\, P(m_i) \,] \\ &= -\log [\, \prod_a (p_{ia})^{c_{ia}} \,] \\ &= -c_{ia} \log [\, \prod_a (p_{ia})] \\ &= -\sum_a c_{ia} \log (p_{ia}) \end{aligned}$$

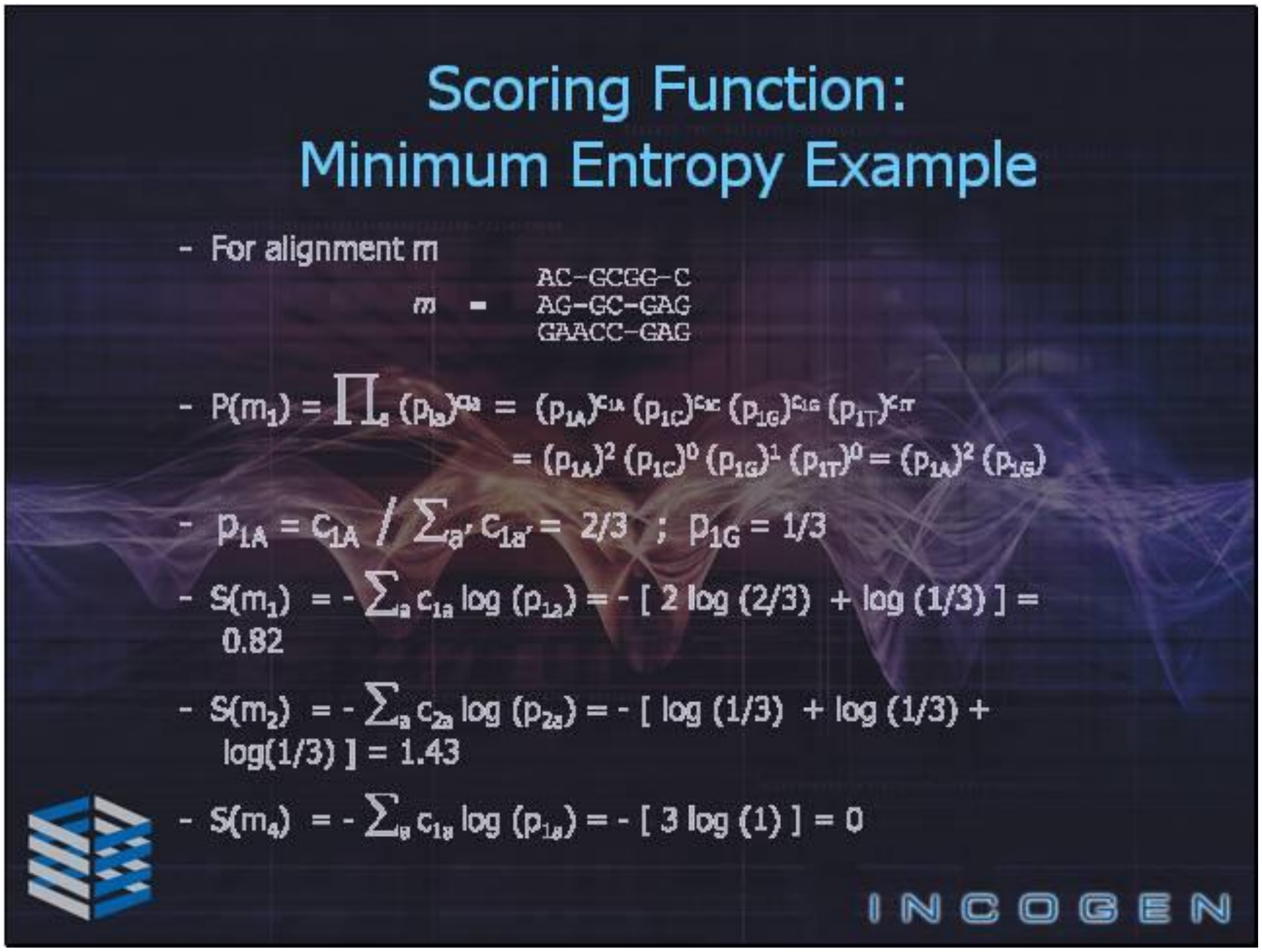
- Measures variability observed in an aligned columns of residue
- Estimate for $p_{ia} = \dfrac{c_{ia}}{\sum_{a'} c_{ia'}}$
- "Good alignment" minimizes total entropy $\sum_i S(m_i)$

INCOGEN

Slide notes

The first scoring function we are going to discuss is the minimum entropy scoring function. The goal of this scoring algorithm is to minimize the entropy, or randomness, in the in the alignment. To calculate the entropy of the alignment, first, we must calculate the probability of column i and then use that probability to calculate a score for that column. This score measures the variability observed in the aligned column i. By minimizing the sum of this column score over all of the columns, we minimize the entropy and create a “good” alignment.

Slide 11 - Scoring Function: Minimum Entropy Example



Scoring Function: Minimum Entropy Example

- For alignment m

```
           AC-GCGG-C
  m  =     AG-GC-GAG
           GAACC-GAG
```

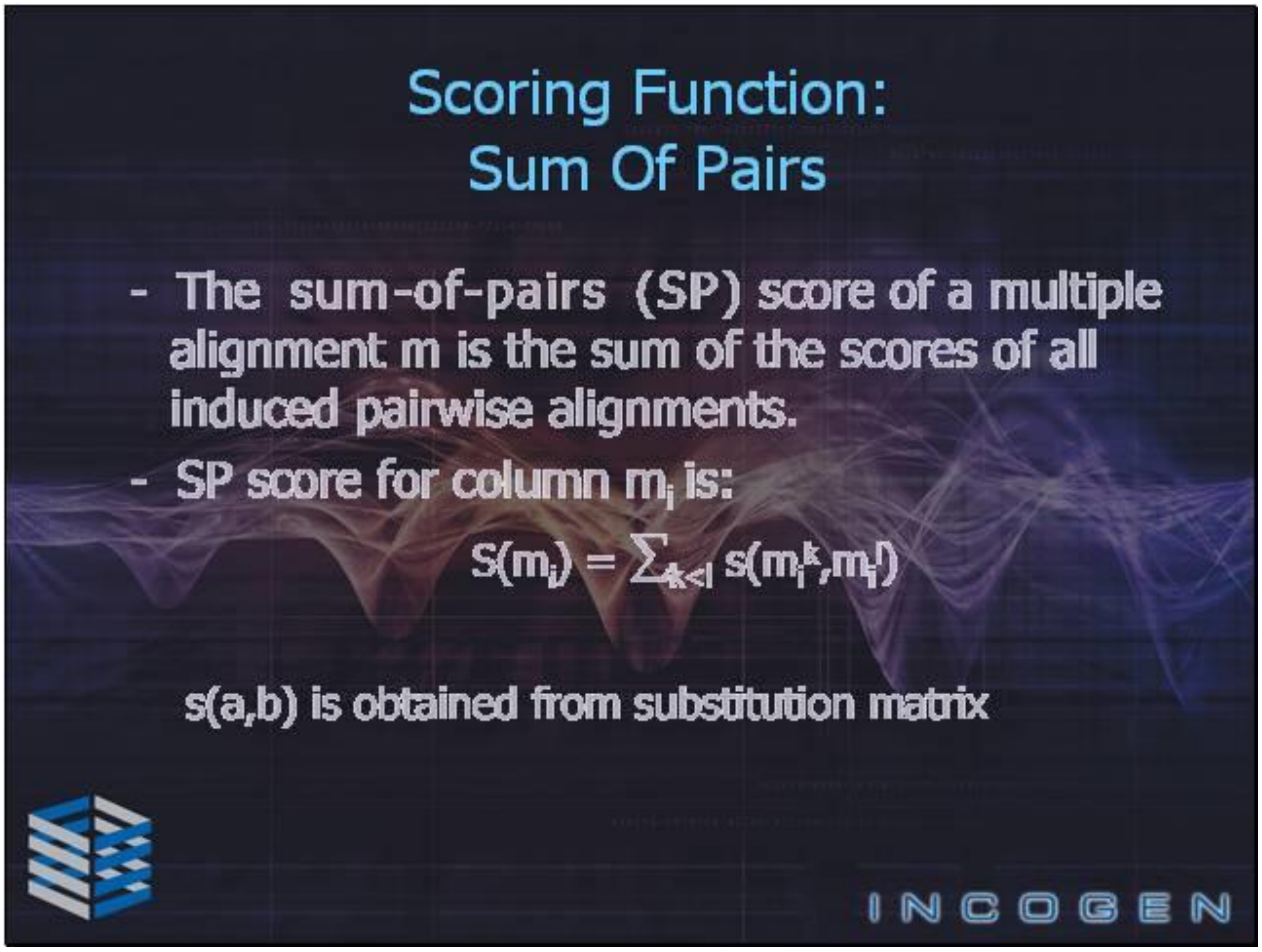
- $P(m_1) = \prod_a (p_{1a})^{c_{1a}} = (p_{1A})^{c_{1A}} (p_{1C})^{c_{1C}} (p_{1G})^{c_{1G}} (p_{1T})^{c_{1T}}$
  $= (p_{1A})^2 (p_{1C})^0 (p_{1G})^1 (p_{1T})^0 = (p_{1A})^2 (p_{1G})$
- $p_{1A} = c_{1A} / \sum_{a'} c_{1a'} = 2/3 \;;\; p_{1G} = 1/3$
- $S(m_1) = -\sum_a c_{1a} \log (p_{1a}) = -[\, 2 \log (2/3) + \log (1/3)\,] = 0.82$
- $S(m_2) = -\sum_a c_{2a} \log (p_{2a}) = -[\, \log (1/3) + \log (1/3) + \log(1/3)\,] = 1.43$
- $S(m_4) = -\sum_a c_{1a} \log (p_{1a}) = -[\, 3 \log (1)\,] = 0$

INCOGEN

Slide notes

Let’s look at a simple example. Here is a short multiple alignment consisting of three nucleotide sequences and nine columns.

Slide 12 - Scoring Function: Sum Of Pairs



Scoring Function: Sum Of Pairs

- The sum-of-pairs (SP) score of a multiple alignment m is the sum of the scores of all induced pairwise alignments.
- SP score for column m_i is:

$$S(m_i) = \sum_{k<l} s(m_i^k, m_i^l)$$

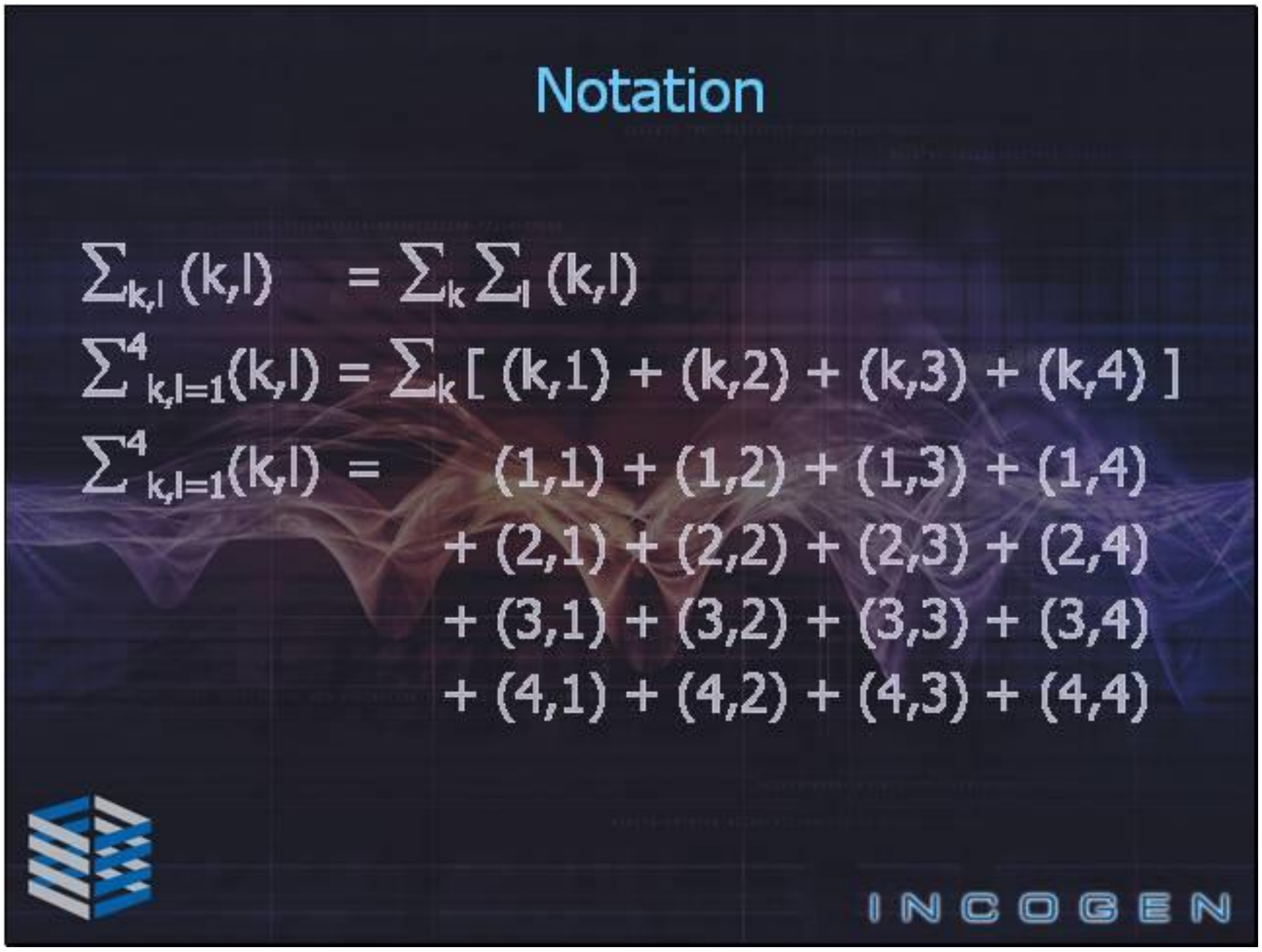
s(a,b) is obtained from substitution matrix

INCOGEN

Slide notes

Another scoring algorithm is the sum of pairs algorithm. In this scoring algorithm, the score of an MSA is the sum of the scores of all of the pairwise alignments.

Slide 13 - Notation



Notation

$$\sum_{k,l} (k,l) = \sum_k \sum_l (k,l)$$

$$\sum^4_{k,l=1}(k,l) = \sum_k [(k,1) + (k,2) + (k,3) + (k,4)]$$

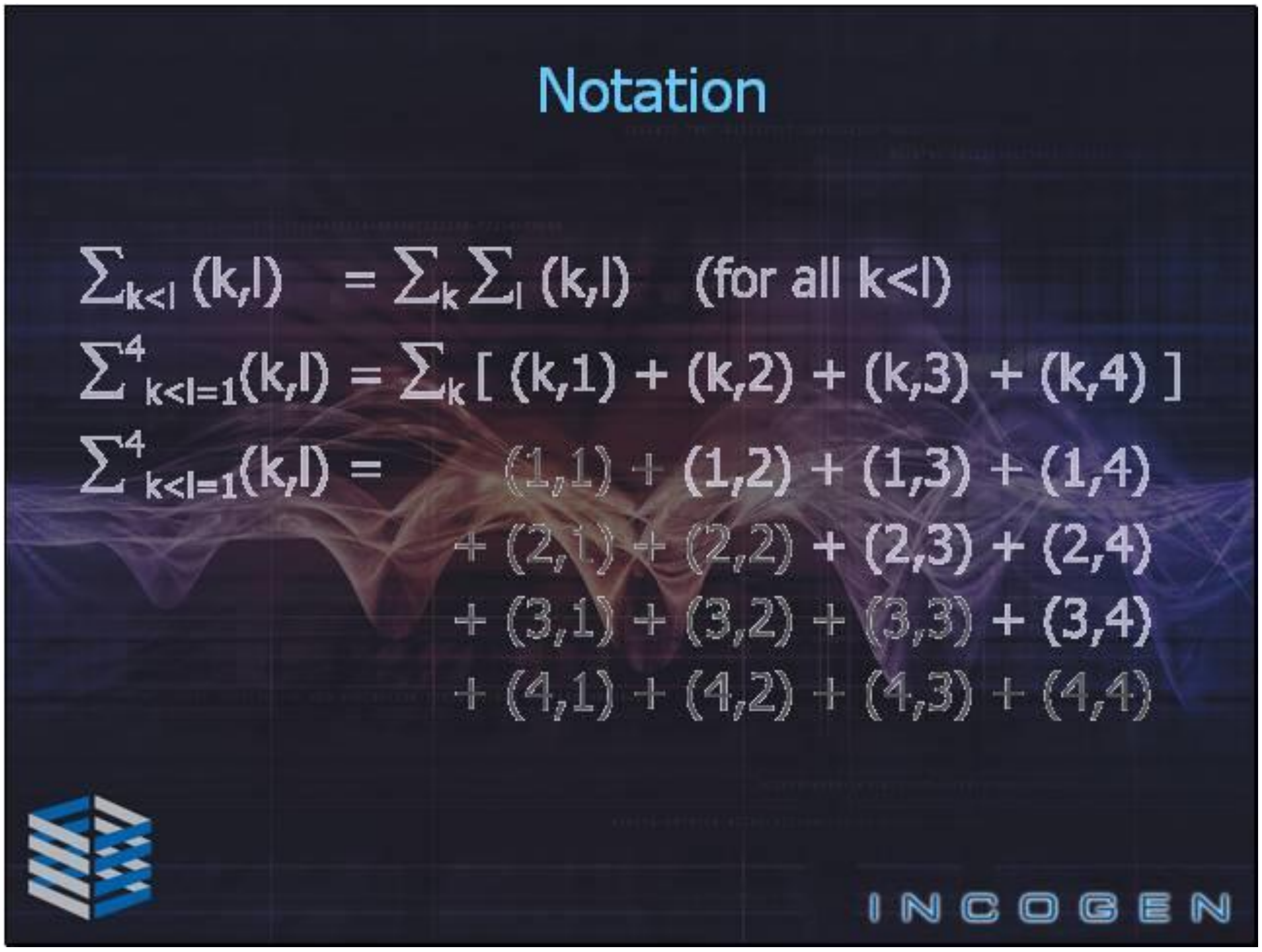
$$\begin{aligned}\sum^4_{k,l=1}(k,l) = \; & (1,1) + (1,2) + (1,3) + (1,4) \\ & + (2,1) + (2,2) + (2,3) + (2,4) \\ & + (3,1) + (3,2) + (3,3) + (3,4) \\ & + (4,1) + (4,2) + (4,3) + (4,4)\end{aligned}$$

INCOGEN

Slide notes

The notation for this scoring algorithm is fairly straight forward. At first glance, we want to do a sum of the alignment scores between every sequence k and every sequence l. First, we take the sum over every sequence l and then we take the sum over every sequence k. When we do that, however, we find that we are including the alignments of all of the sequences with themselves, such as (1,1), and that we are double-counting all of the other alignments, since the alignment of sequence 1 with sequence 2 is the same as the alignment of sequence 2 with sequence 1.

Slide 14 - Notation



Notation

$$\sum_{k<l} (k,l) = \sum_k \sum_l (k,l) \quad \text{(for all } k<l\text{)}$$

$$\sum^4_{k<l=1}(k,l) = \sum_k [\ (k,1) + (k,2) + (k,3) + (k,4)\ ]$$

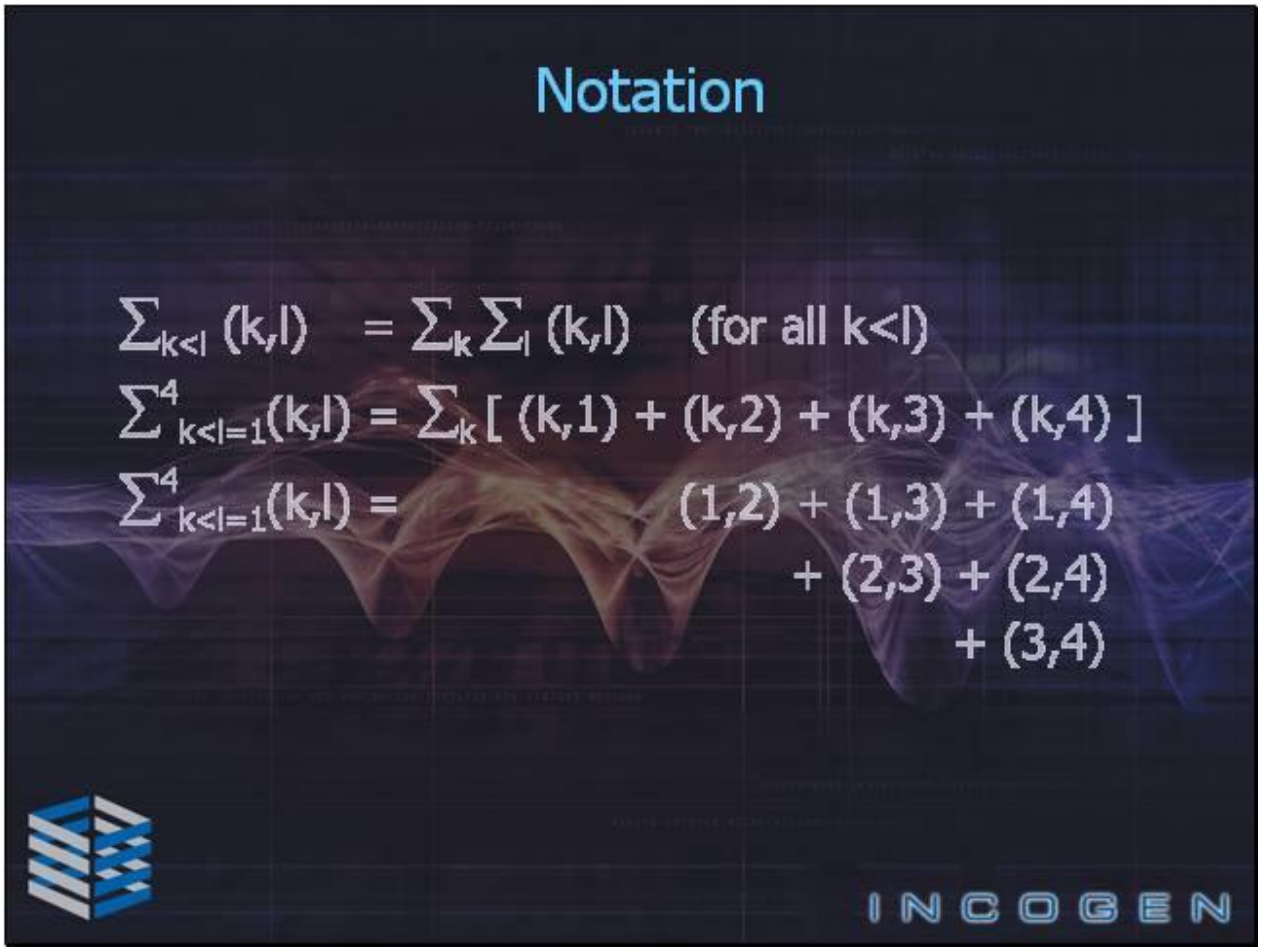
$$\begin{aligned}\sum^4_{k<l=1}(k,l) = \quad & (1,1) + (1,2) + (1,3) + (1,4) \\ & + (2,1) + (2,2) + (2,3) + (2,4) \\ & + (3,1) + (3,2) + (3,3) + (3,4) \\ & + (4,1) + (4,2) + (4,3) + (4,4)\end{aligned}$$

INCOGEN

Slide notes

What we really want is the sum of the alignment scores between every sequence k and every other sequence l, where k < l. This will prevent counting alignments between a sequence and itself, and it will prevent double-counting the other alignments,

Slide 15 - Notation



Notation

$\sum_{k<l} (k,l) \quad = \sum_k \sum_l (k,l) \quad$ (for all k<l)

$\sum^4_{k<l=1}(k,l) = \sum_k [\ (k,1) + (k,2) + (k,3) + (k,4)\ ]$

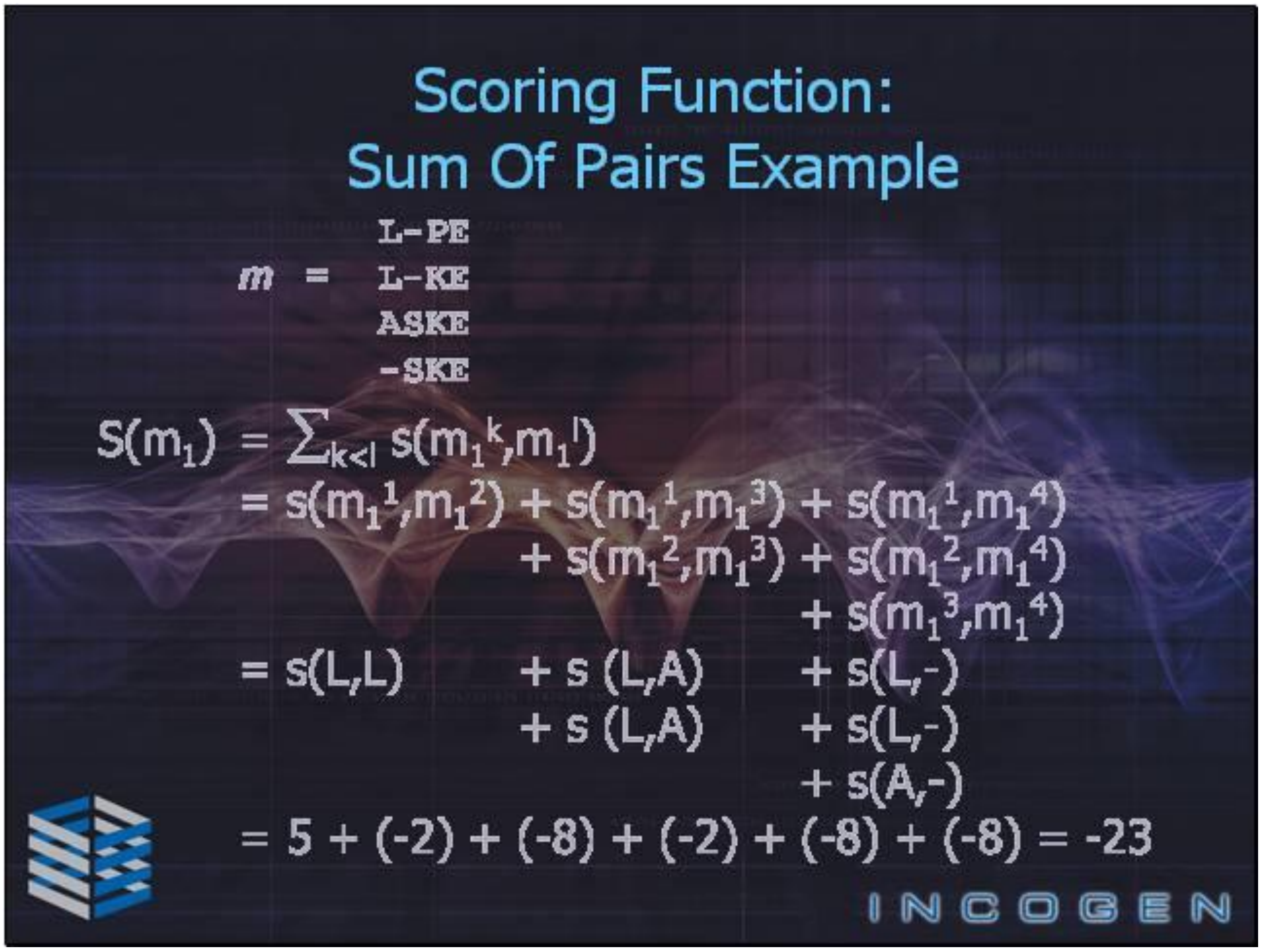
$$\begin{aligned}\sum^4_{k<l=1}(k,l) = \quad & (1,2) + (1,3) + (1,4) \\ & + (2,3) + (2,4) \\ & + (3,4)\end{aligned}$$

INCOGEN

Slide notes

…leaving us with this formula.

Slide 16 - Scoring Function: Sum Of Pairs Example



Scoring Function: Sum Of Pairs Example

```
         L-PE
m  =     L-KE
         ASKE
         -SKE
```

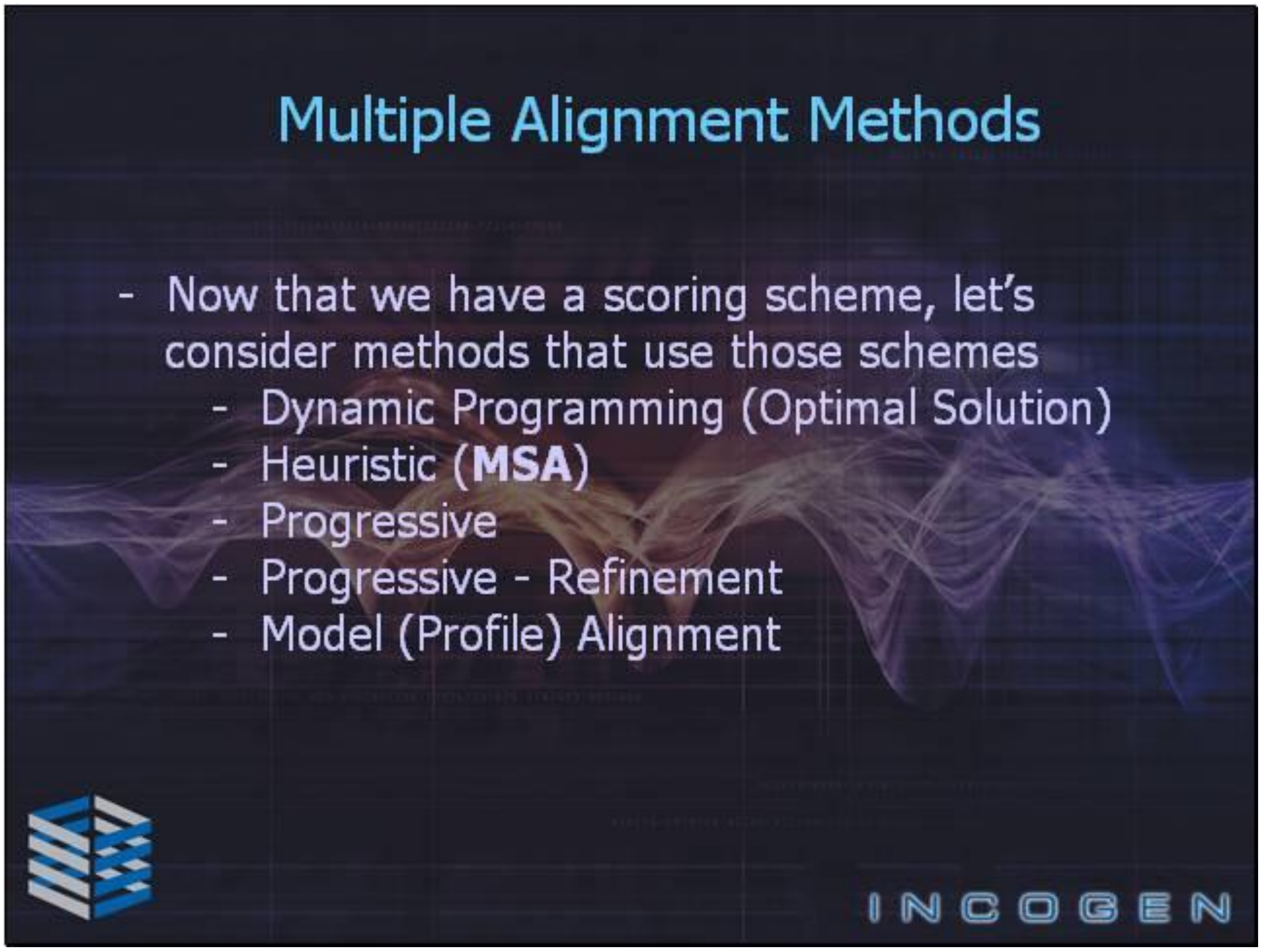
$$S(m_1) = \sum_{k<l} s(m_1^k, m_1^l)$$
$$= s(m_1^1,m_1^2) + s(m_1^1,m_1^3) + s(m_1^1,m_1^4) + s(m_1^2,m_1^3) + s(m_1^2,m_1^4) + s(m_1^3,m_1^4)$$
$$= s(L,L) + s(L,A) + s(L,-) + s(L,A) + s(L,-) + s(A,-)$$
$$= 5 + (-2) + (-8) + (-2) + (-8) + (-8) = -23$$

INCOGEN

Slide notes

Let’s look at an example. For this particular example, we have used BLOSUM50 as the pairwise sequence alignment scoring matrix.

Slide 17 - Multiple Alignment Methods



Multiple Alignment Methods

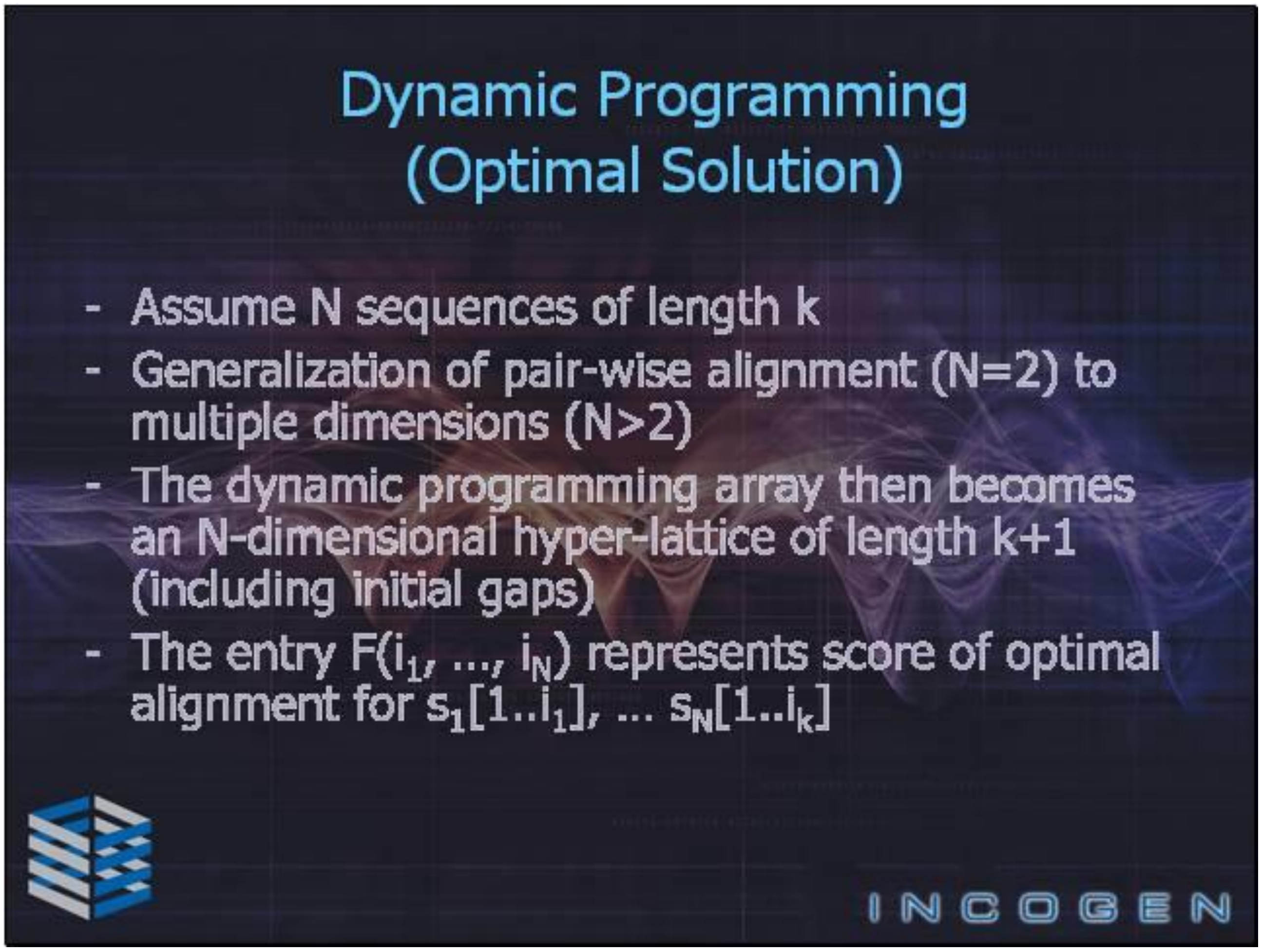
- Now that we have a scoring scheme, let’s consider methods that use those schemes
  - Dynamic Programming (Optimal Solution)
  - Heuristic (**MSA**)
  - Progressive
  - Progressive - Refinement
  - Model (Profile) Alignment

INCOGEN

Slide notes

Now that we have a scoring function, let’s take a brief look at the methods that use these functions: dynamic programming, heuristic, progressive, progressive with refinement, and model or profile alignment.

Slide 18 - Dynamic Programming (Optimal Solution)



Dynamic Programming (Optimal Solution)

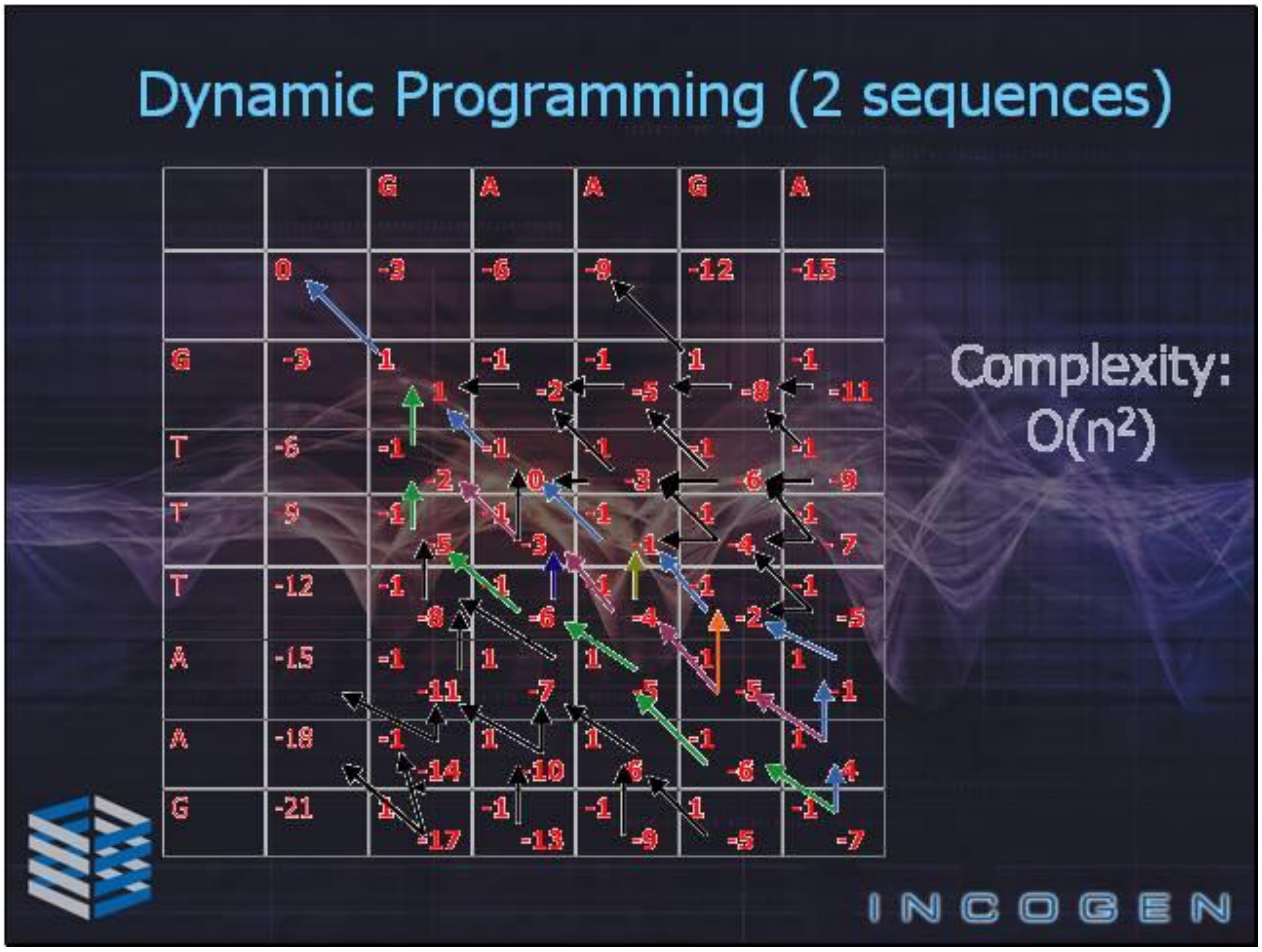
- Assume N sequences of length k
- Generalization of pair-wise alignment (N=2) to multiple dimensions (N>2)
- The dynamic programming array then becomes an N-dimensional hyper-lattice of length k+1 (including initial gaps)
- The entry $F(i_1, ..., i_N)$ represents score of optimal alignment for $s_1[1..i_1], ... s_N[1..i_k]$

INCOGEN

Slide notes

Let’s look at dynamic programming first. Since this is the method used to find pairwise alignments, it seems like an obvious first choice to calculate multiple alignments. The modifications to the pairwise alignment algorithm are fairly straightforward. We now have k sequences, where k > 2, and the dynamic programming array now has N dimensions instead of two. The calculations themselves remain the same.

Slide 19 - Slide 19



Dynamic Programming (2 sequences)

		G	A	A	G	A
	0	-3	-6	-9	-12	-15
G	-3	1 1	-1 -2	-1 -5	1 -8	-1 -11
T	-6	-1 -2	-1 0	-1 -3	-1 -6	-1 -9
T	-9	-1 -5	-1 -3	-1 -1	-1 -4	-1 -7
T	-12	-1 -8	-1 -6	-1 -4	-1 -2	-1 -5
A	-15	-1 -11	1 -7	1 -5	-1 -5	1 -1
A	-18	-1 -14	1 -10	1 -6	-1 -6	1 -4
G	-21	1 -17	-1 -13	-1 -9	1 -5	-1 -7

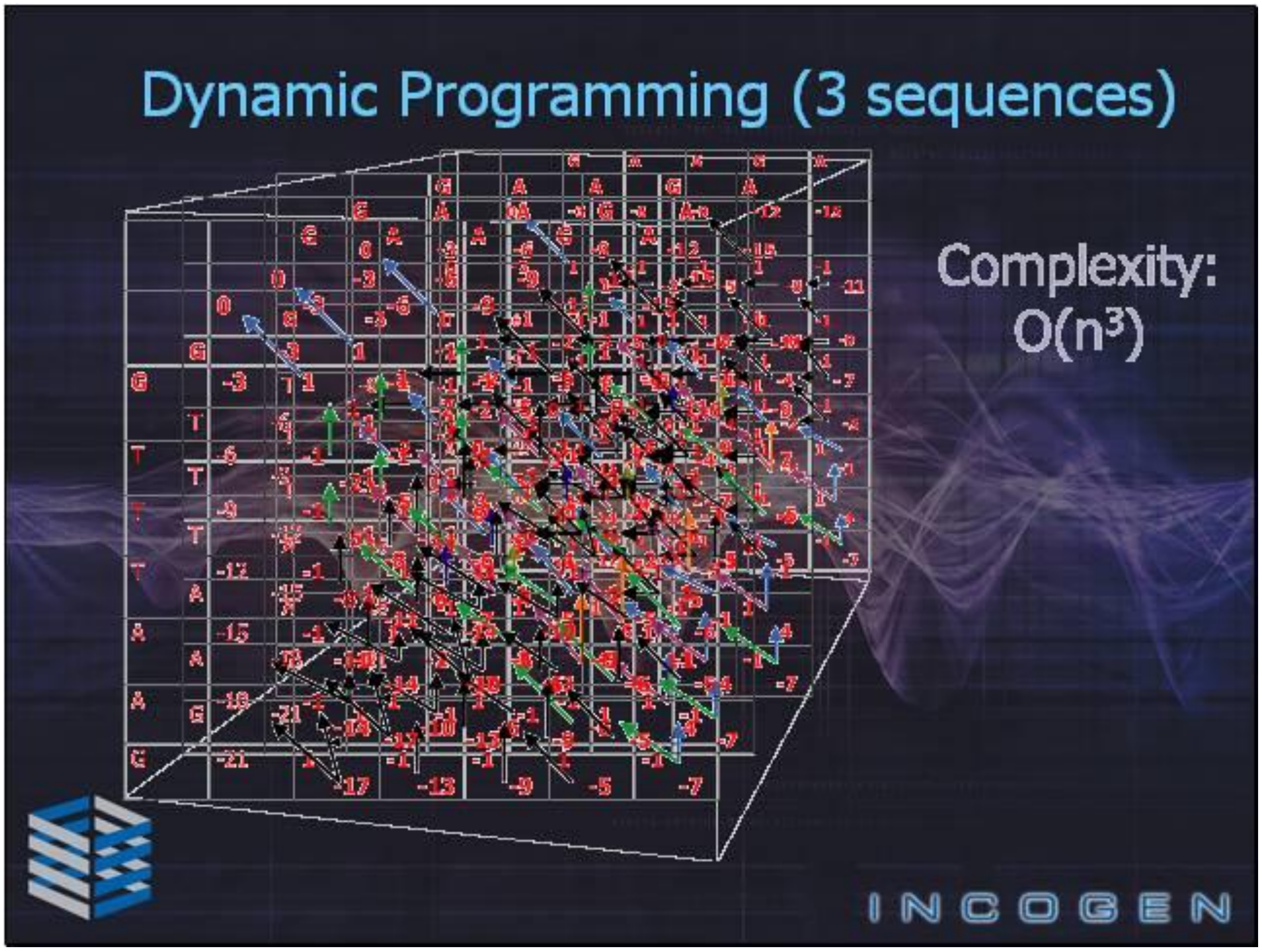
Complexity: $O(n^2)$

INCOGEN

Slide notes

This is an example of what the dynamic programming array looks like when aligning two sequences. The complexity of this algorithm is O(n2), where n is the length of the sequence.

Slide 20 - Slide 20



Dynamic Programming (3 sequences)

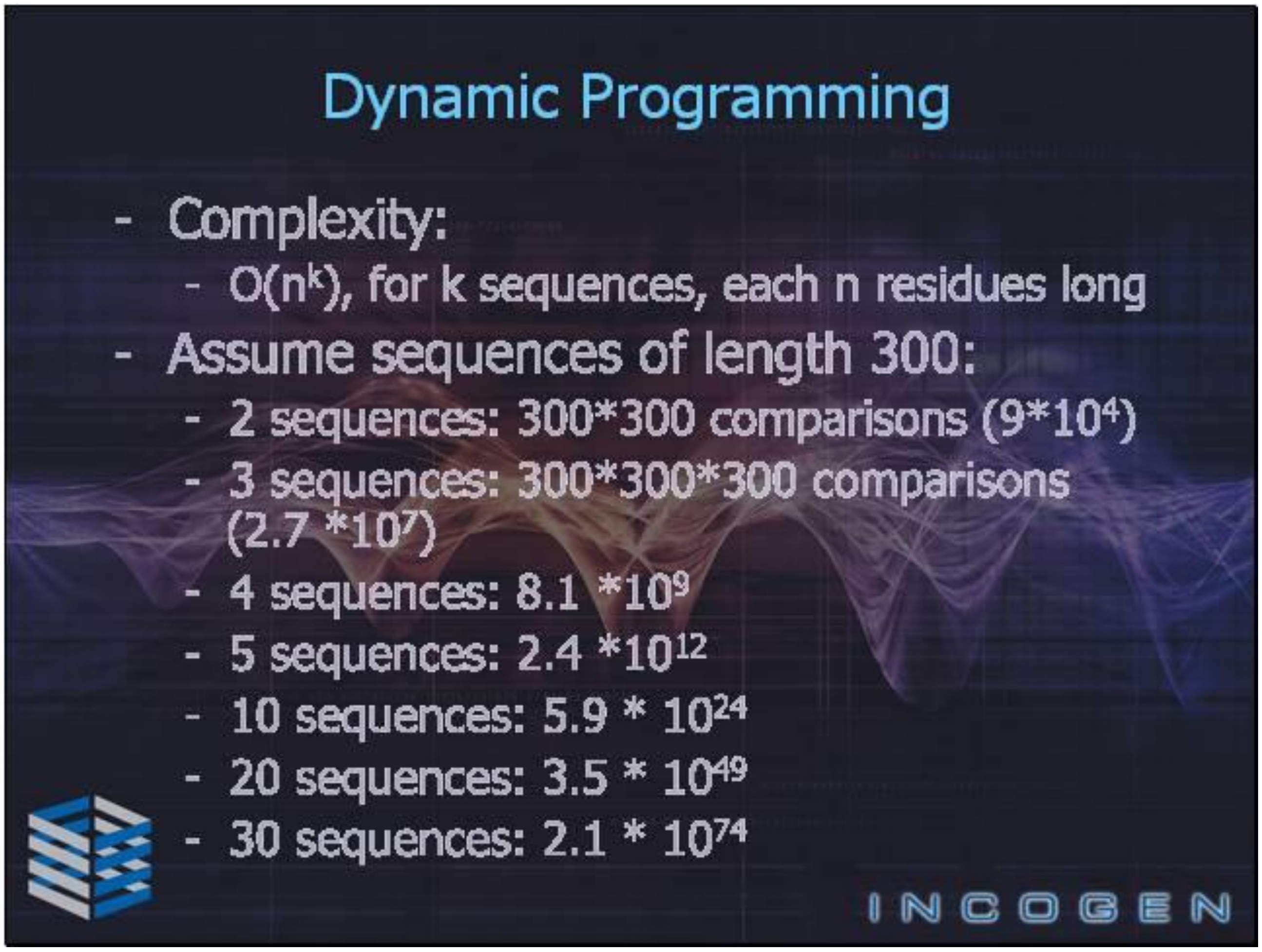
Complexity: $O(n^3)$

INCOGEN

Slide notes

If we move to a multiple sequence alignment with three sequences, our dynamic programming array looks like a cube. Visually, this is a bit more confusing to look at, and the complexity increased to O(n3).

Slide 21 - Dynamic Programming



Dynamic Programming

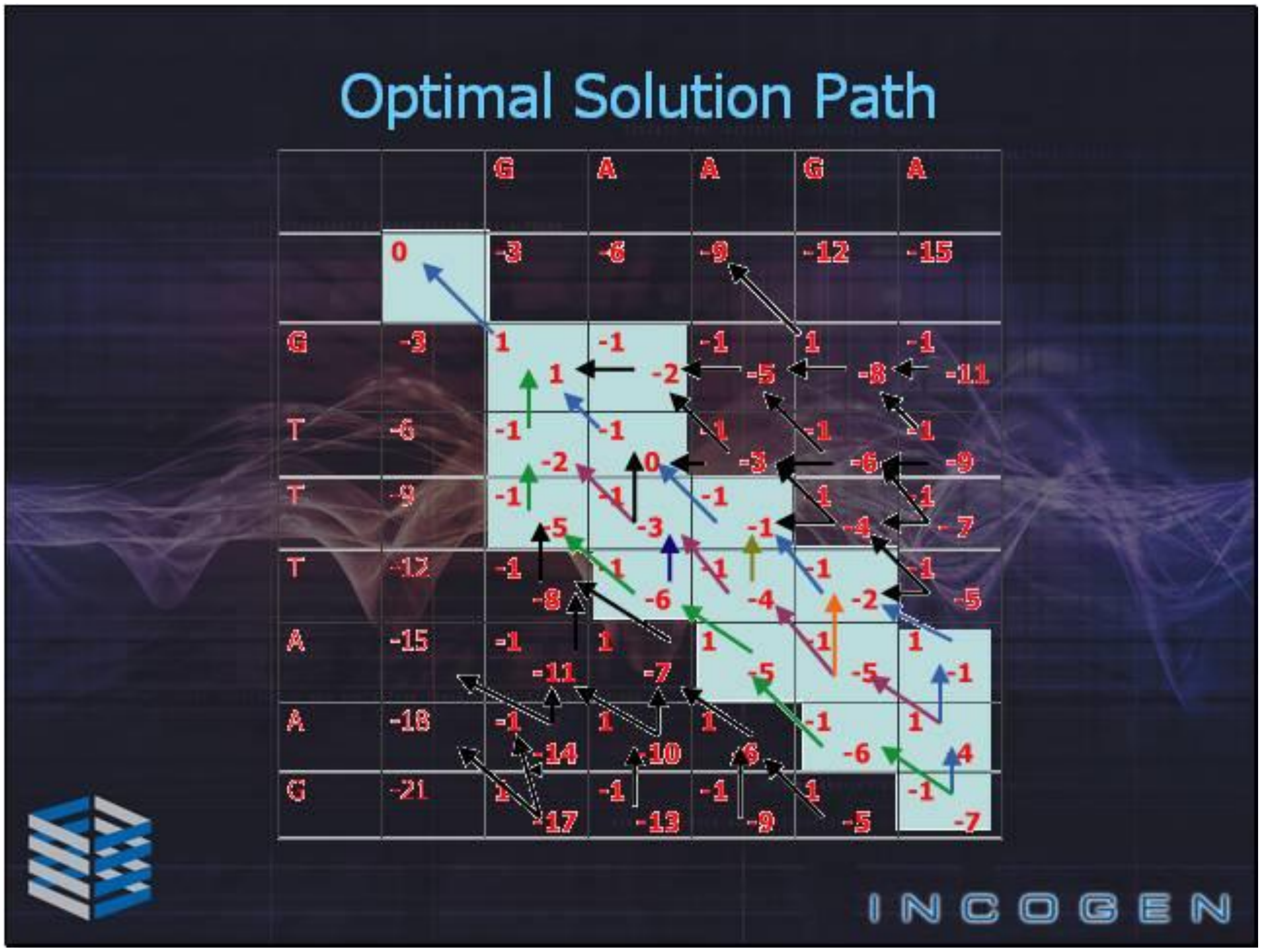
- Complexity:
  - $O(n^k)$, for k sequences, each n residues long
- Assume sequences of length 300:
  - 2 sequences: 300*300 comparisons ($9*10^4$)
  - 3 sequences: 300*300*300 comparisons ($2.7 *10^7$)
  - 4 sequences: $8.1 *10^9$
  - 5 sequences: $2.4 *10^{12}$
  - 10 sequences: $5.9 * 10^{24}$
  - 20 sequences: $3.5 * 10^{49}$
  - 30 sequences: $2.1 * 10^{74}$

INCOGEN

Slide notes

So, if we have k sequences, each n residues long, the complexity of the dynamic programming algorithm is O(nk). This grows very quickly ask increases. Let me show you a few examples using sequences of length 300.

Slide 22 - Slide 22



Optimal Solution Path

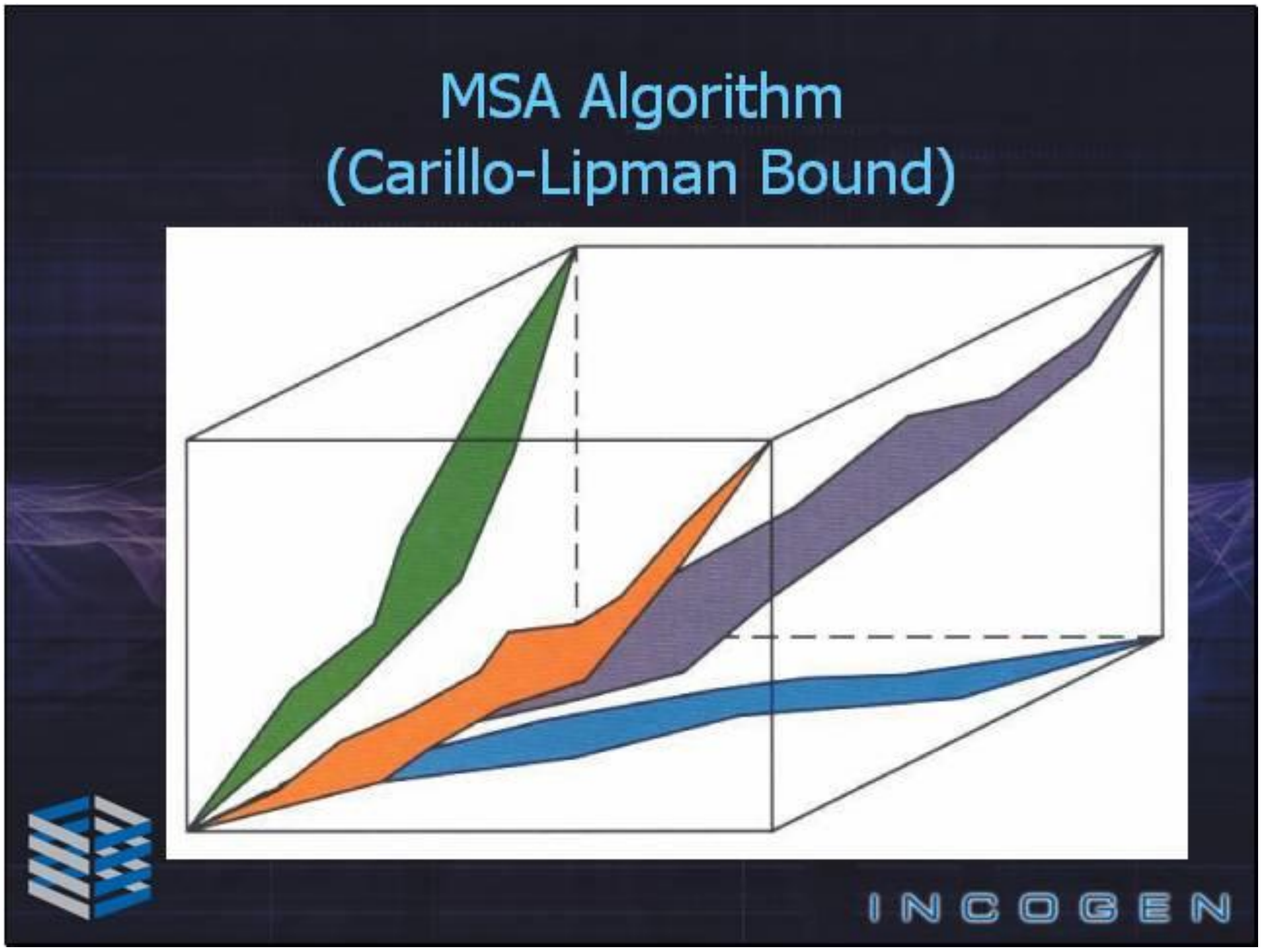
		G	A	A	G	A
	0	-3	-6	-9	-12	-15
G	-3	1 1	-1 -2	-1 -5	1 -8	-1 -11
T	-6	-1 -2	-1 0	-1 -3	-1 -6	-1 -9
T	-9	-1 -5	-1 -3	-1 -1	1 -4	1 -7
T	-12	-1 -8	1 -6	-1 -4	-1 -2	-1 -5
A	-15	-1 -11	1 -7	1 -5	-1 -5	1 -1
A	-18	-1 -14	1 -10	1 -6	-1 -6	1 -4
G	-21	1 -17	-1 -13	-1 -9	1 -5	-1 -7

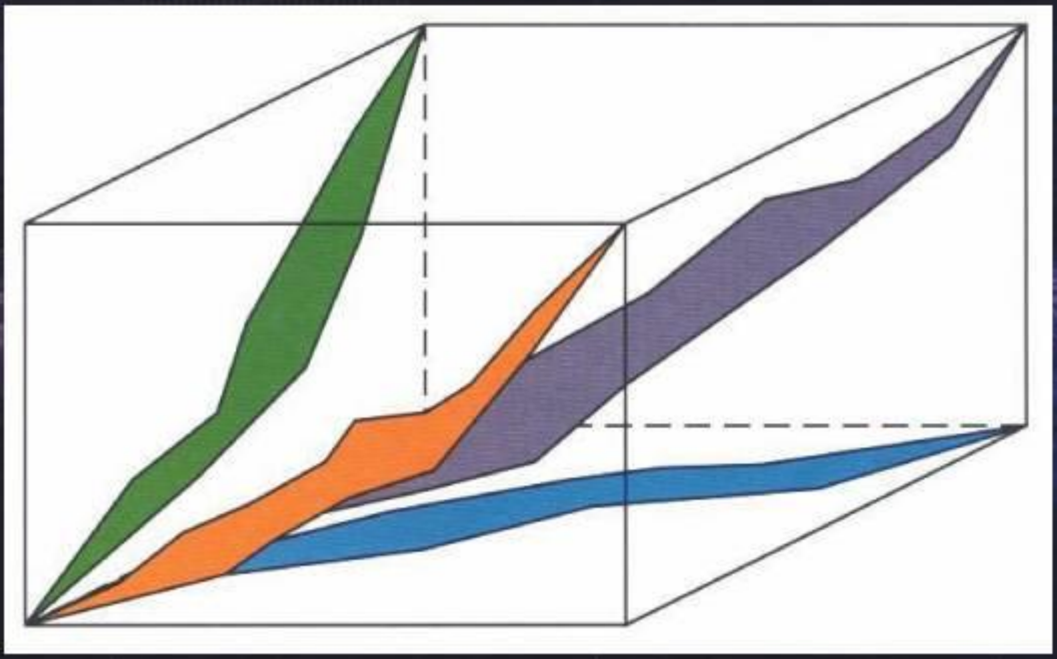
INCOGEN

Slide notes

We can decrease the number of comparisons that we need to do if we remember that for global alignments, the solution is generally found within a small area around the diagonal of the dynamic programming array. We can use a heuristic method to eliminate the areas where solutions are rarely found. This eliminates a lot of generally unneeded calculations at the expense of the rarely found solution that lies outside this region.

Slide 23 - Slide 23

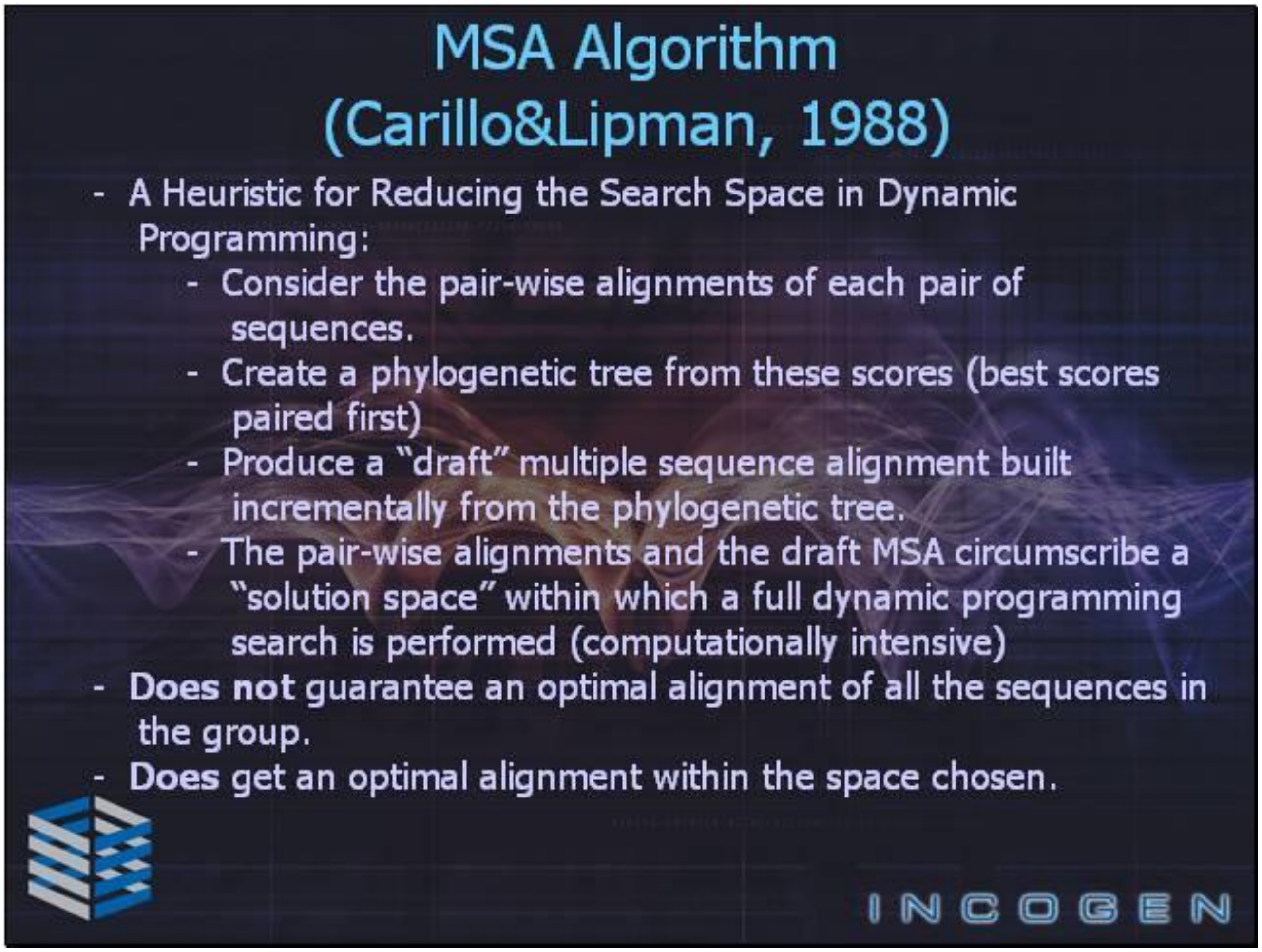

MSA Algorithm (Carillo-Lipman Bound)



Slide notes

An example of a heuristic algorithm is the MSA algorithm, which uses the Carillo-Lipman Bound procedure. This procedure provides a bound in the form of a polyhedron around the diagonal in a hypercube. This bounds the search space for finding the MSA of a set of sequences.

Slide 24 - MSAAlgorithm (Carillo&Lipman, 1988)



MSA Algorithm (Carillo&Lipman, 1988)

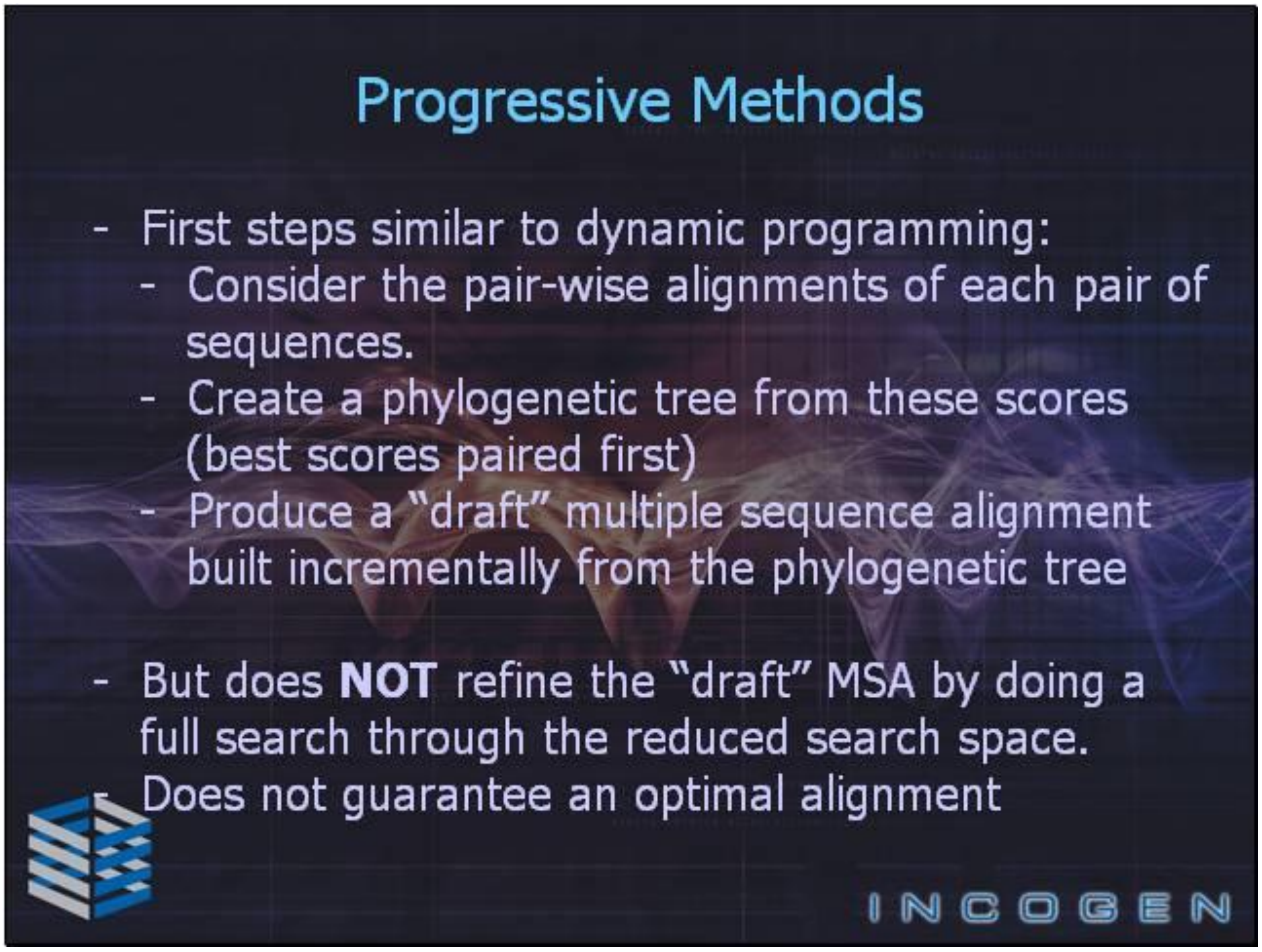
- A Heuristic for Reducing the Search Space in Dynamic Programming:
  - Consider the pair-wise alignments of each pair of sequences.
  - Create a phylogenetic tree from these scores (best scores paired first)
  - Produce a “draft” multiple sequence alignment built incrementally from the phylogenetic tree.
  - The pair-wise alignments and the draft MSA circumscribe a “solution space” within which a full dynamic programming search is performed (computationally intensive)
- **Does not** guarantee an optimal alignment of all the sequences in the group.
- **Does** get an optimal alignment within the space chosen.

INCOGEN

Slide notes

First, we consider the pairwise alignments of each pair of sequences and create a phylogenetic tree from these scores. We then produce a “draft” MSA built from the phylogenetic tree. The pairwise alignments and draft MSA provide us with a reduced solution space on which we can use dynamic programming to find the solution. This method does not guarantee an optimal alignment for all the sequences in the group because so much of the solution space is excluded from the search. It does, however, give us an optimal alignment from within the search space.

Slide 25 - Progressive Methods



Progressive Methods

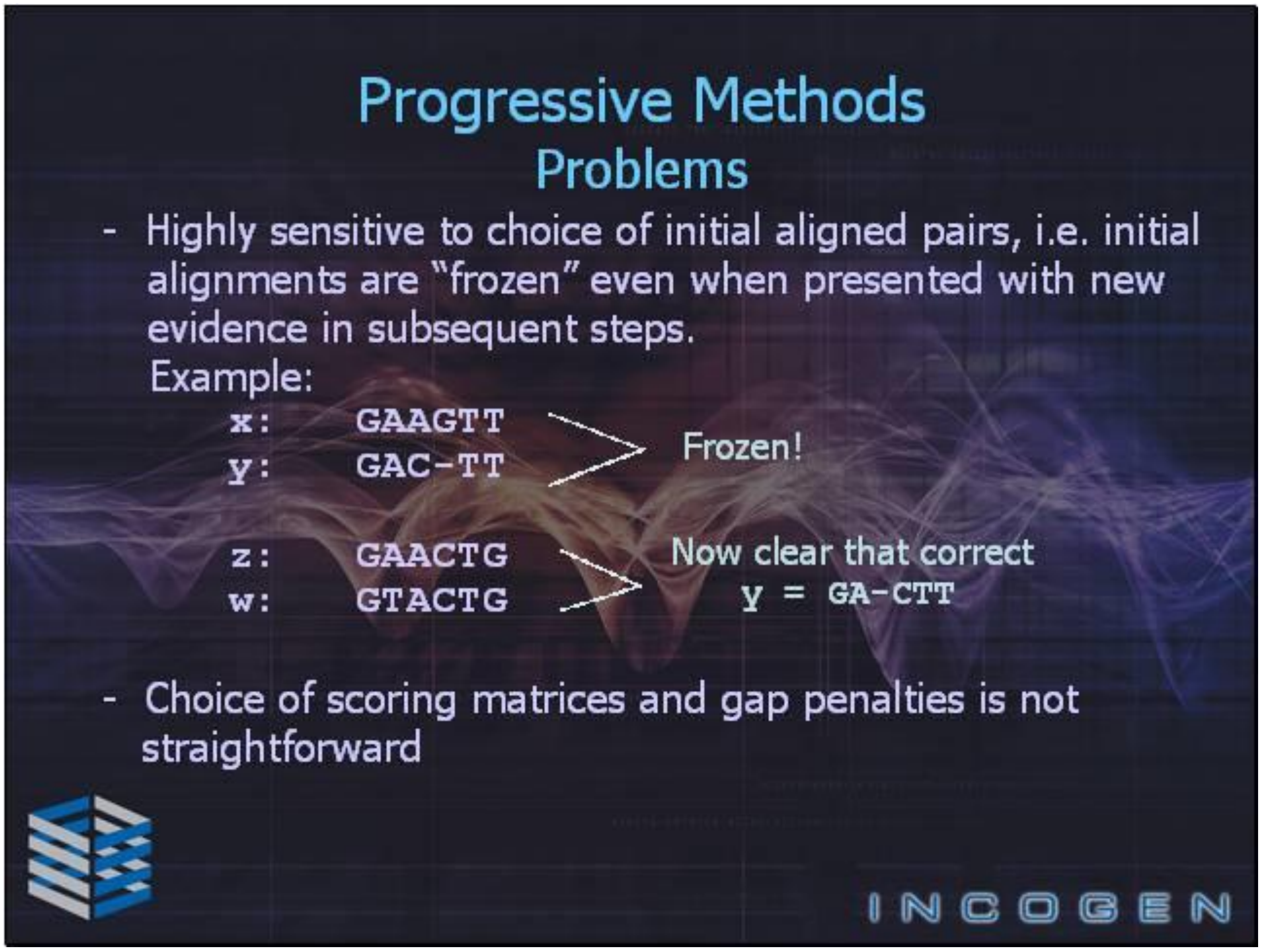
- First steps similar to dynamic programming:
  - Consider the pair-wise alignments of each pair of sequences.
  - Create a phylogenetic tree from these scores (best scores paired first)
  - Produce a "draft" multiple sequence alignment built incrementally from the phylogenetic tree
- But does **NOT** refine the "draft" MSA by doing a full search through the reduced search space.
- Does not guarantee an optimal alignment

INCOGEN

Slide notes

Progressive methods are another way of calculating MSAs. The first steps used in progressive methods are very similar to those used in the MSA algorithm. However, the progressive methods do not, by default, refine the draft MSA by doing a full search in the smaller search area. It also does not guarantee an optimal alignment.

Slide 26 - Progressive Methods Problems



Progressive Methods
Problems

- Highly sensitive to choice of initial aligned pairs, i.e. initial alignments are "frozen" even when presented with new evidence in subsequent steps.

  Example:

```
x:   GAAGTT  \
              > Frozen!
y:   GAC-TT  /

z:   GAACTG  \
              > Now clear that correct
w:   GTACTG  /     y = GA-CTT
```

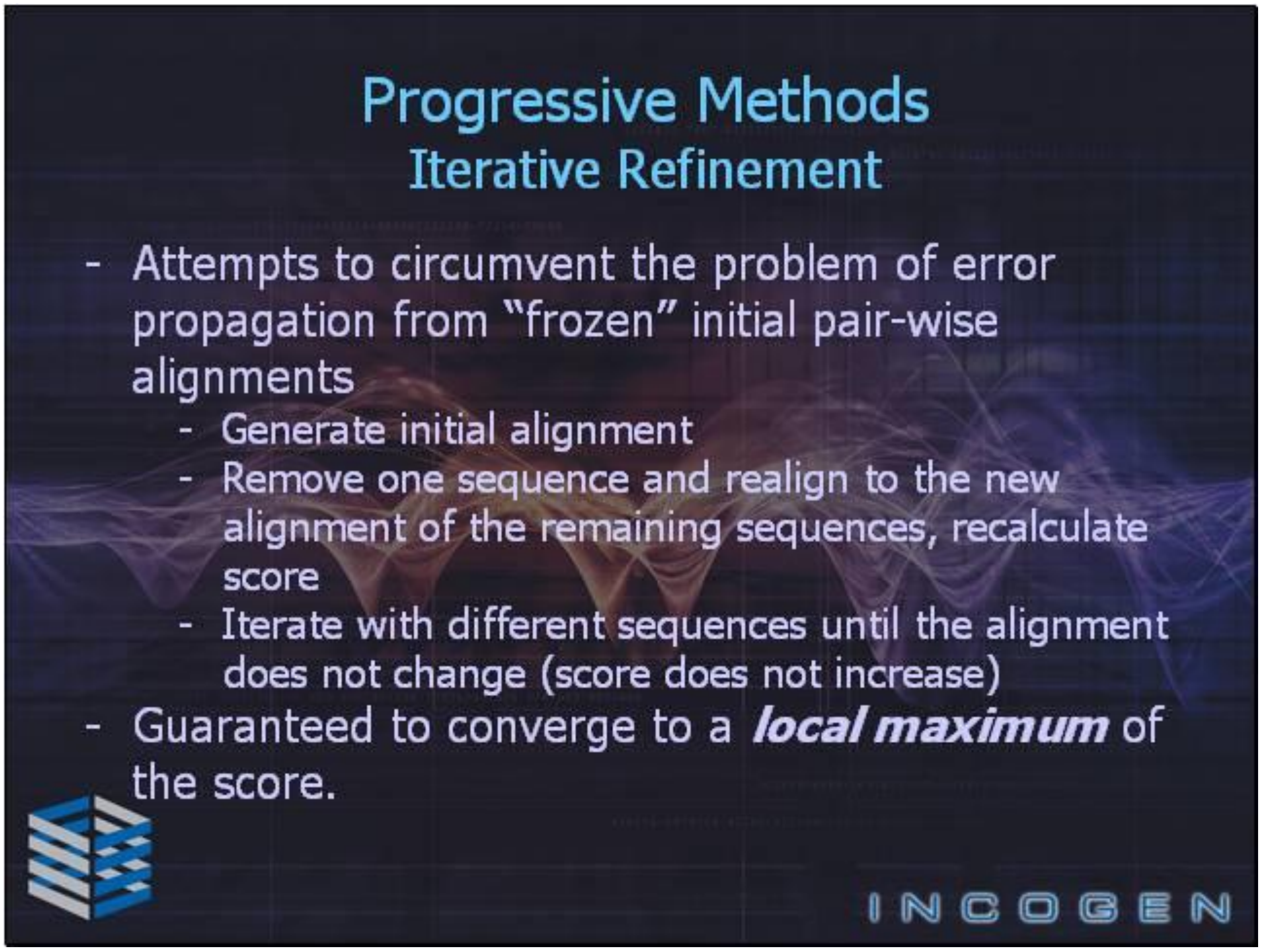
- Choice of scoring matrices and gap penalties is not straightforward

INCOGEN

Slide notes

The progressive methods have a few drawbacks. First, they are highly sensitive to the choice of initial aligned pairs. These initial pairs are "frozen" even if subsequent steps show that they are not correct. For example, there are a couple of equivalent looking pairwise alignments for these two sequences, and the algorithm happens to have chosen this one. This alignment is now frozen and cannot be changed. However, as we look at other sequences that we need to align, it becomes obvious that the gap in sequence y should have been placed elsewhere. The choice of scoring matrices and gap penalties can make it more likely that you will get the correct initial alignments, but choosing the best one is not straightforward. The likelihood of large errors in the initial alignments increases as the sequences become more distantly related.

Slide 27 - Progressive Methods Iterative Refinement



Progressive Methods
Iterative Refinement

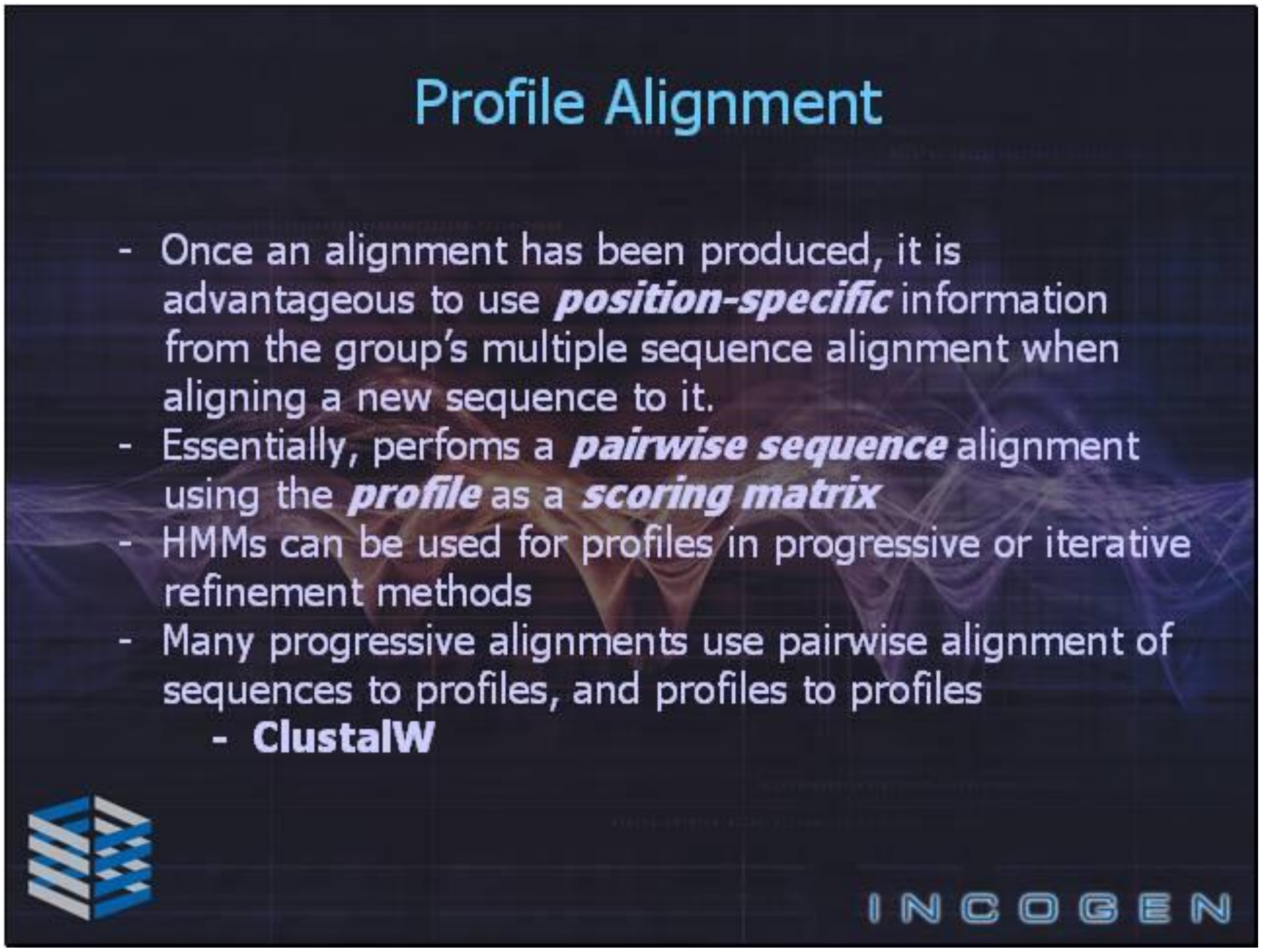
- Attempts to circumvent the problem of error propagation from "frozen" initial pair-wise alignments
    - Generate initial alignment
    - Remove one sequence and realign to the new alignment of the remaining sequences, recalculate score
    - Iterate with different sequences until the alignment does not change (score does not increase)
- Guaranteed to converge to a ***local maximum*** of the score.

INCOGEN

Slide notes

Some researchers have attempted to improve upon the progressive methods by adding refinement steps to the procedure. In this case, we generate the initial alignment and then remove a sequence from the alignment and align it with the remainder of the MSA. We continue to do this until the alignment score no longer increases. This procedure is guaranteed to converge to a local maximum. If the initial alignment is poor, it may not converge to the global maximum.

Slide 28 - Profile Alignment



Profile Alignment

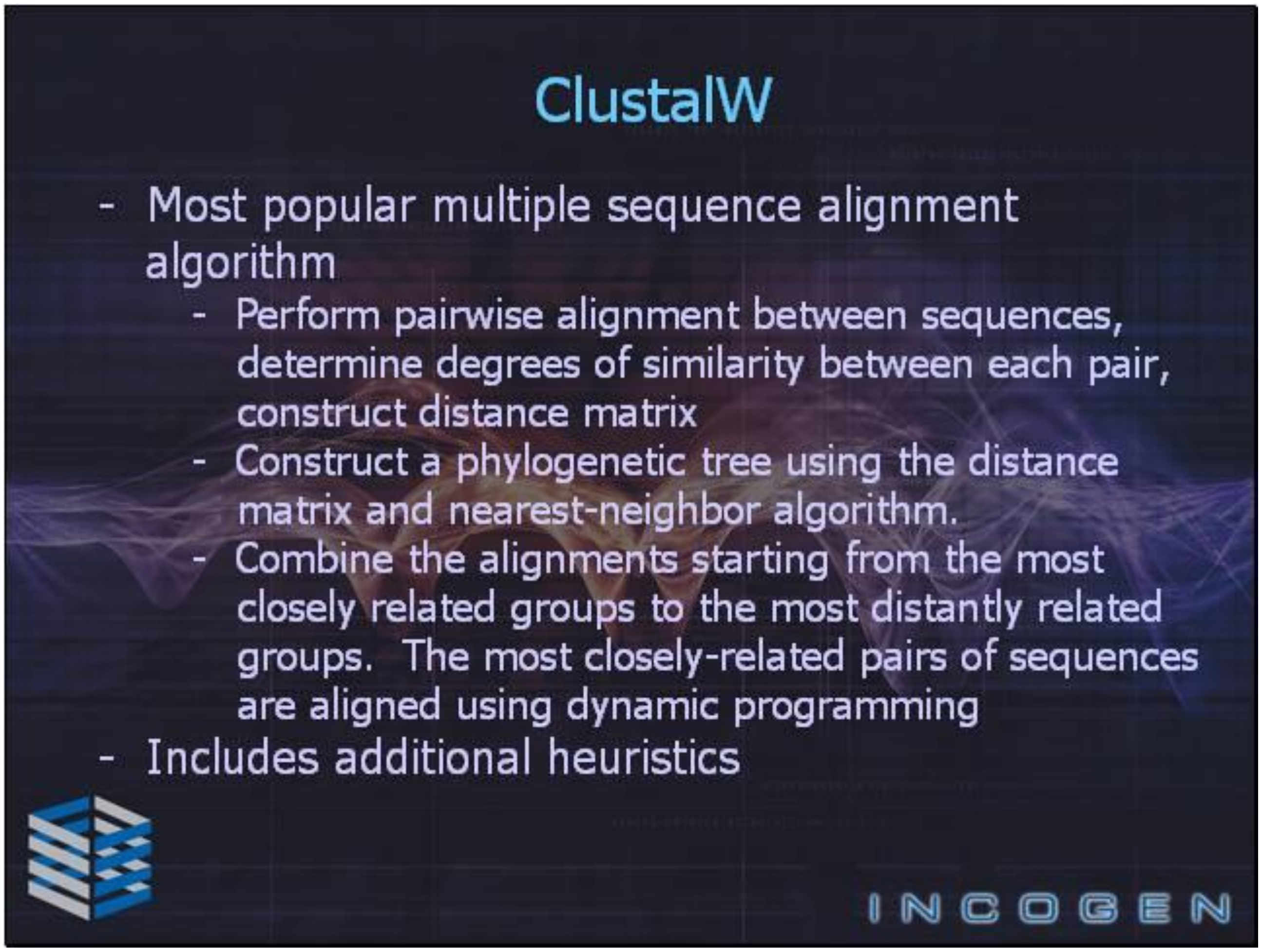
- Once an alignment has been produced, it is advantageous to use ***position-specific*** information from the group's multiple sequence alignment when aligning a new sequence to it.
- Essentially, perfoms a ***pairwise sequence*** alignment using the ***profile*** as a ***scoring matrix***
- HMMs can be used for profiles in progressive or iterative refinement methods
- Many progressive alignments use pairwise alignment of sequences to profiles, and profiles to profiles
  - **ClustalW**

INCOGEN

Slide notes

Once an alignment has been generated, it is helpful to use the position specific information from the MSA when adding new sequences to the alignment. Essentially, we perform a pairwise sequence alignment of the new sequence to the MSA's profile. Many progressive alignment methods, such as ClustalW, use pairwise alignment of sequences to profiles and profiles to profiles.

Slide 29 - ClustalW



ClustalW

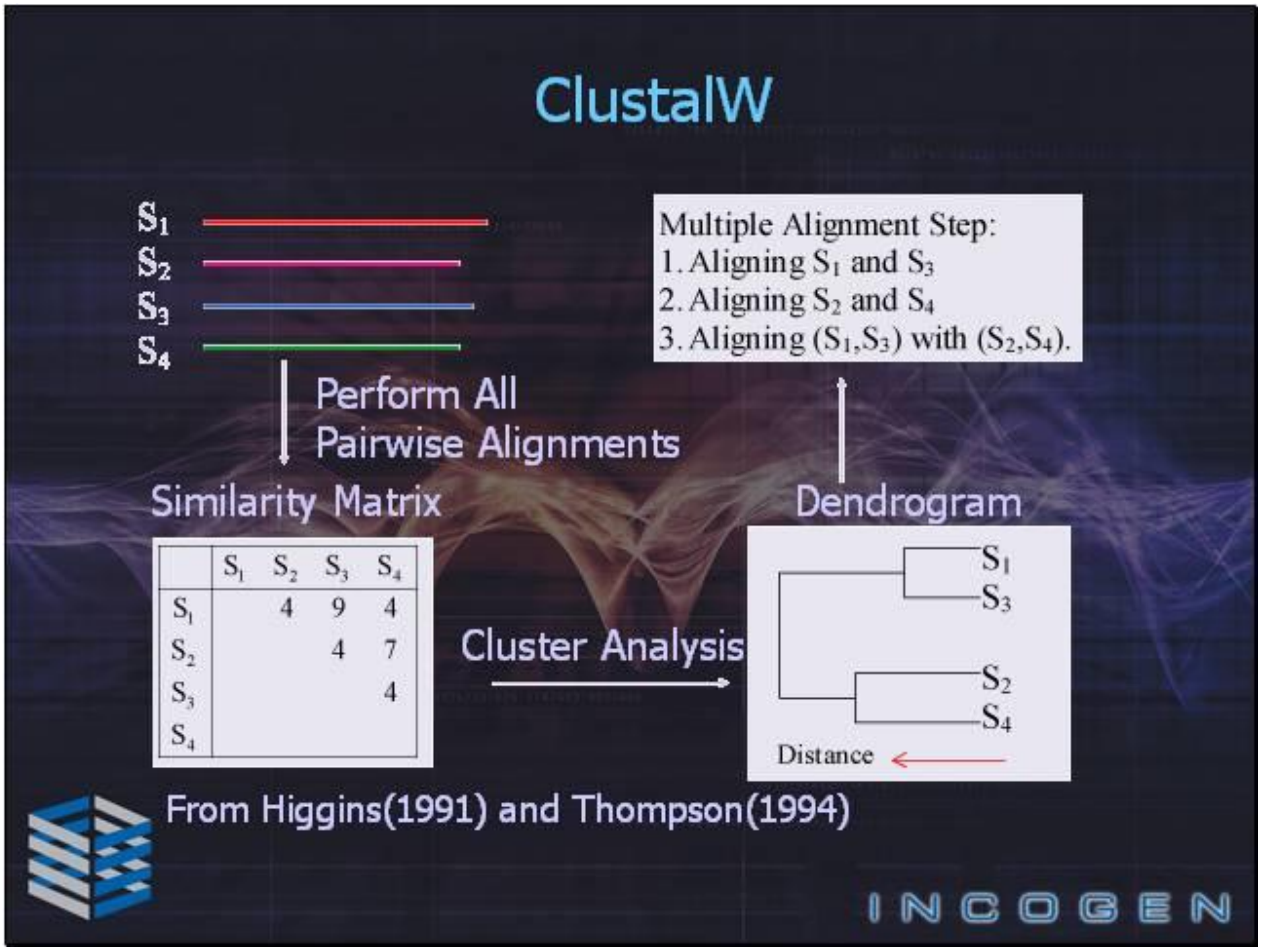
- Most popular multiple sequence alignment algorithm
  - Perform pairwise alignment between sequences, determine degrees of similarity between each pair, construct distance matrix
  - Construct a phylogenetic tree using the distance matrix and nearest-neighbor algorithm.
  - Combine the alignments starting from the most closely related groups to the most distantly related groups. The most closely-related pairs of sequences are aligned using dynamic programming
- Includes additional heuristics

INCOGEN

Slide notes

ClustalW is currently the most popular multiple sequence alignment algorithm. It performs a pairwise sequence alignment between all of the sequences and constructs a phylogenetic tree from the results. It then combines the results starting from the most closely related groups to the most distantly related groups, using dynamic programming to align the most closely related pairs of sequences.

Slide 30 - Slide 30



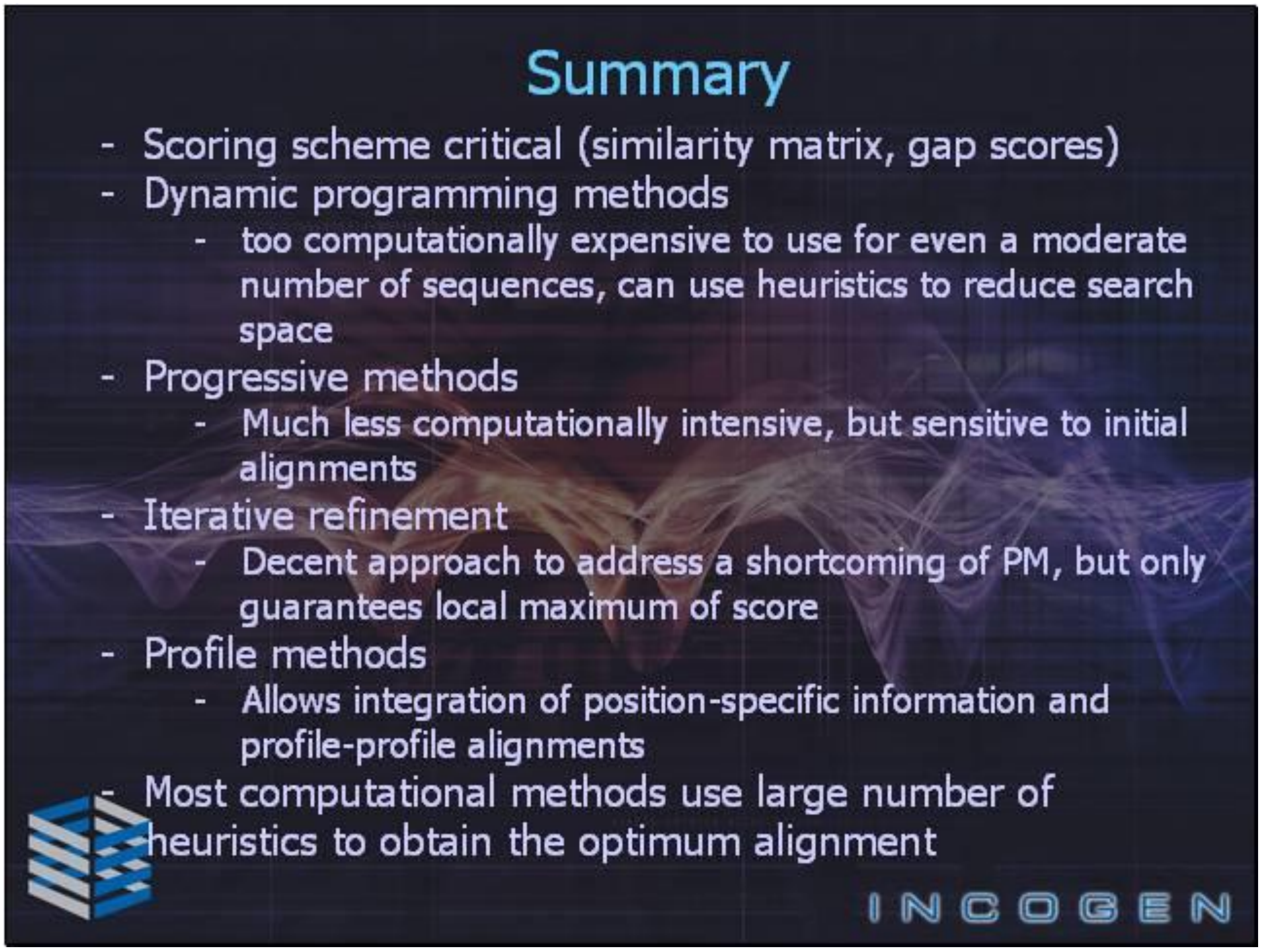
Multiple Alignment Step:
1. Aligning S_1 and S_3
2. Aligning S_2 and S_4
3. Aligning (S_1,S_3) with (S_2,S_4).

	S_1	S_2	S_3	S_4
S_1		4	9	4
S_2			4	7
S_3				4
S_4				

Slide notes

So, ClustalW takes a group of sequences and performs all pairwise alignments. It then calculates a similarity matrix, which it analyzes to see how distantly related the groups of sequences are. It then aligns the sequences and groups of sequences, aligning the most closely related at each step, until the MSA is complete.

Slide 31 - Summary



Summary

- Scoring scheme critical (similarity matrix, gap scores)
- Dynamic programming methods
  - too computationally expensive to use for even a moderate number of sequences, can use heuristics to reduce search space
- Progressive methods
  - Much less computationally intensive, but sensitive to initial alignments
- Iterative refinement
  - Decent approach to address a shortcoming of PM, but only guarantees local maximum of score
- Profile methods
  - Allows integration of position-specific information and profile-profile alignments
- Most computational methods use large number of heuristics to obtain the optimum alignment

INCOGEN

Slide notes

In summary, choosing the best scoring scheme is critical to the creation of a meaningful MSA. Dynamic programming methods, while guaranteeing an optimal alignment, are too computationally expensive to use for even moderate numbers of sequences, although there are heuristics that can be used to reduce the number of calculations. Progressive methods are much less computationally expensive, but they are very sensitive to initial alignments and may not produce good alignments, especially for distantly related sequences. Using an iterative approach improves the alignments produced by progressive methods, but it is still sensitive to the initial alignment. The profile methods allow integration of position-specific information and profile-profile alignments. In general, most computational methods use a large number of heuristics to obtain an optimal alignment.