Statistical Modelling and Methods in Bioinformatics

…

Loyola University Chicago
Department of Microbiology and Immunology
Timothy E. O’Brien

10th March 2017

Why Replicate Our Studies?

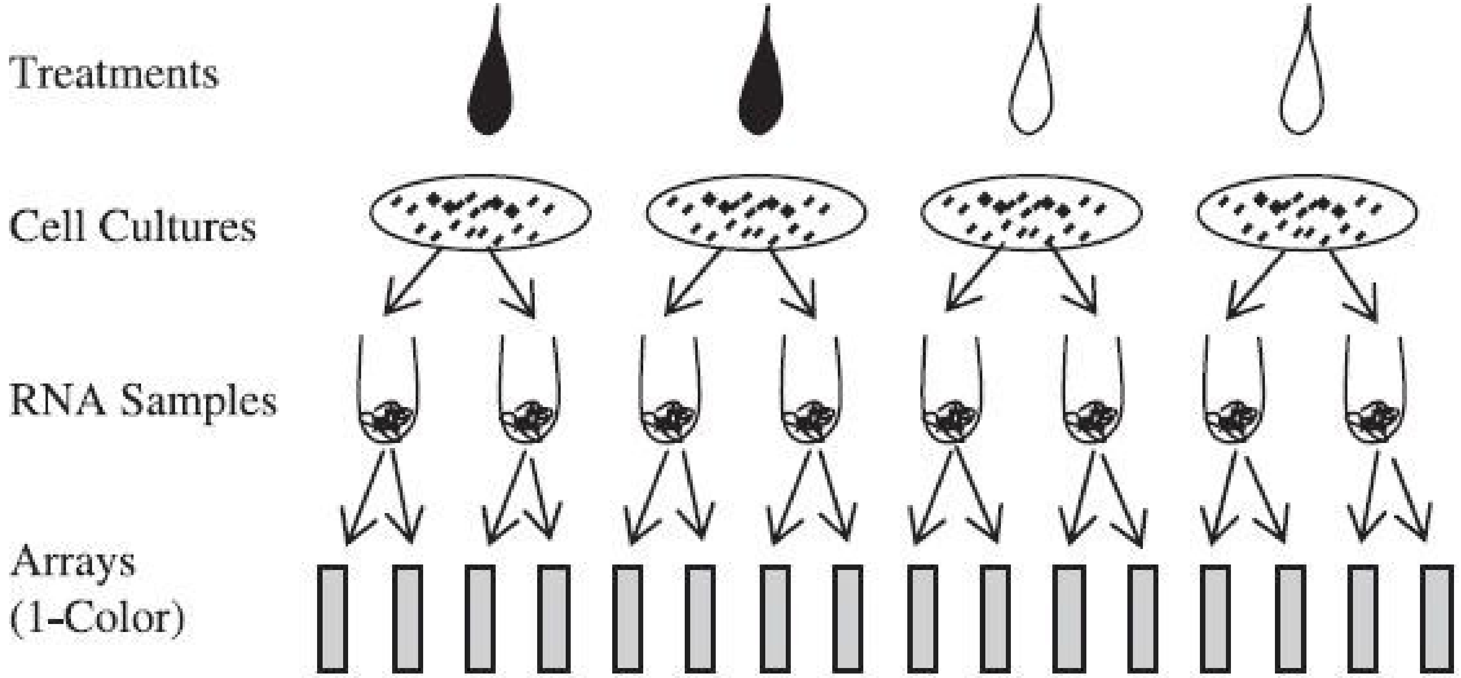
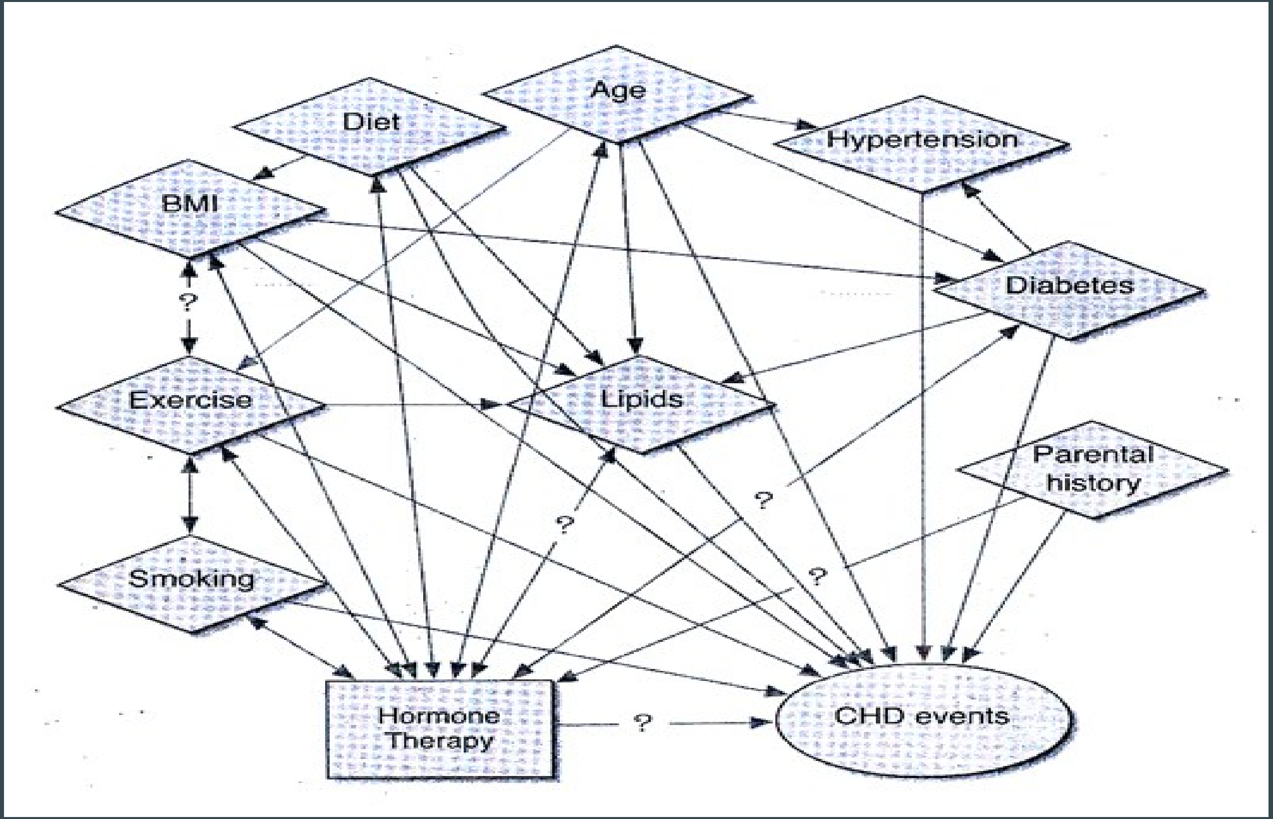


Figure 9.1 Different levels of replication in a microarray experiment.

“In this experiment, the independent cell cultures are biological replicates. The replicates at RNA samples and arrays are technical replicates which are similar to the repeated measurements. They are less useful for identifying significantly expressed genes between the two treatments. However, technical replicates are essential in experiments designed for evaluating the technology and in identifying the sources of variation. The variability between the duplicated arrays estimates the variability of the procedure after RNA extraction and the variability between the duplicated RNA samples estimates the variability from both RNA extraction and the array hybridization.” (Lee, pp.203-4)

↑ Power: Statistical Modelling May Help

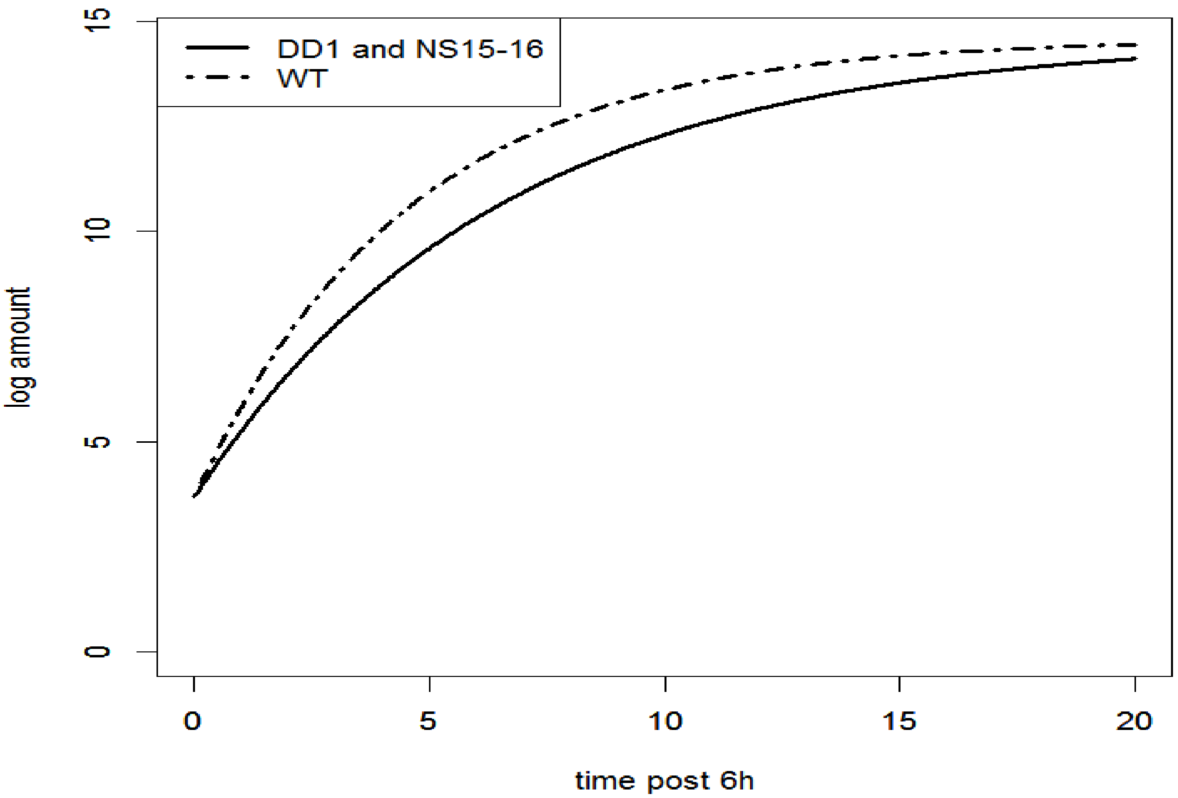


The Marginal Table:

	Yes	No	Total	% diseased
M	53	430	483	11.0%
F	15	176	191	7.9%

The Stratified Tables for the Same Data:

Younger patients						Older patients				
	Disease status						Disease status			
	Yes	No	Total	% diseased			Yes	No	Total	% diseased
M	53	414	467	11.3%		M	0	16	16	0.0%
F	11	37	48	22.9%		F	4	139	143	2.8%

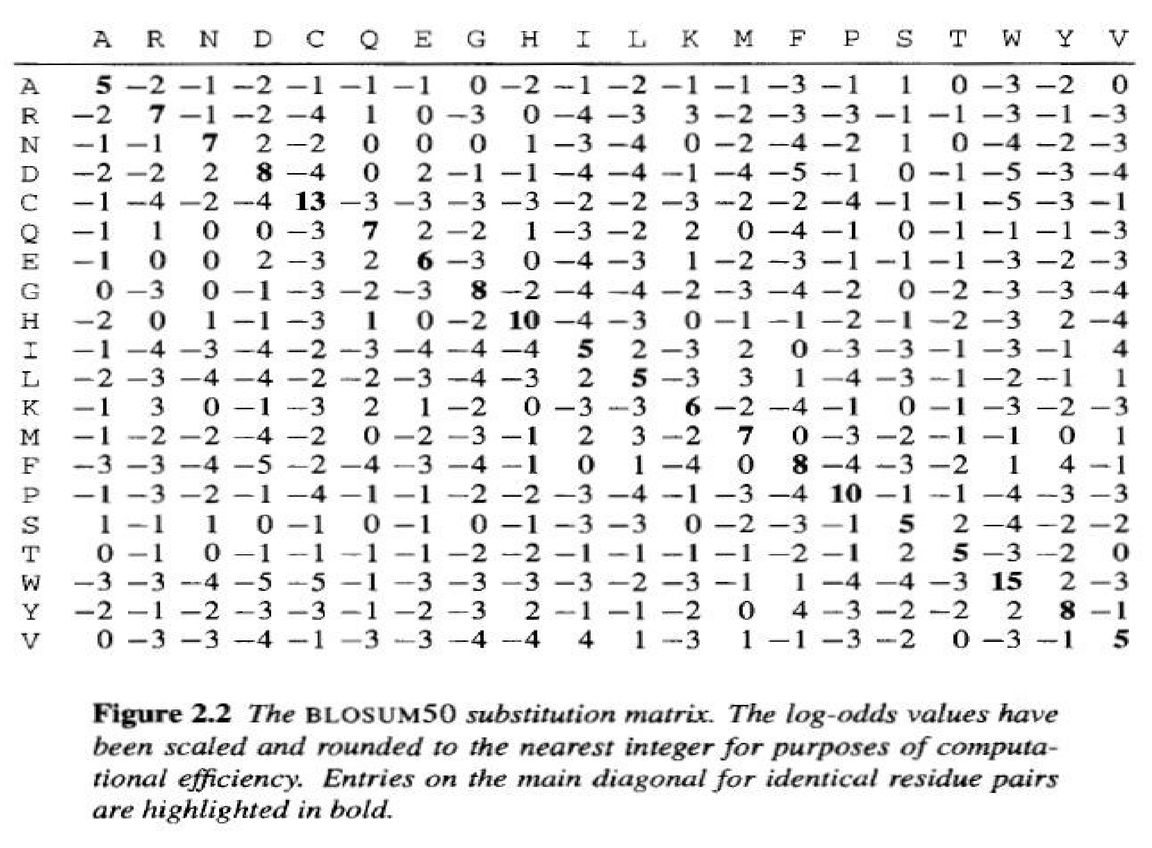

DD1 and NS15-16
WT
log amount
0
5
10
15
0
5
10
15
20
time post 6h

DNA/Protein Sequence Alignment Methods

- Traditional Methods:
  - Global: Needleman/Wunsch method
  - Local: Smith/Waterman algorithm
  - End-Space Free

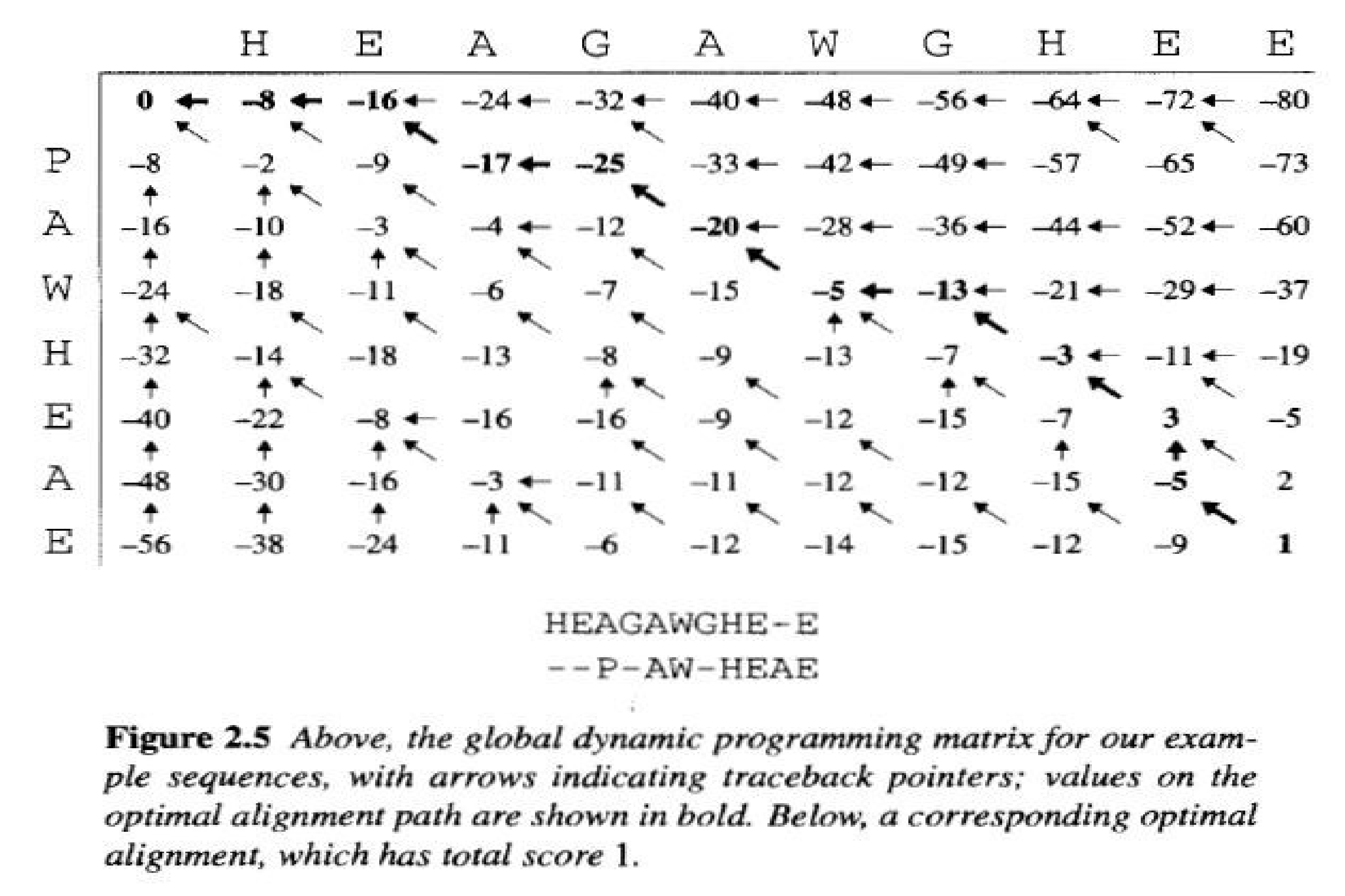
- Modern Methods using Hidden Markov Models

Needleman/Wunsch Global Algorithm:



	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	**5**	−2	−1	−2	−1	−1	−1	0	−2	−1	−2	−1	−1	−3	−1	1	0	−3	−2	0
R	−2	**7**	−1	−2	−4	1	0	−3	0	−4	−3	3	−2	−3	−3	−1	−1	−3	−1	−3
N	−1	−1	**7**	2	−2	0	0	0	1	−3	−4	0	−2	−4	−2	1	0	−4	−2	−3
D	−2	−2	2	**8**	−4	0	2	−1	−1	−4	−4	−1	−4	−5	−1	0	−1	−5	−3	−4
C	−1	−4	−2	−4	**13**	−3	−3	−3	−3	−2	−2	−3	−2	−2	−4	−1	−1	−5	−3	−1
Q	−1	1	0	0	−3	**7**	2	−2	1	−3	−2	2	0	−4	−1	0	−1	−1	−1	−3
E	−1	0	0	2	−3	2	**6**	−3	0	−4	−3	1	−2	−3	−1	−1	−1	−3	−2	−3
G	0	−3	0	−1	−3	−2	−3	**8**	−2	−4	−4	−2	−3	−4	−2	0	−2	−3	−3	−4
H	−2	0	1	−1	−3	1	0	−2	**10**	−4	−3	0	−1	−1	−2	−1	−2	−3	2	−4
I	−1	−4	−3	−4	−2	−3	−4	−4	−4	**5**	2	−3	2	0	−3	−3	−1	−3	−1	4
L	−2	−3	−4	−4	−2	−2	−3	−4	−3	2	**5**	−3	3	1	−4	−3	−1	−2	−1	1
K	−1	3	0	−1	−3	2	1	−2	0	−3	−3	**6**	−2	−4	−1	0	−1	−3	−2	−3
M	−1	−2	−2	−4	−2	0	−2	−3	−1	2	3	−2	**7**	0	−3	−2	−1	−1	0	1
F	−3	−3	−4	−5	−2	−4	−3	−4	−1	0	1	−4	0	**8**	−4	−3	−2	1	4	−1
P	−1	−3	−2	−1	−4	−1	−1	−2	−2	−3	−4	−1	−3	−4	**10**	−1	−1	−4	−3	−3
S	1	−1	1	0	−1	0	−1	0	−1	−3	−3	0	−2	−3	−1	**5**	2	−4	−2	−2
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	2	**5**	−3	−2	0
W	−3	−3	−4	−5	−5	−1	−3	−3	−3	−3	−2	−3	−1	1	−4	−4	−3	**15**	2	−3
Y	−2	−1	−2	−3	−3	−1	−2	−3	2	−1	−1	−2	0	4	−3	−2	−2	2	**8**	−1
V	0	−3	−3	−4	−1	−3	−3	−4	−4	4	1	−3	1	−1	−3	−2	0	−3	−1	**5**

Figure 2.2 *The* BLOSUM50 *substitution matrix. The log-odds values have been scaled and rounded to the nearest integer for purposes of computational efficiency. Entries on the main diagonal for identical residue pairs are highlighted in bold.*

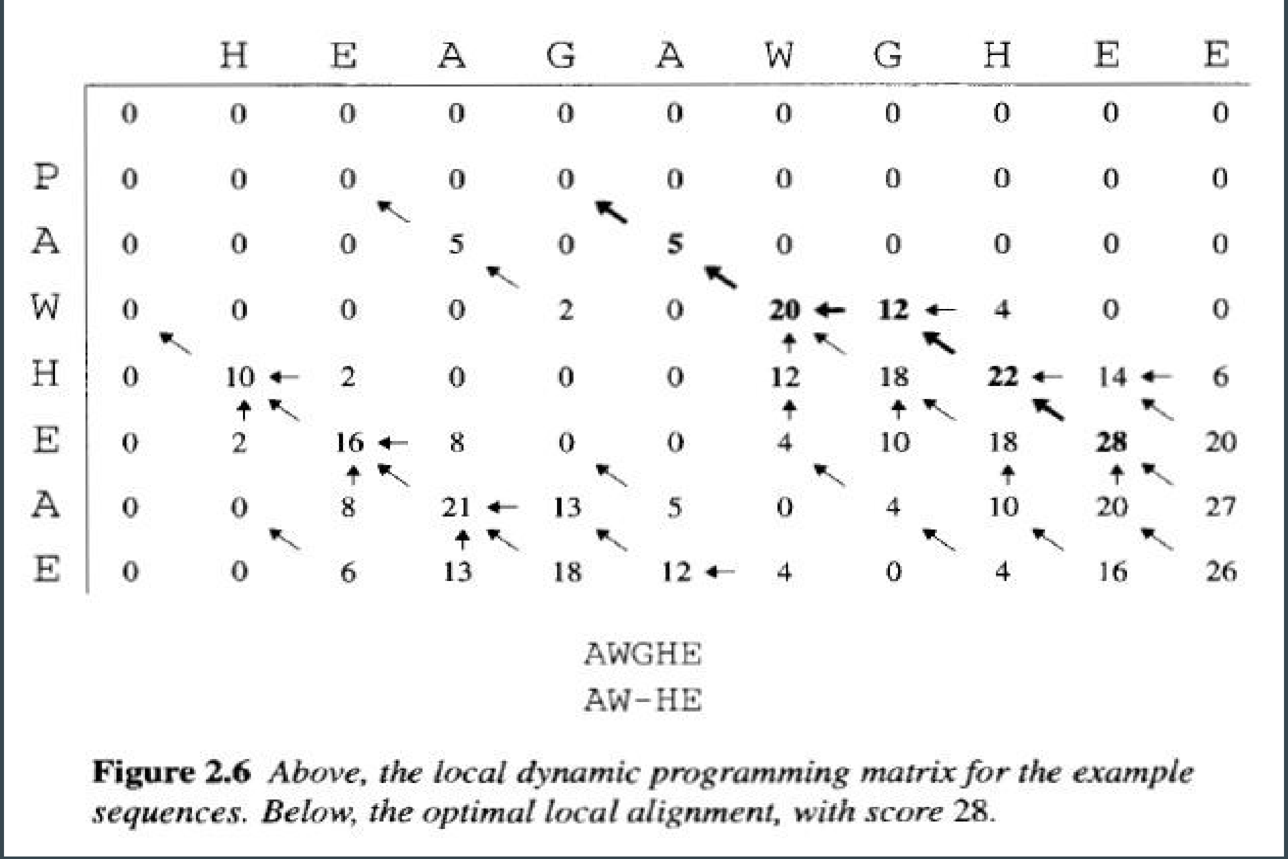


		H	E	A	G	A	W	G	H	E	E
	0	**–8** ←	**–16** ←	–24 ←	–32 ←	–40 ←	–48 ←	–56 ←	–64 ←	–72 ←	–80 ←
P	–8	–2 ↖	–9 ↖	**–17** ↖	**–25** ←	–33 ↖	–42 ←	–49 ←	–57 ←	–65 ↖	–73 ↖
A	–16 ↑	–10 ↑	–3 ↖	–4 ↖	–12 ←	**–20** ↖	–28 ←	–36 ←	–44 ←	–52 ←	–60 ←
W	–24 ↑	–18 ↑	–11 ↑	–6 ↖	–7 ↖	–15 ↖	**–5** ↖	**–13** ←	–21 ←	–29 ←	–37 ←
H	–32 ↑	–14 ↖	–18 ↖	–13 ↖	–8 ↖	–9 ↖	–13 ↑	–7 ↖	**–3** ↖	–11 ←	–19 ←
E	–40 ↑	–22 ↑	–8 ↖	–16 ←	–16 ↑	–9 ↖	–12 ↖	–15 ↑	–7 ↖	**3** ↖	–5 ↖
A	–48 ↑	–30 ↑	–16 ↑	–3 ↖	–11 ←	–11 ↖	–12 ↖	–12 ↖	–15 ↑	**–5** ↑	2 ↖
E	–56 ↑	–38 ↑	–24 ↑	–11 ↑	–6 ↖	–12 ↖	–14 ↖	–15 ↖	–12 ↖	–9 ↖	**1** ↖

```
HEAGAWGHE-E
--P-AW-HEAE
```

Figure 2.5 *Above, the global dynamic programming matrix for our example sequences, with arrows indicating traceback pointers; values on the optimal alignment path are shown in bold. Below, a corresponding optimal alignment, which has total score* 1.

Smith/Waterman Local Alignment Method:

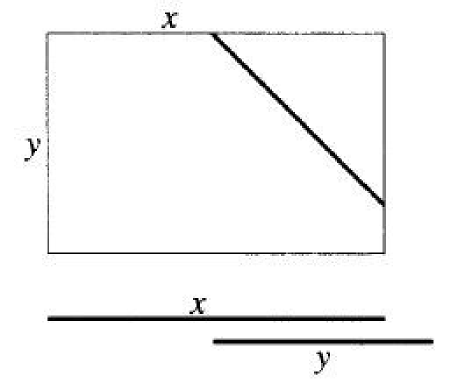


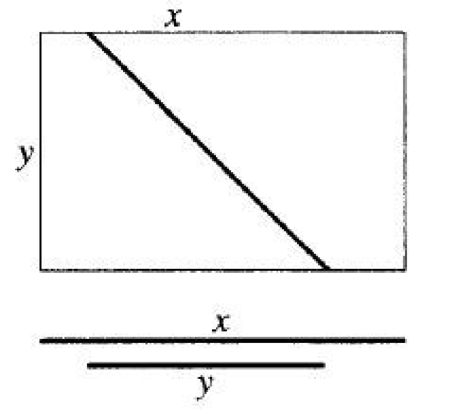
		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	5	0	**5**	0	0	0	0	0
W	0	0	0	0	2	0	**20**	**12**	4	0	0
H	0	10	2	0	0	0	12	18	**22**	14	6
E	0	2	16	8	0	0	4	10	18	**28**	20
A	0	0	8	21	13	5	0	4	10	20	27
E	0	0	6	13	18	12	4	0	4	16	26

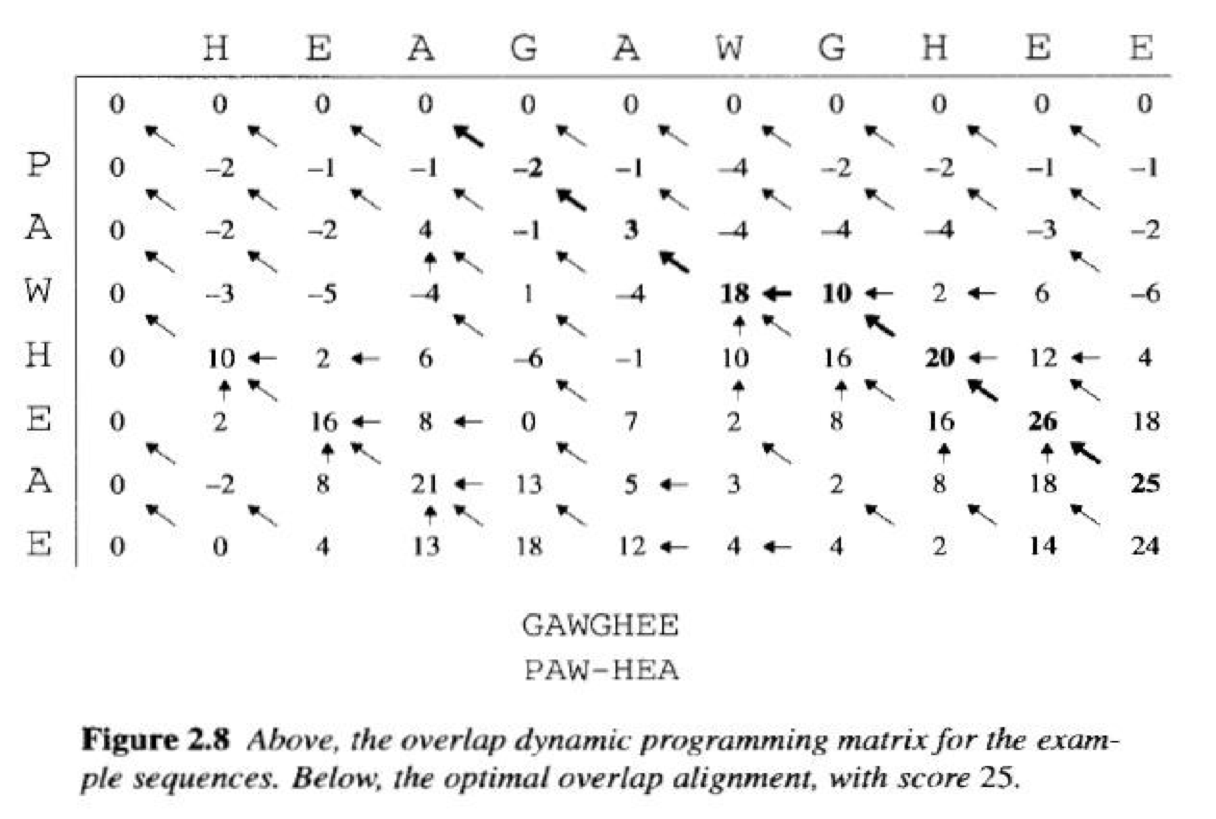
```
AWGHE
AW-HE
```

Figure 2.6 *Above, the local dynamic programming matrix for the example sequences. Below, the optimal local alignment, with score 28.*

End-Space Free Alignment Algorithm:







		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	−2	−1	−1	**−2**	−1	−4	−2	−2	−1	−1
A	0	−2	−2	4	−1	**3**	−4	−4	−4	−3	−2
W	0	−3	−5	−4	1	−4	**18** ←	**10** ←	2 ←	6	−6
H	0	10 ←	2 ←	6	−6	−1	10	16	**20** ←	12 ←	4
E	0	2	16 ←	8 ←	0	7	2	8	16	**26**	18
A	0	−2	8	21 ←	13	5 ←	3	2	8	18	**25**
E	0	0	4	13	18	12 ←	4 ←	4	2	14	24

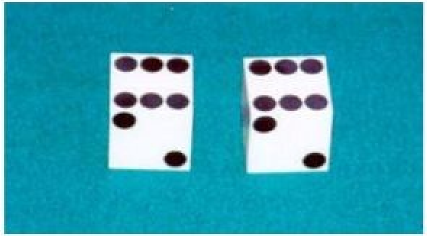
```
GAWGHEE
PAW-HEA
```

Figure 2.8 *Above, the overlap dynamic programming matrix for the example sequences. Below, the optimal overlap alignment, with score 25.*

Using Hidden Markov Models (HMM) for Sequence Alignment:

- Uses the EM (Expectation/Maximization) Algorithm
- http://www.nature.com/nbt/journal/v22/n10/pdf/nbt1004-1315.pdf

Example: The Occasionally Dishonest Casino



A casino has two dice:

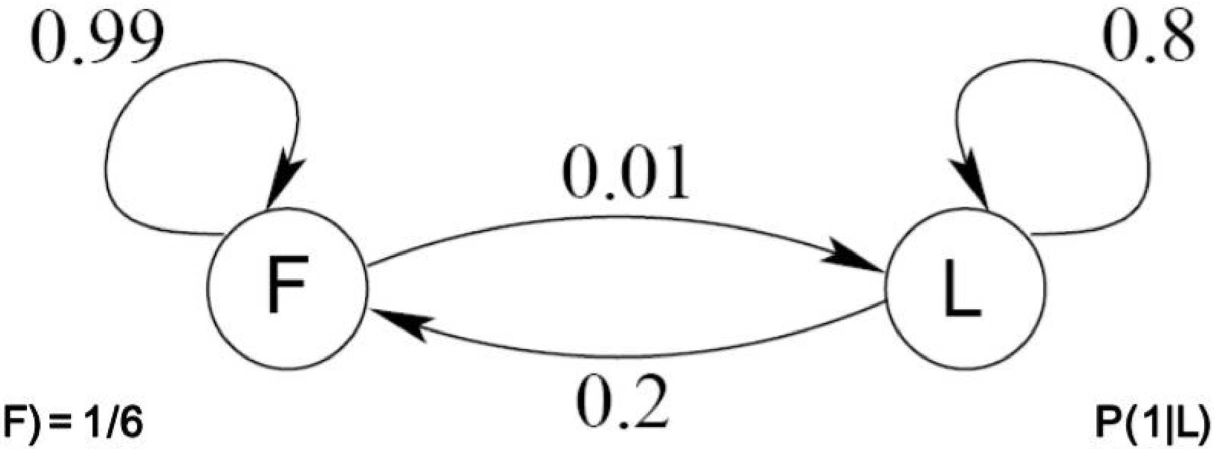
- Fair die:
  P(1) = P(2) = P(3) = P(4) = P(5) = P(6) = 1/6
- Loaded die:
  P(1) = P(2) = P(3) = P(4) = P(5) = 1/10; P(6) = 1/2

- Dealer switches between dice as:
  - Prob(Fair → Loaded) = 0.01
  - Prob(Loaded → Fair) = 0.2
  - Transitions between dice obey a Markov process

Game:

1. You bet $1
2. You roll (always with a fair die)
3. Casino player rolls
   (maybe with fair die, maybe with loaded die)
4. Highest number wins $2

An HMM for the occasionally dishonest casino



P(1|F) = 1/6
P(2|F) = 1/6
P(3|F) = 1/6
P(4|F) = 1/6
P(5|F) = 1/6
P(6|F) = 1/6

P(1|L) = 1/10
P(2|L) = 1/10
P(3|L) = 1/10
P(4|L) = 1/10
P(5|L) = 1/10
P(6|L) = 1/2

Question # 1 – Evaluation

GIVEN

A sequence of rolls by the casino player

124552646214614613613666166466163661636616361…

QUESTION

How likely is this sequence, given our model of how the casino works?

This is the EVALUATION problem in HMMs

Question # 2 – Decoding

GIVEN

A sequence of rolls by the casino player

1245526462146146136136661664661636616366163…

QUESTION

What portion of the sequence was generated with the fair die, and what portion with the loaded die?

This is the DECODING question in HMMs

Question # 3 – Learning

GIVEN

A sequence of rolls by the casino player

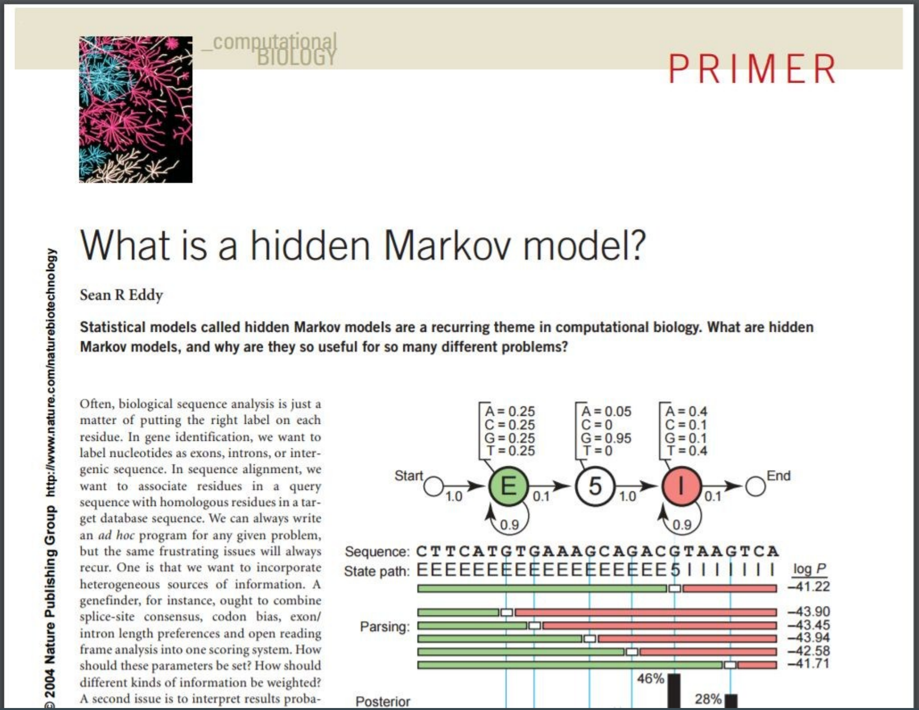
124552646214614613613666166466163661636616361651…

QUESTION

How “loaded” is the loaded die? How “fair” is the fair die? How often does the casino player change from fair to loaded, and back?

This is the LEARNING question in HMMs

HMM Article



_computational BIOLOGY

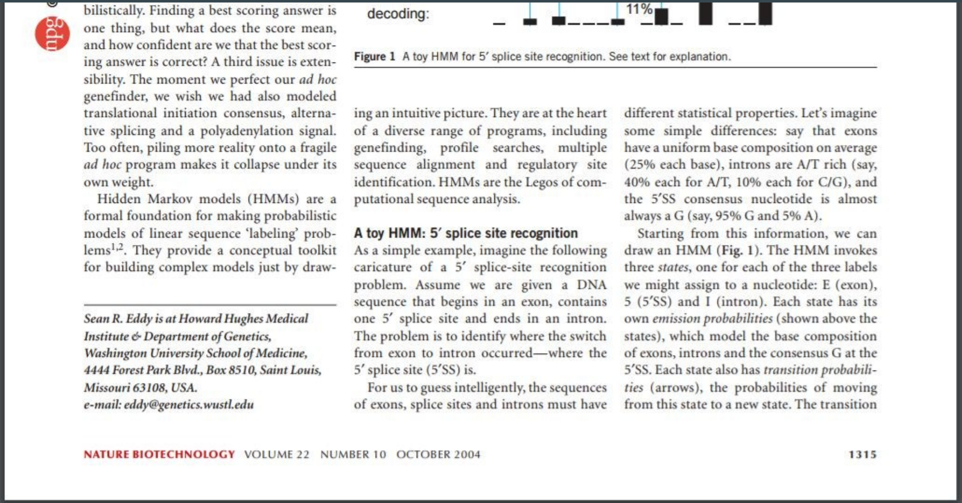
PRIMER

What is a hidden Markov model?

Sean R Eddy

Statistical models called hidden Markov models are a recurring theme in computational biology. What are hidden Markov models, and why are they so useful for so many different problems?

Often, biological sequence analysis is just a matter of putting the right label on each residue. In gene identification, we want to label nucleotides as exons, introns, or intergenic sequence. In sequence alignment, we want to associate residues in a query sequence with homologous residues in a target database sequence. We can always write an *ad hoc* program for any given problem, but the same frustrating issues will always recur. One is that we want to incorporate heterogeneous sources of information. A genefinder, for instance, ought to combine splice-site consensus, codon bias, exon/intron length preferences and open reading frame analysis into one scoring system. How should these parameters be set? How should different kinds of information be weighted? A second issue is to interpret results proba-



bilistically. Finding a best scoring answer is one thing, but what does the score mean, and how confident are we that the best scoring answer is correct? A third issue is extensibility. The moment we perfect our *ad hoc* genefinder, we wish we had also modeled translational initiation consensus, alternative splicing and a polyadenylation signal. Too often, piling more reality onto a fragile *ad hoc* program makes it collapse under its own weight.

Hidden Markov models (HMMs) are a formal foundation for making probabilistic models of linear sequence 'labeling' problems[1,2]. They provide a conceptual toolkit for building complex models just by drawing an intuitive picture. They are at the heart of a diverse range of programs, including genefinding, profile searches, multiple sequence alignment and regulatory site identification. HMMs are the Legos of computational sequence analysis.

Sean R. Eddy is at Howard Hughes Medical Institute & Department of Genetics, Washington University School of Medicine, 4444 Forest Park Blvd., Box 8510, Saint Louis, Missouri 63108, USA.
e-mail: eddy@genetics.wustl.edu

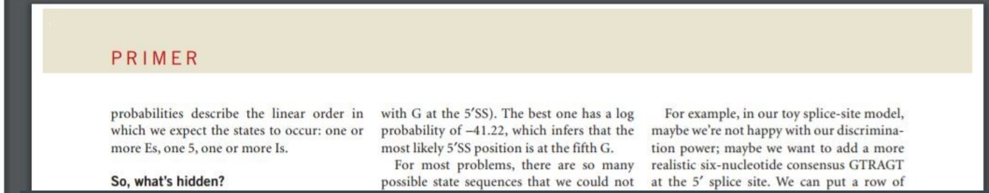
Figure 1 A toy HMM for 5′ splice site recognition. See text for explanation.

A toy HMM: 5′ splice site recognition

As a simple example, imagine the following caricature of a 5′ splice-site recognition problem. Assume we are given a DNA sequence that begins in an exon, contains one 5′ splice site and ends in an intron. The problem is to identify where the switch from exon to intron occurred—where the 5′ splice site (5′SS) is.

For us to guess intelligently, the sequences of exons, splice sites and introns must have different statistical properties. Let's imagine some simple differences: say that exons have a uniform base composition on average (25% each base), introns are A/T rich (say, 40% each for A/T, 10% each for C/G), and the 5′SS consensus nucleotide is almost always a G (say, 95% G and 5% A).

Starting from this information, we can draw an HMM (**Fig. 1**). The HMM invokes three *states*, one for each of the three labels we might assign to a nucleotide: E (exon), 5 (5′SS) and I (intron). Each state has its own *emission probabilities* (shown above the states), which model the base composition of exons, introns and the consensus G at the 5′SS. Each state also has *transition probabilities* (arrows), the probabilities of moving from this state to a new state. The transition



probabilities describe the linear order in which we expect the states to occur: one or more Es, one 5, one or more Is.

So, what's hidden?

with G at the 5′SS). The best one has a log probability of –41.22, which infers that the most likely 5′SS position is at the fifth G.

For most problems, there are so many possible state sequences that we could not

For example, in our toy splice-site model, maybe we're not happy with our discrimination power; maybe we want to add a more realistic six-nucleotide consensus GTRAGT at the 5′ splice site. We can put a row of

Thanks for your attention and questions