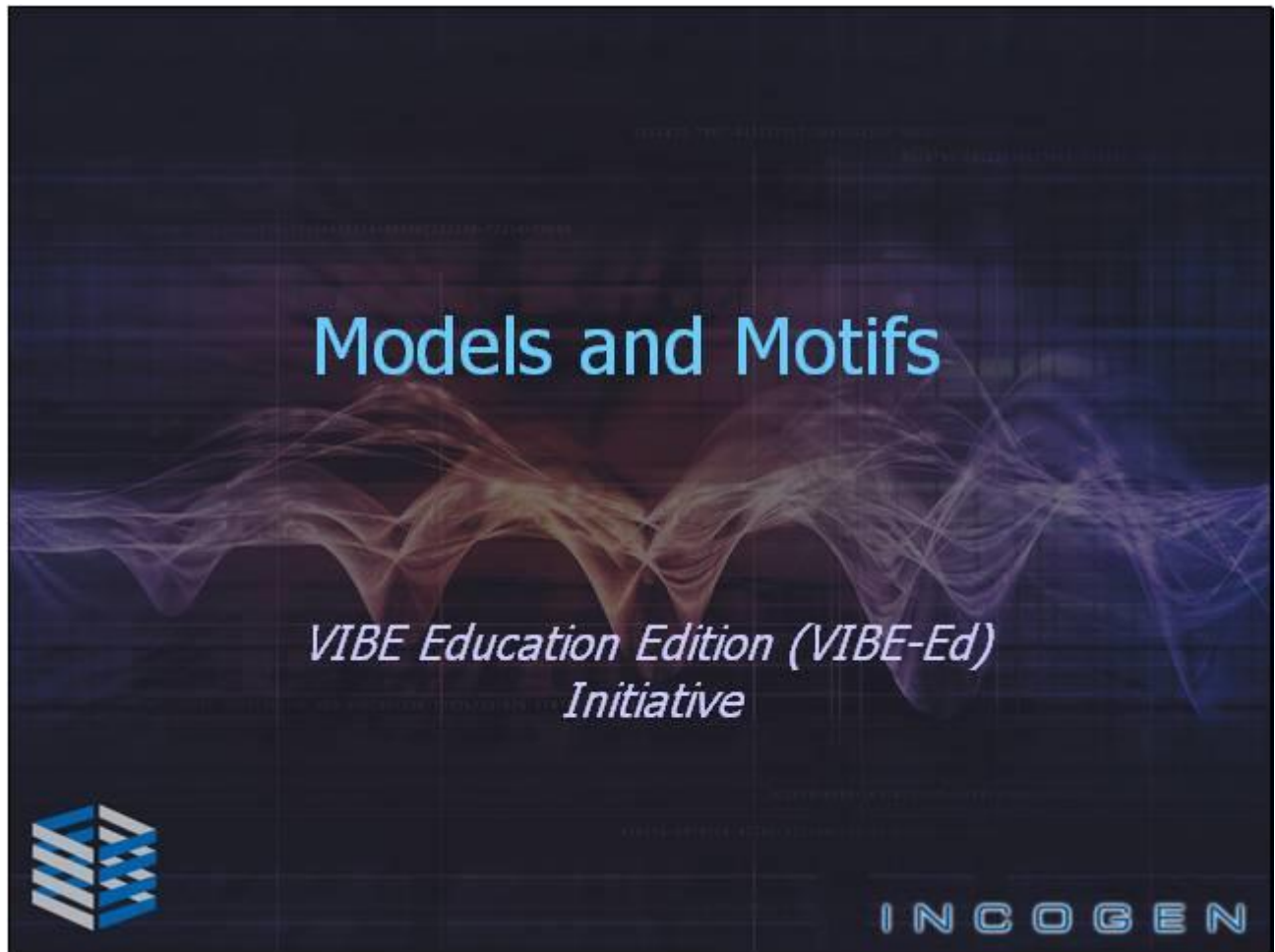


**Slide 1 - Models and Motifs**



**Slide notes**

This presentation is designed to give you an introduction to models and motifs.

## Beyond Pair-wise Sequence Comparison

- Often work with sequence fragments that we want to recognize and classify
- Use of consensus models built from multiple sequence alignments from protein families
- Consensus model can exploit additional information, such as the position and identity of residues that are more or less conserved throughout the family (insertions/deletions)





INCOGEN

### Slide notes

As geneticists, we often work with sequence fragments that we want to identify and classify. Consensus models, built from alignments of multiple sequences, can exploit additional information, such as position-specific information and identity of conserved residues, to help us accomplish that.

## Terminology

- **Domain:** An independently folded structural unit
- **Block:** ungapped multiple alignment of a conserved region of protein sequences
- **Conserved Pattern or Motif:** Highly similar region in an alignment of protein sequences
- **PSSM:** Position-Specific Scoring Matrix, also called **profile** or **model**, computed from each block
- Blocks, profiles, motifs, and patterns can be represented as special cases of the Hidden Markov Model (HMM) approach





### Slide notes

Let's start the discussion with a few definitions. A domain is an independently folded structural unit. A block is an ungapped multiple alignment of a conserved region of protein sequences. A conserved pattern or motif is a highly similar region in an alignment of protein sequences. A Position-Specific Scoring Matrix (PSSM), also called a profile or a model, can be calculated from each block. The blocks, profiles, motifs, and patterns can be represented as special cases of the Hidden Markov Model (HMM) approach.

# Conserved Regions in Protein Sequences - Local vs. Global

- Local
  - Pattern: short, simplest, but limited (no gaps)
  - Motif: conserved element of a sequence alignment, usually predictive of structural or functional region
- Global (across whole alignment)
  - Matrix
  - Profile
  - Hidden Markov Model



## Slide notes

Just as pairwise sequence alignments can be either global or local, multiple sequence alignments, or MSAs, can also be global or local. Patterns, which are short conserved regions containing no gaps, and motifs, which are conserved regions that typically represent structural or functional elements, are typically found by performing a local multiple sequence alignment. On the other hand, a profile analysis is performed by determining a global MSA and then removing the more highly conserved regions in the alignment into a separate, smaller MSA that can be used to search for other, related sequences. Similarly, the HMM approach begins with a global alignment and calculates probabilities for each position in the alignment that describe how likely that position is to be conserved in related sequences.

## Patterns

- Short, highly conserved regions
- Can be presented as regular expressions

Example:

**[AG]-x-V-x(2)-{YW}**

- [] shows either residue
- X is any residue
- X(2) any residue in the next 2 positions
- {} shows any residue except these



INCOGEN

### Slide notes

Short, highly conserved regions are called patterns. Patterns contain no insertions or deletions and can typically be represented with a regular expression.

## Patterns / Regular Expression (cont'd)


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

[AT] - [CG] - A - [ACGT] - [ACGT] - {G}, or  
[AT] - [CG] - A - [ACGT] - X - {G}, or  
[AT] - [CG] - A - X(2) - {G}

Three new sequences:

A	G	A	C	T	A	?
A	T	A	G	T	A	?
T	C	A	C	A	A	?

- The regular expression can:
  - Determine if a new sequence in question fits the criteria of the search or not



### Slide notes

For example, here is an alignment of five short sequences that we want to characterize using a regular expression. The first step is to look in each column of the alignment and write down what we see. For example, the third column contains only A's while the fourth column contains all of the residues. Then we begin to consolidate the regular expression into a more compact form. Now, let's compare some new sequences to see if they fit into this pattern. This first sequence matches. The second does not, since it does not have a C or a G as its second residue. The third sequence also matches the pattern. So, the regular expression is useful for finding sequences that fit the pattern.


## Patterns / Regular Expression (cont'd)

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

→

- The regular expression **cannot**:
  - Determine **how well** the new sequence in question fits the criteria of the search

Deal with regions containing **gaps/deletions**



INCOGEN

**Slide notes**

So, what happens if the regions we are aligning have different lengths; that is, the aligned region contains gaps and/or insertions. Unfortunately, regular expressions cannot deal with regions that contain gaps or insertions. Neither can it tell you how well the new sequence fits the search criteria.

## Position-Specific Scoring Matrix (Profiles, Motifs, Models)

- Position-specific table or matrix containing comparison information for aligned sequences
- Columns represent positions in sequences
- Rows contain score for alignment of position with each residue

Used to find sequences similar to alignment rather than one sequence



INCOGEN

### Slide notes

For this, we need a position-specific scoring matrix.. A position-specific scoring table or matrix contains comparison information about the aligned sequences. The columns in the table represent positions in the sequence, and the rows in the table contain the score for the alignment of the position with each residue. The table is then used to find sequences similar to the alignment.



Slide 9 - Example of a PSSM

## Example of a PSSM


F	K	L	L	S	H	C	L	L	V
F	K	A	F	G	Q	T	M	F	Q
Y	P	I	V	G	Q	E	L	L	G
F	P	V	V	K	E	A	I	L	K
F	K	V	L	A	A	V	I	A	D
L	E	F	I	S	E	C	I	I	Q
F	K	L	L	G	N	V	L	V	C

Alignment

A	-10	-10	-1	-6	0	-3	3	-10	-2	-6
C	22	33	10	10	22	20	22	24	19	7
D	-35	0	-32	-33	-7	6	-17	-34	-31	0
E	-27	15	-25	-26	-9	23	-9	-24	-23	-1
F	60	-30	12	14	-26	-20	-15	4	12	-20
G	-30	-20	-28	-32	28	-14	-23	-33	-27	-5
H	-13	-12	-25	-25	-16	14	-22	-22	-23	-10
I	3	-27	21	25	-29	-23	-8	33	19	-23
K	-26	25	-25	-27	-6	4	-15	-27	-26	0
L	14	-28	15	27	-27	-20	-9	33	26	-21
M	3	-15	10	14	-17	-10	-9	25	12	-11
N	-22	-6	-24	-27	1	8	-15	-24	-24	-4
P	-30	24	-21	-28	-14	-10	-22	-24	-26	-18
Q	-32	5	-25	-26	-9	24	-16	-17	-23	7
R	-18	9	-22	-22	-10	0	-18	-23	-22	-4
S	-22	-8	-16	-21	11	2	-1	-24	-10	-4
T	-10	-10	-6	-7	-5	-8	2	-10	-7	-11
V	0	-25	22	25	-10	-26	6	10	16	-16
W	9	-25	-18	-19	-25	-27	-34	-20	-17	-28
Y	34	-18	-1	1	-23	-12	-19	0	0	-18

Corresponding PSSM:  
Match values are  
higher for conserved  
residues


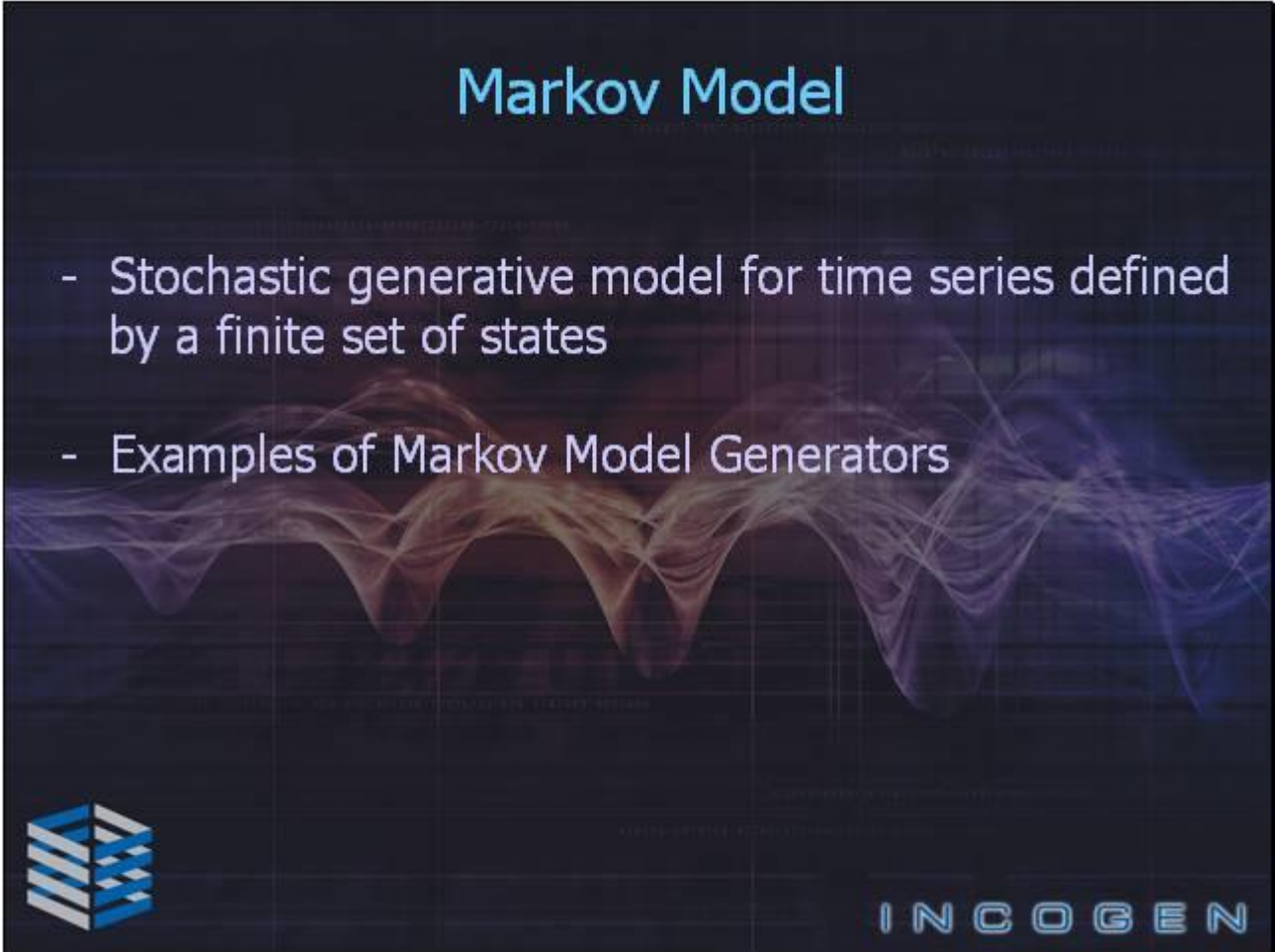


**Slide notes**

Here is an example of an alignment with its position-specific scoring matrix. Note that the values in the table are higher for conserved residues.

# Markov Model

- Stochastic generative model for time series defined by a finite set of states
- Examples of Markov Model Generators

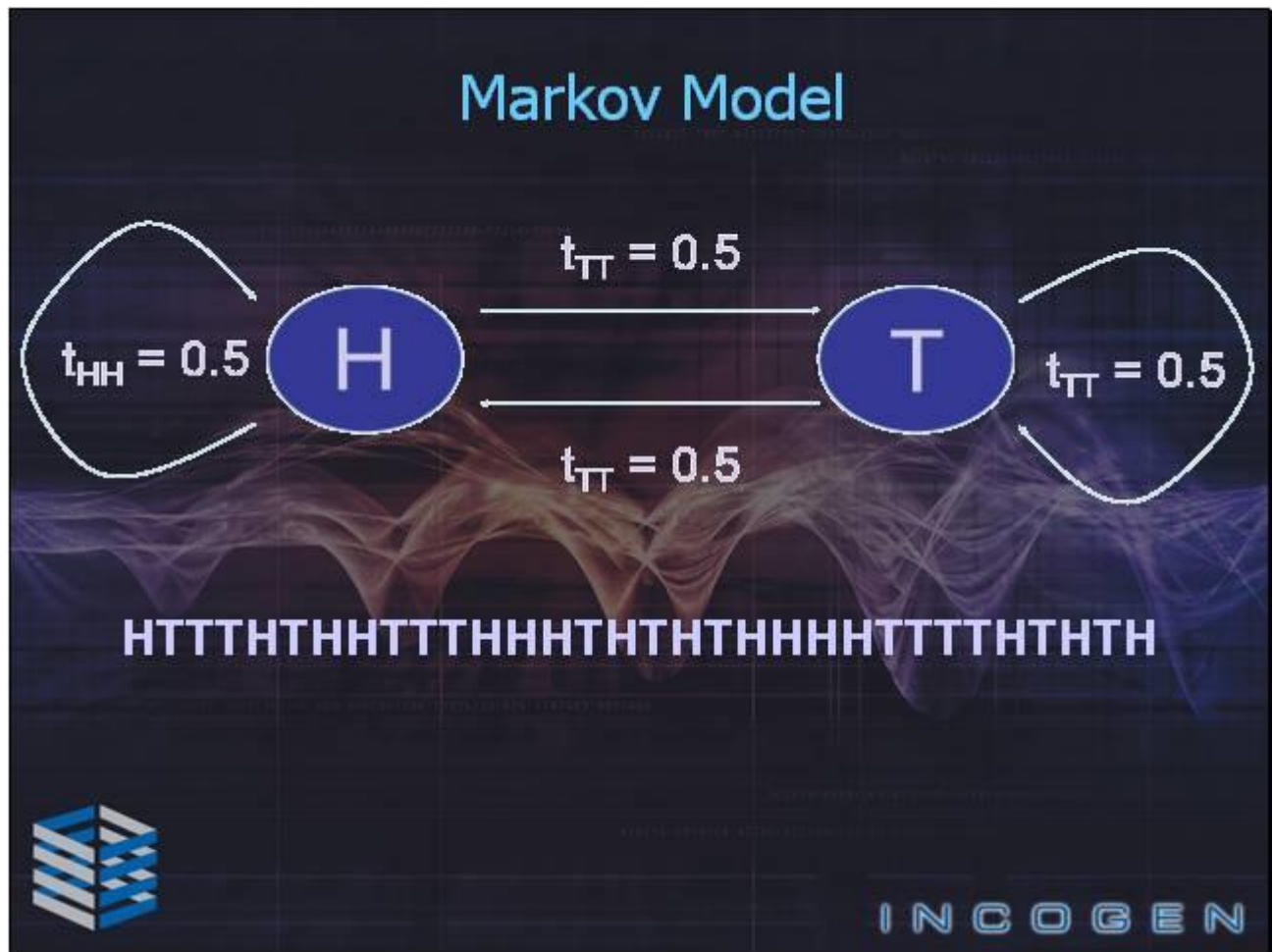


INCOGEN

**Slide notes**

The Markov model is a probabilistic generative model for a time series that is defined by a finite set of states.

Slide 11 - Markov Model example



**Slide notes**

Let's take an everyday example -- the flip of a coin. There are two possible states: Heads, denoted with an H, and Tails, denoted with a T. We have a probability of 0.5 of getting either heads or tails on any flip. A typical sequence this model could generate is presented at the bottom.

## Hidden Markov Model

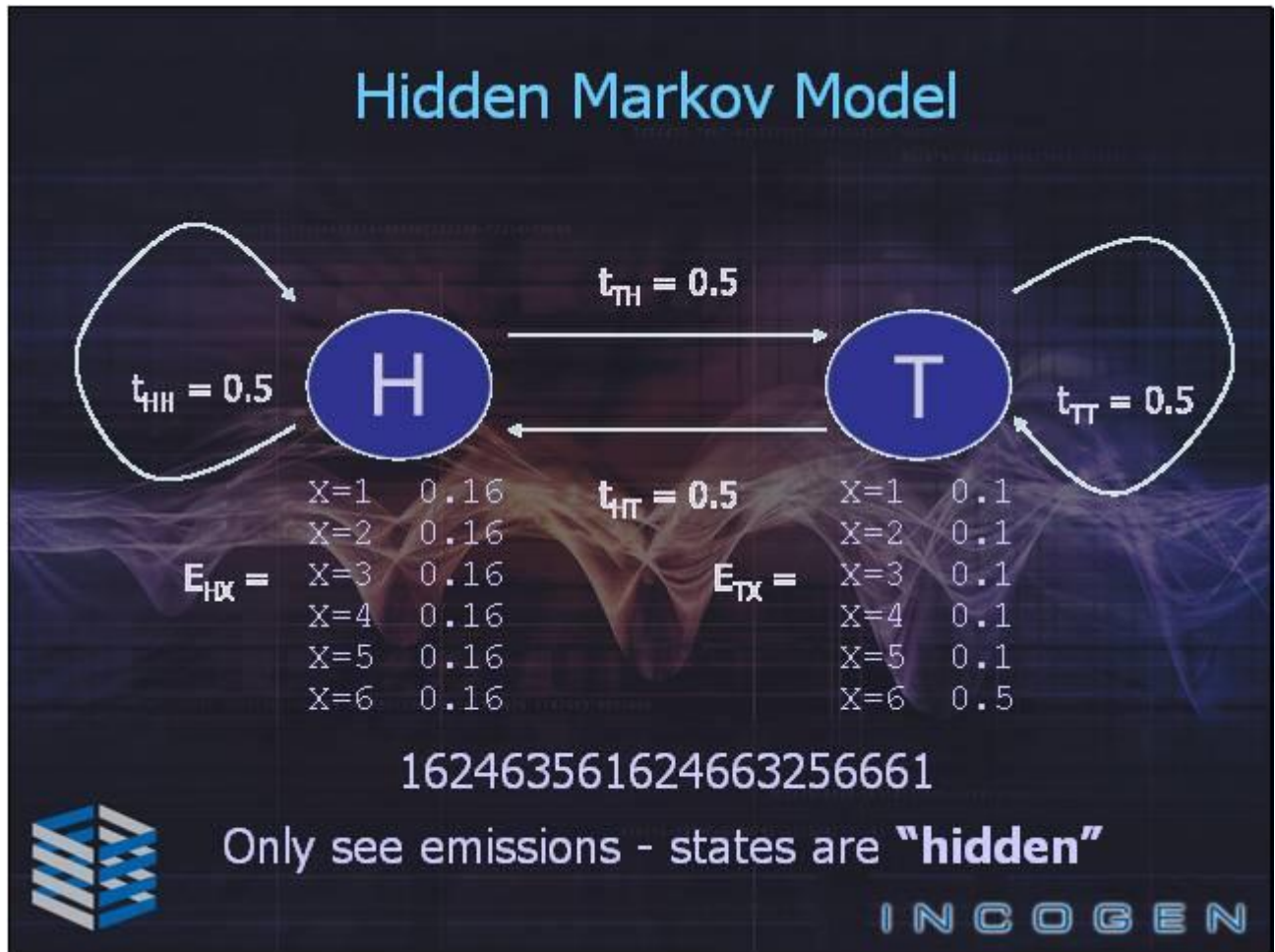
- Stochastic generative model for time series defined by a finite set of states, a discrete alphabet of symbols, a probability transition matrix  $T=(t_{ji})$ , and a probability emission matrix  $E=(e_{ix})$ .
- The system evolves from state to state while emitting symbols from the alphabet.
- When the system is in a given state  $i$ , it has probability  $t_{ji}$  of moving to state  $j$  and probability  $e_{ix}$  of emitting symbol  $X$ .



INCOGEN

### Slide notes

A Hidden Markov Model is slightly more complicated. It contains a "transition probability", that represents the probability of moving from one state to another, and an "emission probability", that represents the probability of generating specific symbols while in a particular state. So we have a discrete alphabet of symbols, a probability transition matrix, and a probability emission matrix. Unlike the Markov Model example we discussed earlier, a Hidden Markov Model does not have to emit a symbol when it moves from state to state, allowing us to incorporate gaps, and each "state" can emit a variety of symbols.





**Slide notes**

Let's look at the same diagram that we used earlier to examine the Markov Model. Now, in addition to the transition probabilities, there is an emission probability matrix for the Heads and the Tails states. Now, when we look at a typical sequence from this model, it is impossible to tell which state the model was in when each symbol was emitted; the states themselves are hidden.

## HMMs in Bioinformatics

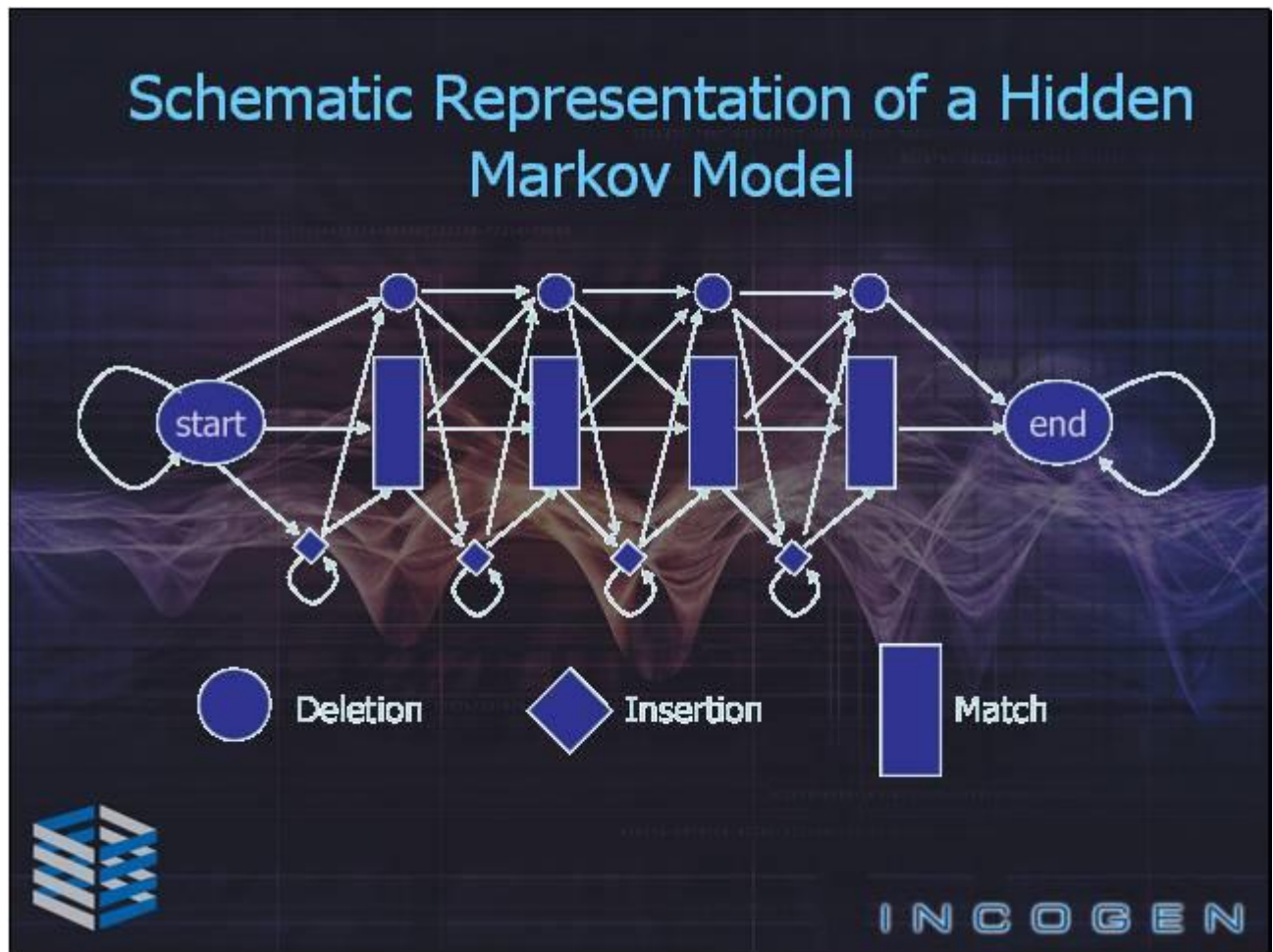
- An HMM can be used to model a family of sequences
- Gaps, insertions and deletions are allowed in the alignments
- Two possible alphabets: NT or AA
- Gives probabilities for each position
- Original software: HMMER
- Given a sequence, we can compute its probability of belonging to the family



### Slide notes

Hidden Markov Models, or HMMs, can be used to model a family of sequences. Gaps, insertions, and deletions are allowed. There are two possible "alphabets" for the model: nucleotide and amino acid. For a given alignment, the model will produce probabilities for each position. Then, we can use the model's probabilities to compute the probability of a sequence belonging the family represented by the model.

Slide 15 - Schematic Representation of a Hidden Markov Model



**Slide notes**

This diagram shows a schematic diagram of an HMM, including deletions, insertions, and up to four matches.

## Deriving the HMM Heuristics

- Start with an alignment of sequences
  - Each column in the alignment generates a state
  - Transition probabilities between states are determined by deletions and insertions
  - Emission probabilities at each state are determined by counting the occurrence of [ATGC] in each column
  - It's easier than it sounds - let's do a (simple) example for an alignment of NT sequences...

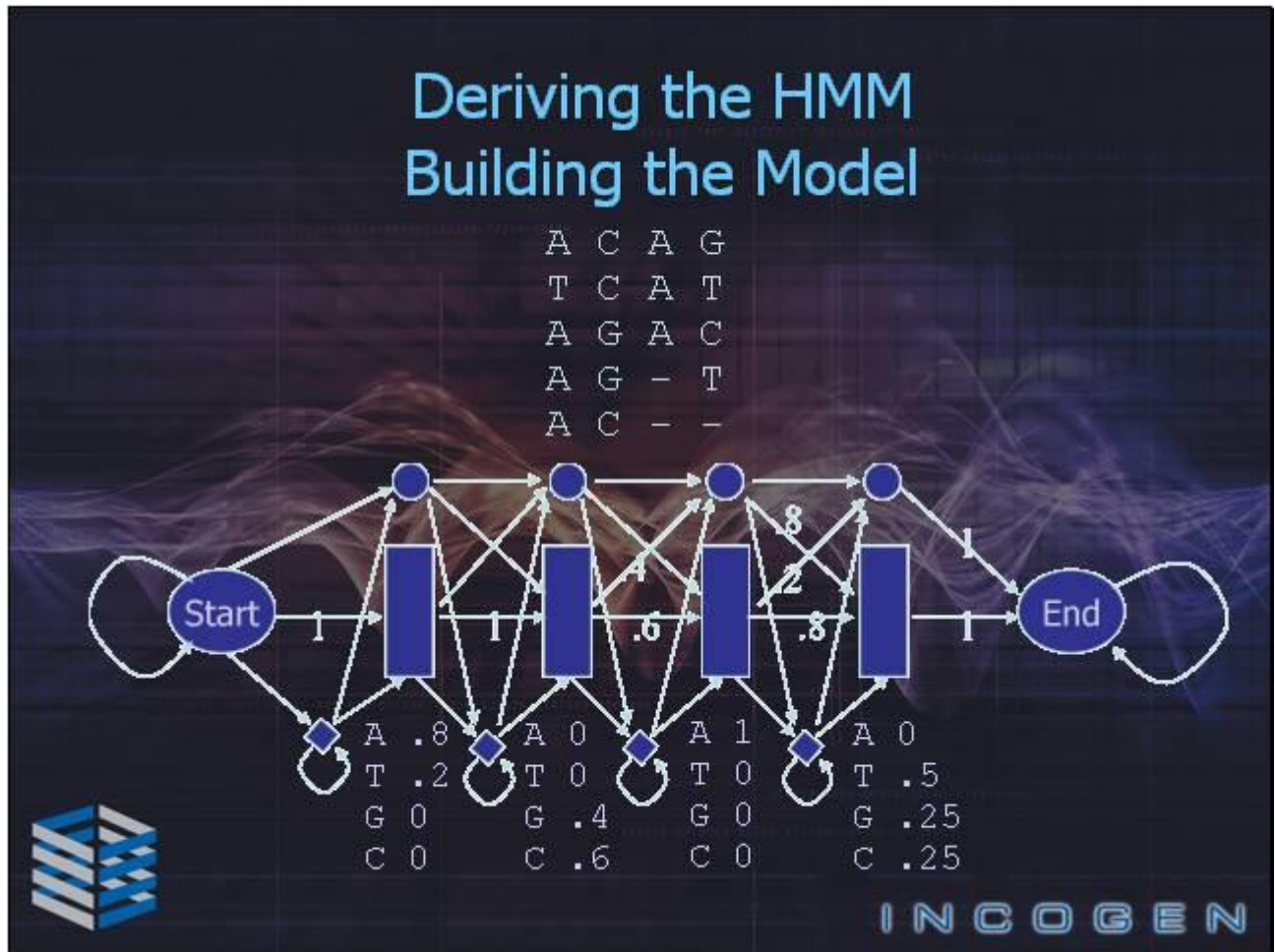


INCOGEN

### Slide notes

To derive an HMM, we first must start with a multiple sequence alignment. Each column in the alignment generates a state in the model. Transition probabilities are determined using deletions and insertions. Emission probabilities at each state are determined by counting the occurrence of each character in each column.





**Slide notes**


Let's try an example. Here is our alignment. Let's begin drawing the model. First, let's draw the start state, the four possible match states, and then the end state. Next, we'll follow with the insertion states, each one of which can occur multiple times before moving to a match state. Next come the deletion states and all of the state transitions that can occur. Next, we'll use the MSA to add transition probabilities, and finally, we'll add the emission probabilities for each state.

## Deriving the HMM Emission Probabilities

	A	C	G	T
Pos 1	0.8	0	0	0.2
Pos 2	0	0.6	0.4	0
Pos 3	1	0	0	0
Pos 4	0	0.25	0.25	0.5

The diagram illustrates an HMM with the following emission probabilities for each hidden state:

- State 1: A: 0.8, T: 0.2, G: 0, C: 0
- State 2: A: 0, T: 0, G: 0.4, C: 0.6
- State 3: A: 1, T: 0, G: 0, C: 0
- State 4: A: 0, T: 0.5, G: 0.25, C: 0.25

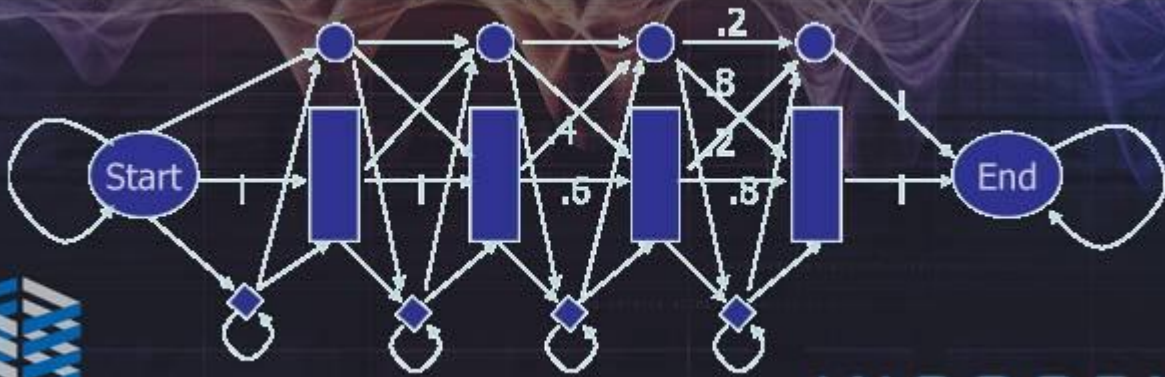


**Slide notes**

These emission probabilities can be stored in a table,

## Deriving the HMM Transition Probabilities

	m->m	m->i	m->d	i->m	i->i	i->d	d->m	d->d	s->m	m->e
Trans 1	0	0	0	0	0	0	0	0	1	0
Trans 2	1	0	0	0	0	0	0	0	0	0
Trans 3	0.6	0	0.4	0	0	0	0	0	0	0
Trans 4	0.8	0	0.2	0	0	0	0.8	0.2	0	0
Trans 5	0	0	0	0	0	0	0	0	0	1



INCOGEN



### Slide notes

as can the transition probabilities. In this table, "m" stands for match, "i" stands for insertion, "d" stands for deletion, "s" stands for start, and "e" stands for end.

## Deriving the HMM

### Constructing/Training the HMM

- EM algorithm (Baum-Welch)
  - Random or heuristic MSA (e.g. ClustalW)
  - Number of match states is number of conserved columns
  - Two-Stage, iterative process:
    - M step: Aligned residues give match state distributions
    - E step: Given this model, realign all the sequences to the model
    - Repeat until convergence
- Viterbi algorithm
  - Make a matrix with rows for sequence elements and columns for states in the model
  - Work backwards row by row, calculating the probability for each state to have emitted that element and putting that probability in a cell.
    - When there are multiple paths, select the highest probability one and store which path was selected.
  - Next row uses results of previous row.
  - Best end probability identifies best total path.
- "Surgery" algorithm
  - Dynamic adjustment of HMM length



#### Slide notes



Since we can't see the states themselves but only what they emit, we must create the HMM using our best guess as to what the states are. There are multiple algorithms that can be used to construct and train an HMM. The first we'll discuss is the EM, or

Baum-Welch, algorithm. This algorithm uses a random MSA, such as the one constructed by ClustalW. The number of match states is defined as the number of conserved columns. This algorithm uses a two stage, iterative process; first, it uses the aligned residues to build a model, and then it realigns the sequences to the model. These two steps are repeated until the model no longer changes. A second algorithm called the Viterbi algorithm uses dynamic programming to calculate the best estimate of the emission probabilities. A third, called the "Surgery" algorithm, uses dynamic adjustments of the HMM length to calculate the best models.

## Using the HMM

### Scoring (aligning) a sequence against a model

- Estimate the probability of sequence  $s$ , given model  $m$ ,  $P(s|m)$ 
  - Multiply probabilities along most likely path (or add logs - less numeric error)
  - Other paths are negligible
- Often expressed as *negative log likelihood* score:  $-\log[P(s|m)]$
- Score is dependent on length of  $s$  and  $m$ .
- Need some way to assess significance!





#### Slide notes

After we create the HMM, we can use it to search for sequences that are similar to the ones used to create the HMM. So, we use the model to estimate the probability that a sequence belongs to the model. This probability is often expressed as negative log likelihood score and is dependent on the length of the sequence and the length of the model. Once we have a score, we need a way to assess its significance, just as we did for pairwise sequence alignments.

# Similarity Searches with HMM

- HMMSearch
  - Query: HMM
  - Target: AA (e.g., Swissprot)
- HMMScan
  - Query: AA
  - Target: HMM (e.g., Pfam)



**Slide notes**

There are two similarity searches that use HMMs. The first, HMMSearch, uses the HMM as the query and searches an amino acid database for sequences that match the model. The second, HMMScan, uses an amino acid as the query and compares it to an HMM.

## Slide 23 - Using HMMs in VIBE

# Using HMMs in VIBE

The screenshot displays the INCOGEN VIBE 3.0 software interface. The main window features a central workflow diagram on a grid. The workflow consists of several steps: 1. RA Seq (represented by a red cup icon), 2. HMM (represented by a purple square icon, highlighted with a yellow border), 3. HMM (represented by a green square icon), 4. HMM (represented by a green square icon), 5. HMM (represented by a blue square icon), and 6. HMM (represented by a blue square icon). The workflow is connected by arrows. On the right side of the interface, there is a panel with various settings and options, including 'DIAGONALS', 'PERMISSIVITY', 'GAP DISTANCE', 'NO RESIDUE GAPS', 'HMM RESIDUE', 'MULTIPLE', 'EMITS', 'ORDEREDLY', 'REVERSE', 'TOP DIAGONALS', and 'TRANSITION WEIGHT'. The INCOGEN logo is visible in the bottom right corner of the slide.

### Slide notes

Both of these similarity searches are available in VIBE.

## Local vs. global HMM scoring

- We can insist that an entire sequence aligns to model, or global scoring.
- Can add "free insertions" at beginning and end, equivalent to penalty-free end gaps.
  - Recognizes a region within a longer sequence
- Can also find highest scoring subregion of sequence (local scoring)
  - Already available from Viterbi calculation, so no additional computational cost.
- Can model domains, as well as whole proteins



INCOGEN

### Slide notes

We can use HMMs for a variety of purposes. We can insist on a global alignment between the sequence and the model. We can allow gaps only at the beginning and end of the HMM so that we can find regions within longer sequences. We can allow local alignments in addition to global alignments. We can also choose to model different lengths of sequences—from domains to whole proteins.



## Applications of HMMs

- Protein sequence applications:
  - MSAs and identifying distant homologs  
E.g. Pfam uses HMMs to define its MSAs
  - Domain definitions
  - Used for fold recognition in protein structure prediction
- Nucleotide sequence applications:
  - Models of exons, genes, etc. for gene recognition.





INCOGEN

### Slide notes

Amino acid HMMs are used to identify multiple sequence alignments and distant homologs, define domains, and recognize folds in protein structure prediction. Nucleotide HMMs are used to model exons and genes; the models are used when trying to recognize genes.

## Advantages of HMMs

- Built on a formal probabilistic basis
- Allows more sensitive searching
- Can use probability theory to guide the scoring parameters
- Probability theory allows a HMM to be trained from unaligned sequences if alignment not known or trusted
- Consistent theory behind gap/insertion penalties
- Less skill and intervention needed to train a good HMM vs. hand constructed profile
  - Can make libraries of hundreds of profile HMMs and apply them on a large scale (**whole genome**)






### Slide notes

HMMs have some distinct advantages. They are built on a formal probabilistic basis. They also allow for more sensitive searching than pairwise alignments do. The probabilistic theory can be used to guide the scoring parameters; the theory also allows us to train an HMM using unaligned sequences if we don't know or trust the alignment. There is a consistent theory behind gap and insertion penalties. Finally, less skill and human intervention is needed to train a good HMM than to train a hand-constructed profile. Because of this, we can create libraries of hundreds of profile HMMs and apply them on a large scale.

## Drawbacks of HMMs

- Do not capture higher-order correlations
  - Assumes identity of a particular position is independent of the identity of all other positions



**Slide notes**

Its one major drawback is that its probabilistic basis does not capture higher-order correlations between sequence positions. It assumes that the identity of a particular position is independent of the identity of all of the other positions in the sequence, and we know that not to be true.