

BIOINFORMATICS

Kristel Van Steen, PhD²

Montefiore Institute - Systems and Modeling

GIGA - Bioinformatics

ULg

kristel.vansteen@ulg.ac.be

CHAPTER 3: SEQUENCE ANALYSIS

1 The biological problem

1.a Relevant questions

1.b Biological words ($k=1$)

2 Probability theory revisited

2.a Probability distributions

2.b Simulating from probability distributions

3 Biological words (k=2)

4 Markov Chains

5 Biological words (k=3)

6 Modeling the number of restriction sites in DNA

7 Sequence comparisons

7.a The biological problem

7.b Pairwise alignment

7.c Global alignment

1 The biological problem

1.a Questions

Relevant questions

Given the DNA sequence

```
AATCGGATGCGCGTAGGATCGGTAGGGTAGGCTTTAAGATCATGCTATTTTCGAGA
TTCGATTCTAGCTAGGTTTAGCTTAGCTTAGTGCCAGAAATCGGATGCGCGTAGGAT
CGGTAGGGTAGGCTTTAAGATCATGCTATTTTCGAGATTTCGATTCTAGCTAGGTTTT
TAGTGCCAGAAATCGTTAGTGCCAGAAATCGATT
```

..

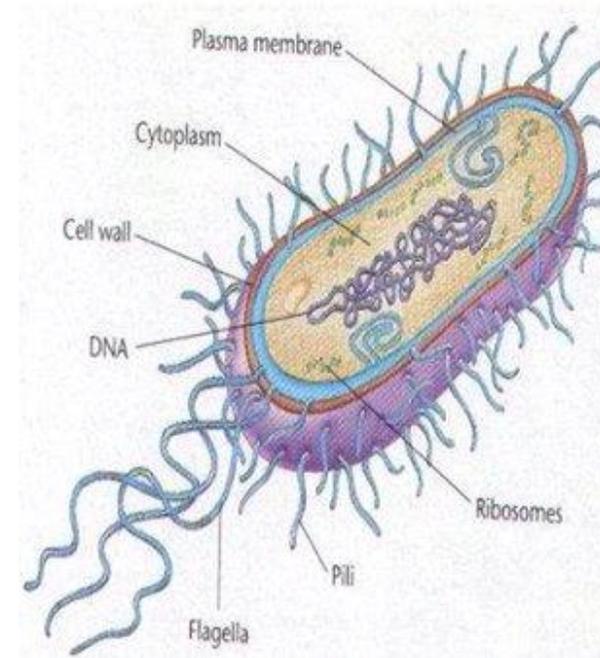
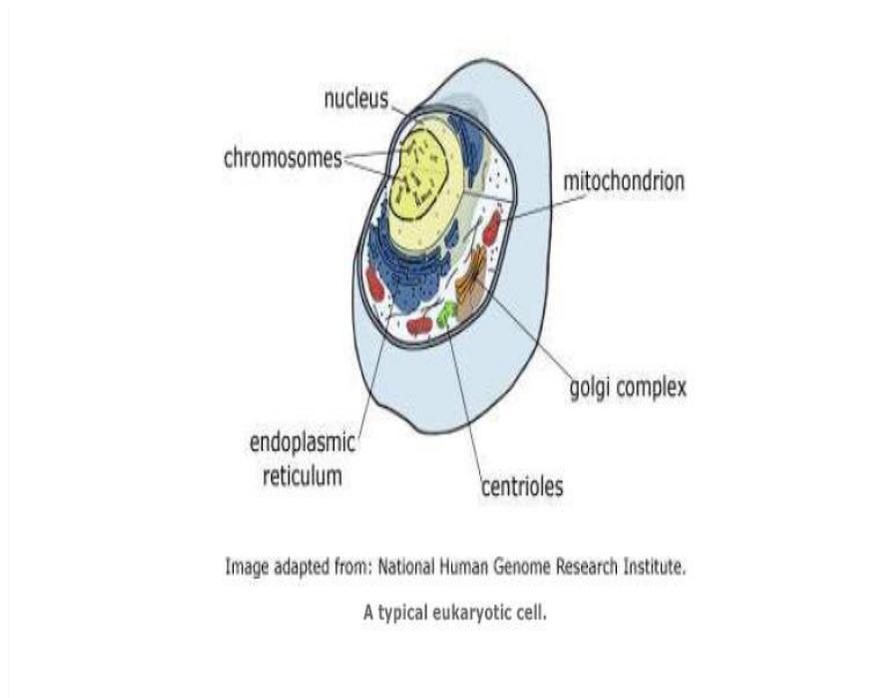
many questions arise

Relevant questions

- Is it likely to be a gene?
- What is the possible expression level?
- What is the possible protein product?
- Can we get the protein product?
- Can we figure out the key residue in the protein product?
- What sort of statistics to be used for describing this sequence?
- Can we determine the organism from which this sequence came?
- Do parameters describing the sequence differ from those describing bulk DNA in that organism?
- What sort of sequence might this be?
 - Protein coding?
 - Centromere / Telomere?

Recall: Different cell types

- *Eukaryotes*: organisms with a rather complex cellular structure.



- *Prokaryotes*: cells without organelles where the genetic information floats freely in the cytoplasm

Base composition for bacterial data:

JCVI CMR Comprehensive Microbial Resource

Search Locus for

Home Genome Tools Searches Comparative Tools Lists Downloads Carts

CMR Manual CMR FAQ CMR Tutorial Links Feedback Survey

Genome Search
Organism name:

Genome List
[View All CMR Genomes](#)

Gene Search
Search by: Locus Match: Exact Inexact
Keywords/Accession:

Data Summary

	Complete	Draft	Totals
Bacteria	509	17	526
Archaea	42	0	42
Viruses	3	0	3
Totals	554	17	571

Welcome to the Comprehensive Microbial Resource

The Comprehensive Microbial Resource (CMR) is a free website used to display information on all of the publicly available, complete prokaryotic genomes. In addition to the convenience of having all of the organisms on a single website, common data types across all genomes in the CMR make searches more meaningful, and cross genome analysis highlight differences and similarities between the genomes. A [CMR Mirror](#) site maintained by the Genome Encyclopedia of Microbes (GEM) in Korea is also available. [More Information](#) | [Publication Information](#)

CMR Menu Bar Tools

CMR offers a wide variety of tools and resources, all of which are available off of our menu bar at the top of each page. Below is an explanation and link for each of these menu options. First time users can use our [CMR tutorial](#) to learn how to navigate this site.

Genome Tools
Find organism lists as well as summary information and analyses for selected genomes.

Searches
Search CMR for genes, genomes, sequence regions, and evidence.

Comparative Tools
Compare multiple genomes based on a variety of criteria, including sequence homology and gene attributes. SNP data is also found under this menu.

Lists
Select and download gene, evidence, and genomic element lists.

Downloads
Download gene sequences or attributes for CMR organisms, or go to our FTP site.

Carts
Select genome preferences from our Genome Cart or download your Gene Cart genes.

Additional Resources

Announcements

Recent News
NEW! March 31, 2009 Data Release [23.0](#) is now available. 116 new genomes have been added to the CMR.

September 17, 2008: A new PANDA release is now available. PANDA is JCVI's Protein And Nucleotide Data Archive. It unifies the archival of the sequences from Taxonomy, GenBank, RefSeq, UniProt, PDB and PRF.

CMR Class Schedule
[June 2 - 4, 2009](#)
[September 29 - October 1, 2009](#)

Latest Releases
Data Release: [23.0](#)
Website Release: [3.0](#)

JCVI's Annotation Service
[Our Annotation Service](#) is a free service which provides automated annotation and tools for analysis of another center's prokaryotic sequence.

Contact Us
Can't find what you are looking for on the site? Want to alert us to new news or tools? Please [contact us](#).

(<http://cmr.jcvi.org/tigr-scripts/CMR/CmrHomePage.cgi>)

Focus on human data and Bioconductor / R Environment

BIOCONDUCTOR
open source software for bioinformatics

Bioconductor is an open source and open development software project for the analysis and comprehension of genomic data.

home getting started overview downloads documentation publications workshops cabig

project news

- ▶ [2009-01-07](#)
R, the open source platform used by Bioconductor, featured in a series of articles in the New York Times.
[More...](#)

QUICK LINKS

- ▶ [Getting Started](#)
- ▶ [Installation](#)
- ▶ [Downloads](#)
- ▶ [Software](#)
- ▶ [Workshops](#)

[BioC2009 Conference](#)
Seattle, WA, 27-28 July 2009. [Conference Material](#)

[Gene expression based on sequencing technologies](#)
Copenhagen, Denmark, 24-25 August 2009. [Details and Registration](#)

[Bioconductor 2.4 released – April 21, 2009](#)
Following the usual 6-month cycle, the Bioconductor community has released Bioconductor 2.4

Methods to answer the questions

-→ Probability theory / Statistics !!!

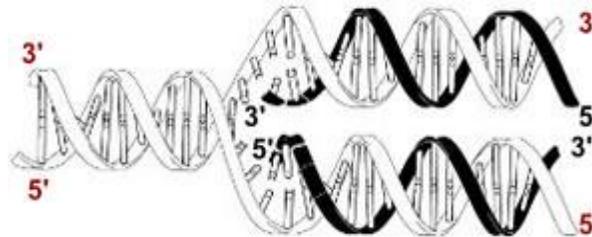
Words

- Words are short strings of letters drawn from an alphabet
- In the case of DNA, the set of letters is A, C, T, G
- A word of length k is called a k -word or k -tuple
- Differences in word frequencies help to differentiate between different DNA sequence sources or regions
- Examples:
 - 1-tuple: individual nucleotide
 - 2-tuple: dinucleotide
 - 3-tuple: codon

1.b Biological words of length 1 or base composition

General remarks

- There are constraints on base composition imposed by the genetic code
- The distribution of individual bases within a DNA molecule is not ordinarily uniform:
 - In prokaryotic genomes, there is an excess of G over C on the leading strands



- This can be described by the “GC skew”, characterized by:
 - $(\#G - \#C) / (\#G + \#C)$
 - # = nr of
- **This will have an influence on the probability model to impose...**

2 Probability theory revisited

2.a Probability distributions

Introduction

- Probability distributions are a fundamental concept in statistics. They are used both on a theoretical level and a practical level.
- Some practical uses of probability distributions are:
 - to calculate confidence intervals for parameters and
 - to calculate critical regions for hypothesis tests.

Introduction

- Before computing an interval or test based on a distributional assumption, we need to verify that the assumption is justified for the given data set.
- In this case, the distribution does not need to be the best-fitting distribution for the data, but an adequate enough model so that the statistical technique yields valid conclusions.
- Simulation studies: one way to obtain empirical evidence for a probability model

Assumptions

- Simple rules specifying a probability model:
 - First base in sequence is either A, C, T or G with prob p_A, p_C, p_T, p_G
 - Suppose the first r bases have been generated, while generating the base at position $r+1$, no attention is paid to what has been generated before. → “independence”
- A, C, T or G is generated with the probabilities above
- Notation for the output of a random string of n bases: L_1, L_2, \dots, L_n (L_i = base inserted at position i of the sequence)

Probability distributions

- In the DNA sequence, **any position** can take either one of $J=4$ values:

$\chi = \{A, C, T, G\}$ = outcome set at a position

- X is a discrete random variable (since its values are uncertain)

- If p_j is the prob that the value x_j occurs, then

- $p_1, \dots, p_J \geq 0$ and $p_1 + \dots + p_J = 1$

- The probability distribution (probability mass function) of X is given by the collection p_1, \dots, p_J

- $P(X=x_j) = p_j, j=1, \dots, J$

- The probability that an event S occurs (subset of χ) is $P(X \in S) =$

$$\sum_{j: x_j \in S} (p_j)$$

Probability distributions

- What is the probability distribution of the number of times a given pattern occurs in a **random DNA sequence** L_1, \dots, L_n ?

- New sequence X_1, \dots, X_n :

$$X_i=1 \text{ if } L_i=A \text{ and } X_i=0 \text{ else}$$

- The number of times N that A appears is the sum

$$N=X_1+\dots+X_n$$

- The prob distr of each of the X_i :

$$P(X_i=1) = P(L_i=A)=p_A$$

$$P(X_i=0) = P(L_i=C \text{ or } G \text{ or } T) = 1 - p_A$$

Independence

- What is a “typical” value of N ?
 - Depends on how the individual X_i (for different i) are interrelated → a simple assumption is the one of independence
- Discrete random variables X_1, \dots, X_n are said to be independent if for any subset of random variables and actual values, the joint distribution equals the product of the component distributions
- According to our simple model, the L_i are independent and hence

$$P(L_1=l_1, L_2=l_2, \dots, L_n=l_n) = P(L_1=l_1) P(L_2=l_2) \dots P(L_n=l_n)$$

Expected values and variances

- Mean and variance are two important properties of real-valued random variables and corresponding probability distributions.
- The “mean” of a discrete random variable X taking values x_1, x_2, \dots (denoted EX (or $E(X)$ or $E[X]$), where E stands for expectation, which is another term for mean) is defined as:

$$EX = \sum_i x_i P(X = x_i).$$

- $EX_i = 1 \times p_A + 0 \times (1 - p_A)$
 - If $Y=cX$, then $EY = cEX$
 - $E(X_1 + \dots + X_n) = EX_1 + \dots + EX_n$
- Because X_i are assumed to be independent and identically distributed (iid):

$$EN = nEX_1 = np_A$$

Expected values and variances

- The idea is to use squared deviations of X from its center (expressed by the mean). Expanding the square and using the linearity properties of the mean, the $\text{Var}(X)$ can also be written as:

$$\text{Var}(X) = E[X^2] - [EX]^2$$

- If $Y=cX$ then $\text{Var}Y = c^2\text{Var}X$
 - The variance of a sum of independent random variables is the sum of the individual variances
-
- For the random variables X_i :
 $\text{Var}X_i = [1^2 \times p_A + 0^2 \times (1 - p_A)] - p_A^2 = p_A(1 - p_A)$
 $\text{Var}N = n\text{Var}X_1 = np_A(1 - p_A)$

Expected values and variances: a note aside

- The expected value of a random variable X gives a measure of its location. Variance is another property of a probability distribution dealing with the spread or variability of a random variable around its mean.

$$\text{Var}(X) = E [X - EX]^2$$

- The positive square root of the variance of X is called its standard deviation $\text{sd}(X)$

Do you know the difference between standard error and standard deviation?

The binomial distribution

- The binomial distribution is used when there are exactly two mutually exclusive outcomes of a trial. These outcomes are appropriately labeled "success" and "failure". The binomial distribution is used to obtain the probability of observing x successes in a fixed number of trials, with the probability of success on a single trial denoted by p . The binomial distribution assumes that p is fixed for all trials.
- The formula for the binomial probability mass function is :

$$P(N = j) = \binom{n}{j} p^j (1 - p)^{n-j}, j = 0, 1, \dots, n$$

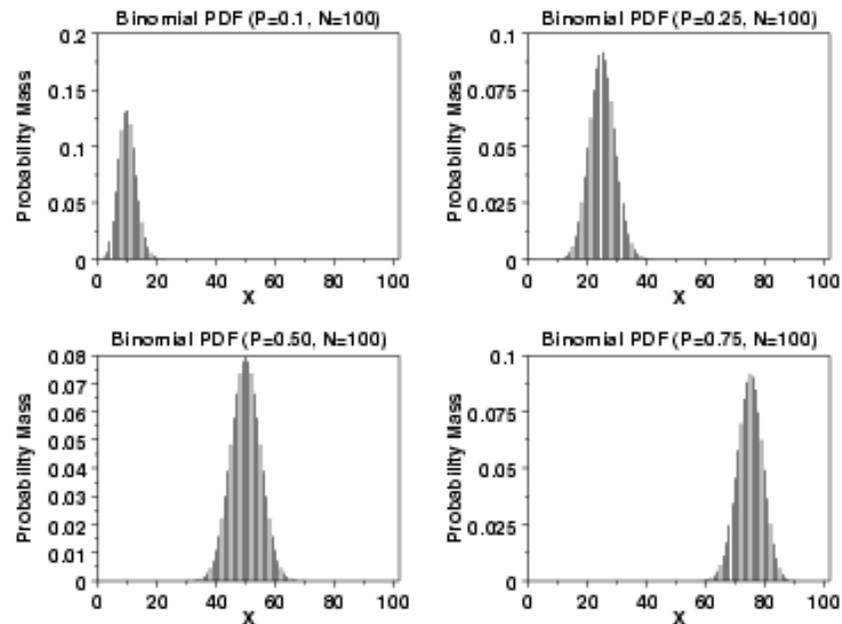
with the binomial coefficient $\binom{n}{j}$ determined by

$$\binom{n}{j} = \frac{n!}{j! (n - j)!}$$

and $j! = j(j-1)(j-2)\dots 3.2.1$, $0! = 1$

The binomial distribution

- The mean is np and the variance is $np(1-p)$ → **cfr with behavior of N!**
- The following is the plot of the binomial probability density function for four values of p and $n = 100$.



2.b Simulating from probability distributions

- The idea here is that we can study the properties of the distribution of N when we can get our computer to output numbers N_1, \dots, N_n having the same distribution as N
 - We can use the sample mean to estimate the expected value EN :

$$\bar{N} = (N_1 + \dots + N_n)/n$$

- Similarly, we can use the sample variance to estimate the true variance of N :

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (N_i - \bar{N})^2$$

Why do we use $(n-1)$ and not n in the denominator?

Simulating from probability distributions

- What is needed to **produce such a string** of numbers N_1, \dots, N_n having the same distribution as N ?
 - Select a number (realization) u from a uniform distribution on $(0,1)$: any number between 0 and 1 is a possible outcome and each is equally likely
 - Set $X_1=1$ if $u \leq p \equiv p_A$ and 0 otherwise.
 - Why does this work? ... $P(X_1 = 1) = P(U \leq p_A) = p_A$
 - Repeating this procedure n times results in a sequence X_1, \dots, X_n from which N can be computed by adding the X 's

Simulating from probability distributions

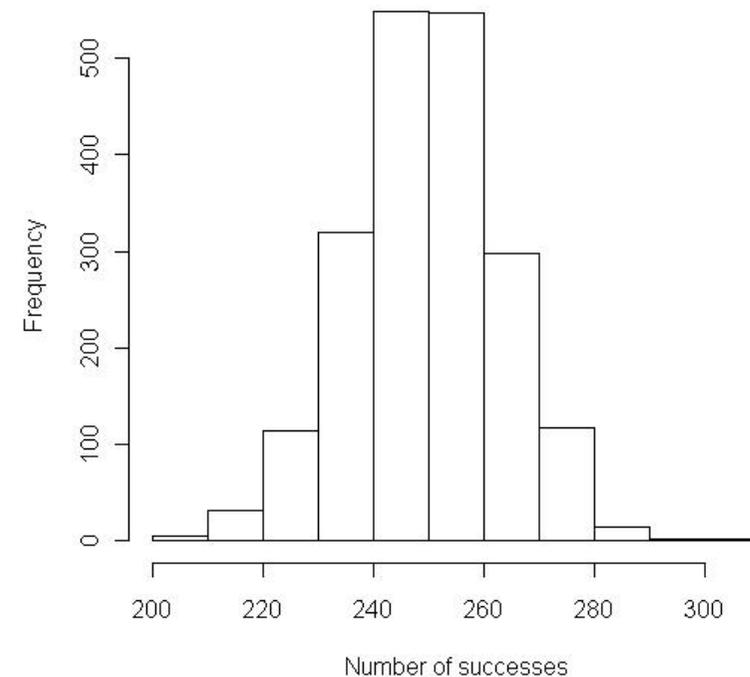
- Simulate a sequence of bases L_1, \dots, L_n :
 - Divide the interval $(0,1)$ in 4 intervals with endpoints
 $p_A, p_A + p_C, p_A + p_C + p_G, 1$
 - If the simulated u lies in the leftmost interval, $L_1=A$
 - If u lies in the second interval, $L_1=C$; if in the third, $L_1=G$ and otherwise $L_1=T$
 - Repeating this procedure n times with different values for U results in a sequence L_1, \dots, L_n

- Use the “sample” function in R:

```
pi <- c(0.25,0.75)
x<-c(1,0)
set.seed(2009)
sample(x,10,replace=TRUE,pi)
```

Simulating from probability distributions

- By looking through a given simulated sequence, we can count the number of times a particular pattern arises (for instance, the base A)
- By repeatedly generating sequences and analyzing each of them, we can get a feel for whether or not our particular pattern of interest is unusual



Simulating from probability distributions

- Using R code:

```
x<- rbinom(2000,1000,0.25)
mean(x)
sd(x)^2
hist(x,xlab="Number of successes",main="")
```

What is the number of observations?

- Suppose we have a sequence of 1000bp and assume that every base occurs with equal probability. How likely are we to observe at least 300 A's in such a sequence?
 - Exact computation using a closed form of the relevant distribution
 - Approximate via simulation
 - Approximate using the Central Limit Theory

Exact computation via closed form of relevant distribution

- The formula for the binomial probability mass function is :

$$P(N = j) = \binom{n}{j} p^j (1 - p)^{n-j}, j = 0, 1, \dots, n$$

and therefore

$$\begin{aligned} P(N \geq 300) &= \sum_{j=300}^{1000} \binom{1000}{j} (1/4)^j (1 - 1/4)^{1000-j} \\ &= 0.00019359032194965841 \end{aligned}$$

		P: exactly 300 out of 1000	
Method 1. exact binomial calculation		0.00004566114740576488	
Method 2. approximation via normal		0.000038	
Method 3. approximation via Poisson	-----		
		P: 300 or fewer out of 1000	
Method 1. exact binomial calculation		0.9998520708293378	
Method 2. approximation via normal		0.999885	
Method 3. approximation via Poisson	-----		
		P: 300 or more out of 1000	
Method 1. exact binomial calculation		0.00019359032194965841	
Method 2. approximation via normal		0.000153	
Method 3. approximation via Poisson	-----		
For hypothesis testing		P: 300 or more out of 1000	
		One-Tail	Two-Tail
Method 1. exact binomial calculation		0.00019359032194965841	0.0003025705168772097
Method 2. approximation via normal		0.000153	0.000306
Method 3. approximation via Poisson	-----		-----

(<http://faculty.vassar.edu/lowry/binomialX.html>)

Approximate via simulation

- Using R code and simulations from the theoretical distribution, $P(N \geq 300)$ can be estimated as 0.000196 via.

```
x<- rbinom(1000000,1000,0.25)
sum(x>=300)/1000000
```

or 0.0001479292 via

```
1-pbinom(300,size=1000,prob=0.25)
pbinom(300,size=1000,prob=0.25,lower.tail=FALSE)
```

Approximate via Central Limit Theory

- The central limit theorem offers a 3rd way to compute probabilities of a distribution
- It applies to sums or averages of iid random variables
- Assuming that X_1, \dots, X_n are iid random variables with mean μ and variance σ^2 , then we know that for the sample average

$$\bar{X}_n = \frac{1}{n} (X_1 + \dots + X_n),$$

$$E\bar{X}_n = \mu \text{ and } \text{Var } \bar{X}_n = \frac{\sigma^2}{n}$$

- Hence,

$$E\left(\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}}\right) = 0, \text{Var}\left(\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}}\right) = 1$$

Approximate via Central Limit Theory

- The central limit theorem states that if the sample size n is large enough,

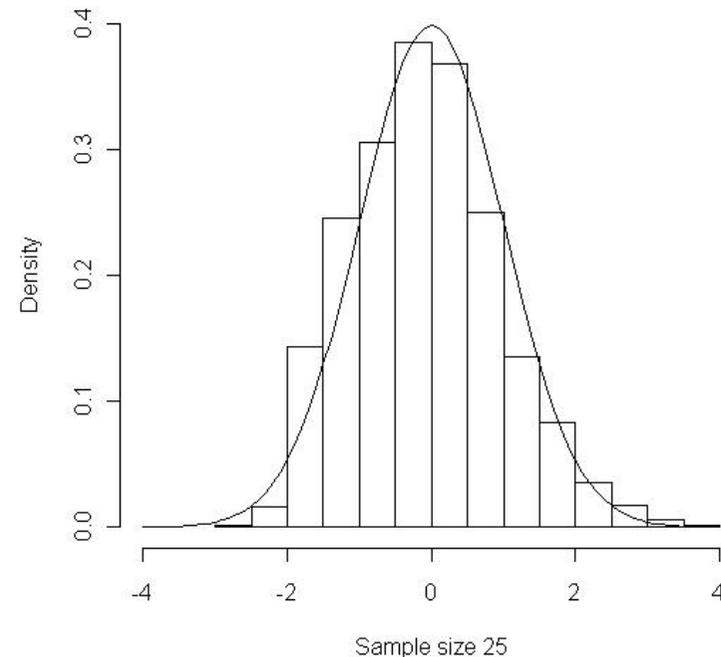
$$P\left(a \leq \frac{\bar{X}_n - \mu}{\frac{\sigma}{\sqrt{n}}} \leq b\right) \approx \phi(b) - \phi(a),$$

with $\phi(\cdot)$ the standard normal distribution defined as

$$\phi(z) = P(Z \leq z) = \int_{-\infty}^z \phi(x) dx$$

The central limit theorem in action using R code

```
bin25<-rbinom(1000,25,0.25)
av.bin25 <- 25*0.25
stdev.bin25 <- sqrt(25*0.25*0.75)
bin25<-(bin25-av.bin25)/stdev.bin25
hist(bin25,xlim=c(-4,4),ylim=c(0.0,0.4),prob=T
RUE,xlab="Sample size 25",main="")
x<-seq(-4,4,0.1)
lines(x,dnorm(x))
```



Approximate via Central Limit Theory

- Estimating the quantity $P(N \geq 300)$ when N has a binomial distribution with parameters $n=1000$ and $p=0.25$,

$$E(N) = n\mu = 1000 \times 0.25 = 250,$$

$$sd(N) = \sqrt{n} \sigma = \sqrt{1000 \times \frac{1}{4} \times \frac{3}{4}} \approx 13.693$$

$$P(N \geq 300) = P\left(\frac{N - 250}{13.693} > \frac{300 - 250}{13.693}\right)$$

$$\approx P(Z > 3.651501) = 0.0001303560$$

- R code:

```
pnorm(3.651501,lower.tail=FALSE)
```

How do the estimates of $P(N \geq 300)$ compare?

3 Biological words of length 2

Introduction

- Dinucleotides are important because physical parameters associated with them can describe the trajectory of the DNA helix through space (such as DNA bending). This may affect gene expression.
- Concentrating on abundances, and assuming the iid model for L_1, \dots, L_n :

$$P(L_i = l_i, L_{i+1} = l_{i+1}) = p_{l_i} p_{l_{i+1}}$$

- Has a given sequence an unusual dinucleotide frequency compared to the iid model?
 - Compare observed O with expected E dinucleotide numbers

$$\chi^2 = \frac{(O-E)^2}{E},$$

with $E = (n - 1)p_{l_i}p_{l_{i+1}}$.

Why $(n-1)$ as factor? **How many df? 1?**

Comparing to the reference

- How to determine which values of χ^2 are unlikely or extreme?

- Recipe:

- Compute the number c given by

$$c = \begin{cases} 1 + 2p_{l_i} - 3p_{l_i}^2, & \text{if } l_i = l_{i+1} \\ 1 - 3p_{l_i}p_{l_{i+1}}, & \text{if } l_i \neq l_{i+1} \end{cases}$$

- Calculate the ratio $\frac{\chi^2}{c}$, where χ^2 is given as before
- If this ratio is larger than 3.84 then conclude that the iid model is not a good fit
- Note: $qchisq(0.95,1) = 3.84$

4 Markov Chains

Introduction

- When moving from bacteria (such as **E. coli**, a common type of bacteria that can get into food, like beef and vegetables) to real genomes, a more complicated probabilistic model is required than the iid model before to capture the dinucleotide properties
- One approach is to use Markov chains.
- Markov chains are a direct generalization of independent trials, where the character at a position may depend on the characters of preceding positions, hence may be conditioned on preceding positions

Conditional probabilities

- If Ω refers to the set of all possible outcomes of a single experiment, A to a particular event, and A^c to the complement $\Omega - A$ of A , then

$$P(A) + P(A^c) = 1,$$

- The conditional probability of A given B is $P(A|B) = \frac{P(A \cap B)}{P(B)}$, $P(B) > 0$
- As a consequence: $P(B|A) = \frac{P(A|B) P(B)}{P(A)}$, also known as Bayes' Theorem
- For B_1, \dots, B_k forming a partition of Ω , this is the B_i are disjoint and the B_i are exhaustive (their union is Ω), the **law of total probability** holds:

$$P(A) = \sum_{i=1}^k P(A \cap B_i) = \sum_{i=1}^k P(A|B_i) P(B_i)$$

The Markov property

- The property will be explained via studying a sequence of random variables X_t , $t=0,1,\dots$ taking values in the state space $\{A,C,T,G\}$
- The sequence $\{X_t, t \geq 0\}$ is called a first-order Markov chain if only the previous neighbor influences the probability distribution of the character at any position, and hence satisfies the Markov property:

$$P(X_{t+1} = j | X_t = i, X_{t-1} = i_{t-1}, \dots, X_0 = i_0) = P(X_{t+1} = j | X_t = i)$$

for $t \geq 0$ and for all $i, j, i_{t-1}, \dots, i_0$ in the state space

- We consider Markov chains that are homogeneous:

$$P(X_{t+1} = j | X_t = i) = p_{ij} \text{ (i.e. independent of the position } t)$$

The Markov property

- The p_{ij} are the elements of a matrix P called the one-step transition matrix of the chain.
- Stepping from one position to the next is one issue, how to start is another issue
 - An initial probability distribution is needed as well
 - It is determined by a vector of probabilities corresponding to every possible initial state value i : $\pi_i^{(0)} = \pi_i = P(X_0 = i)$
- The probability distribution for the states at position 1 can be obtained as follows:

$$\begin{aligned} P(X_1 = j) &= \sum_{i \in \mathcal{X}} P(X_0 = i, X_1 = j) \\ &= \sum_{i \in \mathcal{X}} P(X_0 = i) P(X_1 = j | X_0 = i) = \sum_{i \in \mathcal{X}} \pi_i p_{ij} \end{aligned}$$

The Markov property

- To compute the probability distribution for the states at position 2, we first show that $P(X_2 = j | X_0 = i)$ is the ij -th element of $PP = P^2$

$$\begin{aligned} P(X_2 = j | X_0 = i) &= \sum_{k \in \mathcal{X}} P(X_2 = j, X_1 = k | X_0 = i) \\ &= \sum_{k \in \mathcal{X}} P(X_2 = j | X_1 = k, X_0 = i) P(X_1 = k | X_0 = i) \\ &= \sum_{k \in \mathcal{X}} P(X_2 = j | X_1 = k) P(X_1 = k | X_0 = i) \\ &= \sum_{k \in \mathcal{X}} p_{ik} p_{kj} = (PP)_{ij} \end{aligned}$$

- Therefore

$$\pi_j^{(2)} = P(X_2 = j) = \sum_{i \in \mathcal{X}} \pi_i P_{ij}^2$$

The Markov property

- In a similar way it can be shown that

$$\pi_j^{(t)} = P(X_t = j) = \sum_{i \in \mathcal{X}} \pi_i P_{ij}^t$$

- In principle, it can happen that the distribution $\pi^{(t)}$ is independent of t. This event is then referred to as a stationary distribution of the chain.
 - It occurs when $\sum_{i \in \mathcal{X}} \pi_i p_{ij} = \pi_j$, for all j, or stated differently when
$$\pi = \pi P$$

Creating our own Markov chain simulation in practice (for those interested)

- Assume the observed dinucleotide relative frequencies (each row specifies a base and each column specifies the following base):

	A	C	G	T
A	0.146	0.052	0.058	0.089
C	0.063	0.029	0.010	0.056
G	0.050	0.030	0.028	0.051
T	0.087	0.047	0.063	0.140

- Questions:
 - **How to compute the individual base frequencies?** (law of total prob.)
How to propose initial state parameters to build a Markov chain?
 - **How to compute the transition matrix?**

```
markov1 <- function(x,pi,P,n){
  mg <- rep(0,n)
  mg[1] <- sample(x,1,replace=TRUE,pi)
  for (k in 1:(n-1)){
    mg[k+1] <- sample(x,1,replace=TRUE,P[mg[k],])
  }
  return(mg)
}
```

```
x<-c(1:4)
pi <- c(0.342,0.158, 0.158, 0.342)
# A C G T one-step transition matrix:
P <- matrix(scan(),ncol=4,nrow=4,byrow=T)
0.423 0.151 0.168 0.258
0.399 0.184 0.063 0.354
0.314 0.189 0.176 0.321
0.258 0.138 0.187 0.415
```

```
tmp <- markov1(x,pi,P,50000)
A<- length(tmp[tmp]==1)
C<- length(tmp[tmp]==2)
G<- length(tmp[tmp]==3)
T<- length(tmp[tmp]==4)
(C+G)/(A+C+G+T) # fraction of G+C

count <-0
for (i in 1:49999){
  if (tmp[i]==2 && tmp[i+1]==3)
    count <- count+1
}
count/49999 # abundance of CG dinucleotide as estimated by the model
```

5 Biological words of length 3

Introduction

- There are 61 codons that specify amino acids and three stop codons.
- Since there are 20 common amino acids, this means that most amino acids are specified by more than one codon.
- This has led to the use of a number of statistics to summarize the "bias" in codon usage.
- Since there is variation in codon frequencies, it is interesting to investigate these frequencies in more detail

Predicted relative frequencies

- For a sequence of independent bases L_1, L_2, \dots, L_n the expected 3-tuple relative frequencies can be found by using the logic employed for dinucleotides we derived before
- The probability of a 3-word can be calculated as follows:

$$\mathbb{P}(L_i = r_1, L_{i+1} = r_2, L_{i+2} = r_3) = \mathbb{P}(L_i = r_1)\mathbb{P}(L_{i+1} = r_2)\mathbb{P}(L_{i+2} = r_3).$$

- This provides the expected frequencies of particular codons, using the individual base frequencies

Predicted relative frequencies

			Observed	
			Gene Class I	Gene Class II
Codon Predicted			(502)	(191)
Phe	TTT	0.493	0.551	0.291
	TTC	0.507	0.449	0.709
Ala	GCT	0.246	0.145	0.275
	GCC	0.254	0.276	0.164
	GCA	0.246	0.196	0.240
	GCG	0.254	0.382	0.323
Asn	AAT	0.493	0.409	0.172
	AAC	0.507	0.591	0.828

- Comparison of predicted and observed triplet frequencies in coding sequences for a subset of genes and codons from *E. coli*.
- Figures in parentheses below each gene class show the number of genes in that class. Table 2.3 from Deonier et al 2005.

Predicted relative frequencies

- Medigue et al. (1991) observed three gene classes.
- For Phe and Asn different usage patterns are observed for Gene Class I and Gene Class II.
- For Gene Class II in particular, the observed codon frequencies differ considerably from their predicted frequencies

			Observed	
Codon Predicted			Gene Class I (502)	Gene Class II (191)
Phe	TTT	0.493	0.551	0.291
	TTC	0.507	0.449	0.709
Ala	GCT	0.246	0.145	0.275
	GCC	0.254	0.276	0.164
	GCA	0.246	0.196	0.240
	GCG	0.254	0.382	0.323
Asn	AAT	0.493	0.409	0.172
	AAC	0.507	0.591	0.828

Moderate expr

High expression levels

The codon adaptation index

Problem setting and notation:

- Consider a sequence of amino acids $X = x_1, x_2, \dots, x_L$ representing protein X , with x_k representing the amino acid residue corresponding to codon k in the gene.
- We are interested in comparing the actual codon usage with an alternative model: that the codons employed are the most probable codons for highly expressed genes.
- For the codon corresponding to a particular amino acid at position k in protein X , let p_k be the probability that *this* particular codon is used to code for the amino acid in highly expressed genes
- Let q_k correspond to the probability for *the most frequently used* codon of the corresponding amino acid in highly expressed genes.

The codon adaptation index (Sharp and Li 1987)

- The CAI is defined as

$$\text{CAI} = \left[\prod_{k=1}^L p_k/q_k \right]^{1/L}$$

- It is the geometric mean of the ratios of the probabilities for the codons *actually* used to the probabilities of the codons *most frequently* used in highly expressed genes.
- An alternative way of writing this is

$$\log(\text{CAI}) = \frac{1}{L} \sum_{k=1}^L \log(p_k/q_k).$$

- This expression is in terms of a sum of the logarithms of probability ratios, a form that is encountered repeatedly in other contexts as well.

The codon adaptation index

- The CAI can be shown to be correlated with mRNA levels
- Hence, the CAI for a gene sequence in genomic DNA provides a first approximation of its expression level:
 - if the CAI is relatively large, then we would predict that the expression level is also large.
- As an example, consider the amino acid sequence from the amino terminal end of the *himA* gene of *E. coli* (which codes for one of the two subunits of the protein IHF: length $L = 99$).

```
M   A   L   T   K   A   E   M   S   E   Y   L   F   ...  
ATG GCG CTT ACA AAA GCT GAA ATG TCA GAA TAT CTG TTT ...
```

The codon adaptation index

- Top lines: amino acid sequence and corresponding codons.
- Upper table: probabilities for codons in lower table.
- The probability of the most frequently used codon in highly expressed genes is underlined (Fig 2.2 – Deonier et al 2005).

M A L T K A E M S E Y L F ...
 ATG GCG CTT ACA AAA GCT GAA ATG TCA GAA TAT CTG TTT ...

<u>1.000</u>	<u>0.469</u>	0.018	0.451	<u>0.798</u>	<u>0.469</u>	<u>0.794</u>	<u>1.000</u>	<u>0.428</u>	<u>0.794</u>	0.193	0.018	0.228
	0.057	0.018	<u>0.468</u>	0.202	0.057	0.206		0.319	0.206	<u>0.807</u>	0.018	<u>0.772</u>
	0.275	0.038	0.035		0.275			0.033			0.038	
	0.199	0.033	0.046		0.199			0.007			0.033	
		0.007						0.037			0.007	
		<u>0.888</u>						0.176			<u>0.888</u>	
ATG	GCT	TTA	ACT	AAA	GCT	GAA	ATG	TCT	GAA	TAT	TTA	TTT
	GCC	TTG	ACC	AAG	GCC	GAG		TCC	GAG	TAC	TTG	TTC
	GCA	CTT	ACA		GCA			TCA			CTT	
	GCG	CTC	ACG		GCG			TCG			CTC	
		CTA						AGT			CTA	
		CTG						AGC			CTG	

The codon adaptation index

- The CAI for this fragment of coding sequence is given by

$$\text{CAI} = \left[\frac{1.000}{1.000} \times \frac{0.199}{0.469} \times \frac{0.038}{0.888} \times \frac{0.035}{0.468} \dots \right]^{1/99} .$$

- If every codon in a gene corresponded to the most frequently used codon in highly expressed genes, then the CAI would be 1.0.
- In *E. coli* a sample of 500 protein-coding genes displayed CAI values in the range from 0.2 to 0.85 (Whittam, 1996)

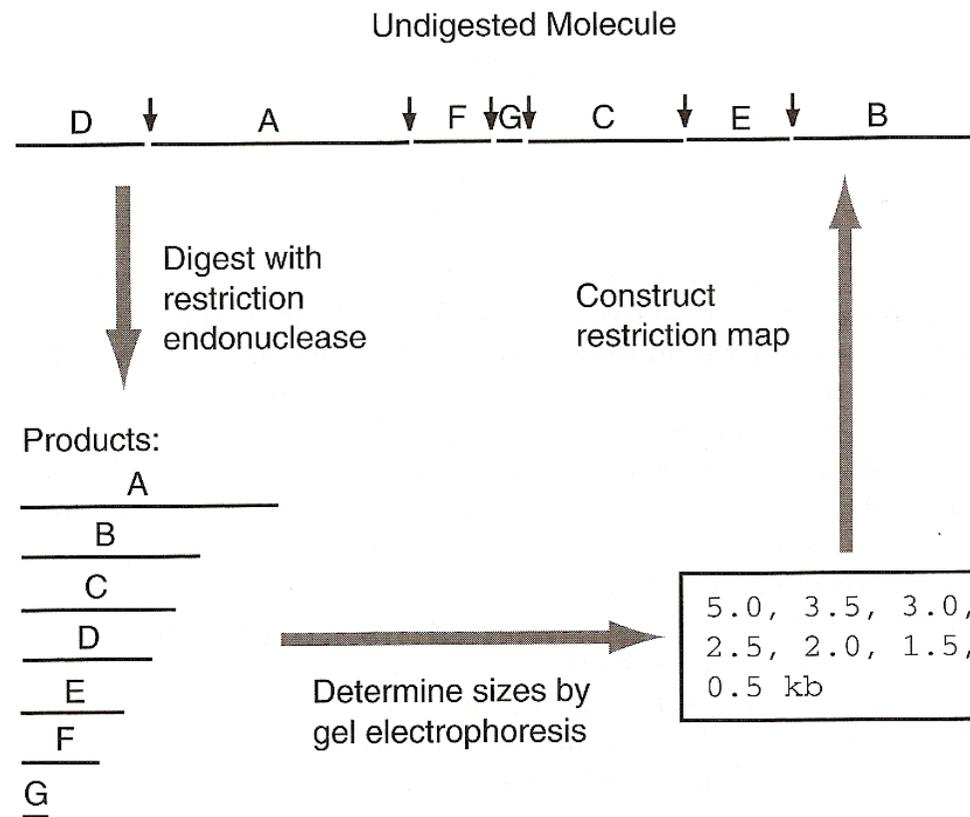
6 Modeling the number of restriction sites in DNA

The biological problem

- A **restriction map** is a display of positions on a DNA molecule where cleavage by one or more restriction endonucleases can occur.
- It is created by determining the ordering of the DNA fragments generated after digestion with one or more restriction endonucleases.
- The restriction map is useful not only for dissecting a DNA segment for further analysis but also as a "fingerprint" or bar code that distinguishes that molecule from any other molecule.
- A graphical summary is given in the following figure (Figure 3.1 – Deonier et al 2005)

The biological problem

- The order of fragments (D, A, F, G, C, E, B) is originally unknown. A variety of techniques may be employed to determine this order.



The biological problem

- Modelling the number of restriction sites in DNA is important when addressing the following question:

If we were to digest the DNA with a restriction endonuclease such as EcoR1, approximately how many fragments would be obtained, and what would be their size distribution?

The number of restriction sites

- Restriction endonuclease recognition sequences have length t (4, 5, 6 or 8 typically), where t is much smaller than n .
- Our model assumes that cleavage can occur between any two successive positions on the DNA.
 - This is wrong in detail because, depending upon where cleavage occurs within the bases of the recognition sequence (which may differ from enzyme to enzyme), there are positions near the ends of the DNA that are excluded from cleavage.
 - However, since t is much smaller than n , the ends of the molecule do not affect the result too much

The number of restriction sites

- We again use X_i to represent the outcome of a trial occurring at position i , but this time X_i does not represent the identity of a base (one of four possible outcomes) but rather whether position i is or is not the beginning of a restriction site.
- In particular,

$$X_i = \begin{cases} 1, & \text{if base } i \text{ is the start of a restriction site,} \\ 0, & \text{if not.} \end{cases}$$

- We denote by p the probability that any position i is the beginning of a restriction site:

$$X_i = \begin{cases} 1, & \text{with probability } p, \\ 0, & \text{with probability } 1 - p. \end{cases}$$

The number of restriction sites

- Unlike with tossing a fair coin, for the case of restriction sites on DNA, p depends upon
 - the base composition of the DNA and
 - the identity of the restriction endonuclease.
- For example:
 - Suppose that the restriction endonuclease is *EcoRI*, with recognition sequence 5'-GAATTC-3'.
 - The site really recognized is duplex DNA, with the sequence of the other strand determined by the Watson-Crick base-pairing rules.
 - Suppose furthermore that the DNA has equal proportions of A, C, G, and T: 0.25.

The number of restriction sites

- The probability that any position is the beginning of a site is the probability that this first position is G, the next one is A, the next one is A, the next one is T, the next one is T, and the last one is C.
- Since, by the iid model, the identity of a letter at any position is independent of the identity of letters at any other position, we see from the multiplication rule that

$$p = \mathbb{P}(\text{GAATTC}) = \mathbb{P}(\text{G})\mathbb{P}(\text{A})\mathbb{P}(\text{A})\mathbb{P}(\text{T})\mathbb{P}(\text{T})\mathbb{P}(\text{C}) = (0.25)^6 \sim 0.00024.$$

- Notice that p is small, a fact that becomes important later.

The number of restriction sites

- The appearance of restriction sites along the molecule is represented by the string X_1, X_2, \dots, X_n ,
- The number of restriction sites is $N = X_1 + X_2 + \dots + X_m$, where $m = n - 5$.
 - The sum has m terms in it because a restriction site of length 6 cannot begin in the last five positions of the sequence, as there aren't enough bases to fit it in.
- For simplicity of exposition we take $m = n$ in what follows.
- What really interests us is the number of "successes" (restriction sites) in n trials.

The number of restriction sites

- If X_1, X_2, \dots, X_n were independent of one another, then the probability distribution of N would be a binomial distribution with parameters n and p ;
 - The expected number of sites would therefore be np
 - The variance would be $np(1 - p)$.
- We remark that despite the X_i are not in fact independent of one another (because of overlaps in the patterns corresponding to X_i and X_{i+1} , for example), the binomial approximation usually works well.
- Computing probabilities of events can be cumbersome when using the probability distribution

$$P(N = j) = \binom{n}{j} p^j (1 - p)^{n-j}, j = 0, 1, \dots, n$$

Poisson approximation to the binomial distribution

- In what follows, we assume that n is large and p is small, and we set $\lambda = np$.
- We know that for $j = 0, 1, \dots, n$,

$$P(N = j) = \binom{n}{j} p^j (1 - p)^{n-j}$$

- Writing

$$\mathbb{P}(N = j) = \frac{n(n-1)(n-2)\cdots(n-j+1)}{j!(1-p)^j} p^j (1-p)^n.$$

and given that the number of restriction sites (j) is small compared to the length of the molecule (n), such that

$$n(n-1)(n-2)\cdots(n-j+1) \approx n^j, (1-p)^j \approx 1,$$

Poisson approximation to the binomial distribution

$$\mathbb{P}(N = j) \approx \frac{(np)^j}{j!} (1 - p)^n = \frac{\lambda^j}{j!} \left(1 - \frac{\lambda}{n}\right)^n.$$

in which $\lambda = np$.

- From calculus, for any x ,

$$\lim_{n \rightarrow \infty} \left(1 - \frac{x}{n}\right)^n = e^{-x}.$$

- Since n is large (often more than 10^4), we replace $\left(1 - \frac{\lambda}{n}\right)^n$ by $e^{-\lambda}$ to get our final approximation in the form

$$\mathbb{P}(N = j) \approx \frac{\lambda^j}{j!} e^{-\lambda}, \quad j = 0, 1, 2, \dots$$

- This is the formula for the Poisson distribution with parameter $\lambda = np = \text{Var}(N) = E(N)$

Poisson approximation to the binomial distribution

- Example:

- To show how this approximation can be used, we estimate the probability that there are no more than two *EcoRI* sites in a DNA molecule of length 10,000, assuming equal base frequencies
- Earlier we obtained $p=0.00024$ for this setting.
- The problem is to compute $P(N \leq 2)$
 - Therefore $\lambda = np = 2.4$
 - Using the Poisson distribution: $P(N \leq 2) \approx 0.570$
 - Interpretation: More than half the time, molecules of length 10,000 and uniform base frequencies will be cut by *EcoRI* two times or less

- R code:

```
ppois(2,2.4)
```

Distribution of restriction fragment lengths

- With this generalization, we assume that restriction sites now occur according to a Poisson process with rate λ per bp. Then the probability of k sites in an interval of length l bp is

$$\mathbb{P}(N = k) = \frac{e^{-\lambda l} (\lambda l)^k}{k!}, \quad k = 0, 1, 2, \dots$$

- We can also calculate the probability that a restriction fragment length X is larger than x . If there is a site at y , then the length of that fragment is greater than x if there are no events in the interval $(y, y + x)$:

$$\mathbb{P}(X > x) = \mathbb{P}(\text{no events in } (y, y + x)) = e^{-\lambda x}, \quad x > 0.$$

Distribution of restriction fragment lengths

- The previous has some important consequences:

$$\mathbb{P}(X \leq x) = \int_0^x f(y)dy = 1 - e^{-\lambda x},$$

so that the density function for X is given by

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0.$$

- The distance between restriction sites therefore follows an exponential distribution with parameter λ
 - The mean distance between restriction sites is $1/\lambda$

Simulating restriction fragment lengths

- If we *simulated* a sequence using the iid model, we could compute the fragment sizes in this simulated sequence and visualize the result
- R code simulating a DNA sequence having 48500 positions and uniform base probabilities:

```
x<-c(1:4)
propn <- c(0.25,0.25,0.25,0.25)
seq2 <- sample(x,48500,replace=TRUE,prob=propn)
seq2[1:15]
length(seq2[])
```

Simulating restriction fragment lengths

- R code identifying the restriction sites in a sequence string, with bases coded numerically:

```
rsite <- function(inseq, seq){  
  # inseq: vector containing input DNA sequence,  
  # A=1, C=2, G=3, T=4  
  # seq: vector for the restriction site, length m  
  # Make/initialize vector to hold site positions found in inseq  
  xxx <- rep(0,length(inseq))  
  m <-length(seq)  
  # To record whether position of inseq matches seq  
  truth <- rep(0,m)
```

```
# Check each position to see if a site starts there
for (i in 1:(length(inseq) - (length(seq) -1))) {
  for (j in 1:m) {
    if (inseq[i+j-1]==seq[j]) {
      truth[j] <- 1 # Record match to jth position
    }
  }
  if (sum(truth[]) == m) { # Check whether all positions match
    xxx[i] <- i      # Record site if all positions match
  }
  truth <- rep(0,m) # Reinitialize for next loop cycle
}
# Write vector of restriction sites positions stored in xxx
L <- xxx[xxx>0]
return(L)
}
```

Simulating restriction fragment lengths

- The restriction sites we look for are for *AluI*, AGCT.

R code invoking the appropriate function:

```
alu1 <- c(1,3,2,4)
alu.map <- rsite(seq2,alu1)
length(alu.map)
alu.map[1:10]
```

- The fragment lengths can be obtained by subtracting positions of successive sites

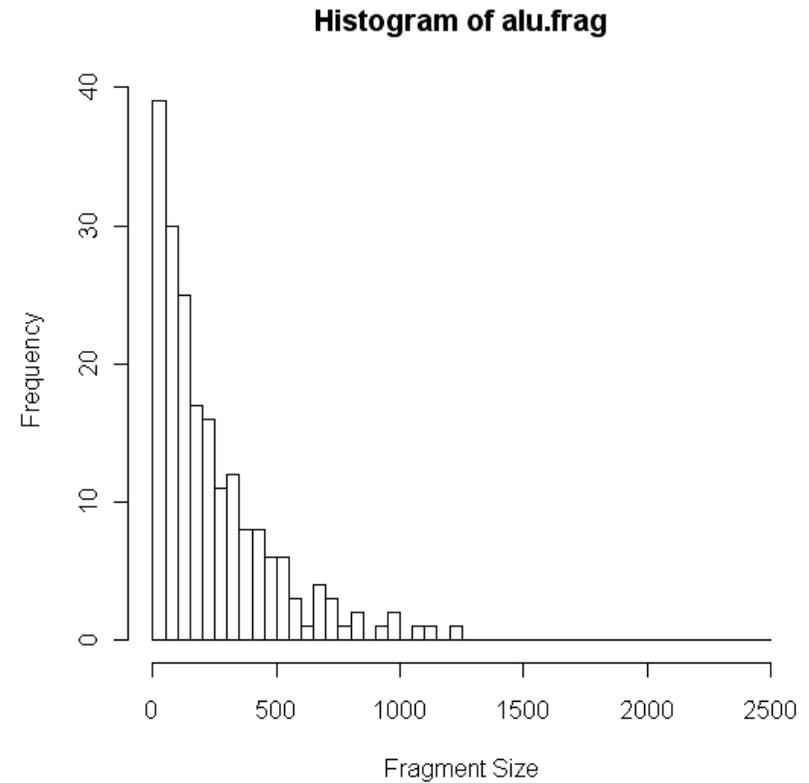
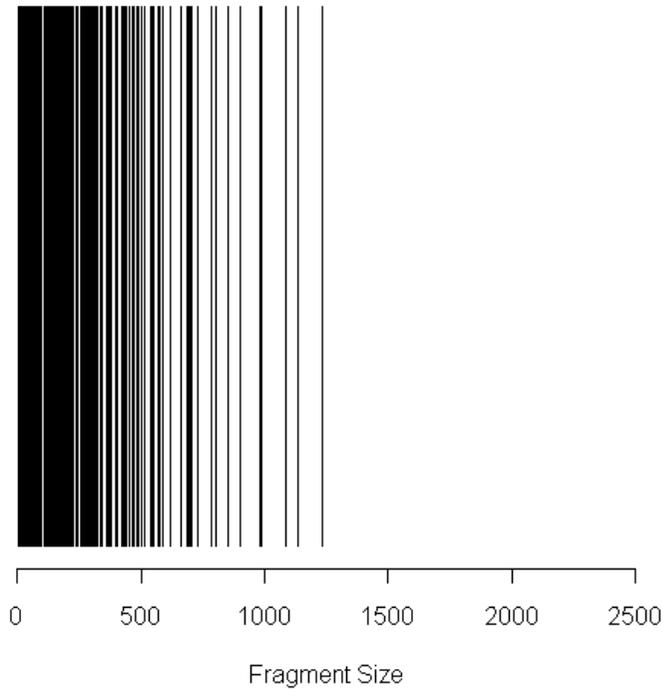
Simulating restriction fragment lengths

- R code doing it for you:

```
flengthr <- function(rmap,N){  
  # rmap is a vector of restriction sites for a linear molecule  
  # N is the length of the molecule  
  frags <- rep(0,length(rmap))  
  # Vector for substraction results: elements initialized to 0  
  rmap <-c(rmap,N)  
  # Adds length of molecule for calculation of end piece  
  for(i in 1:(length(rmap)-1)){  
    frags[i] <- rmap[i+1]-rmap[i]  
  }  
  frags <- c(rmap[1],frags) # First term is left end piece  
  return(frags)  
}
```

```
alu.frag <- flengthr(alu.map,48500)  
alu.frag[1:10]
```

Simulating restriction fragment lengths



Simulating restriction fragment lengths

- The previous plots were obtained via the R code:

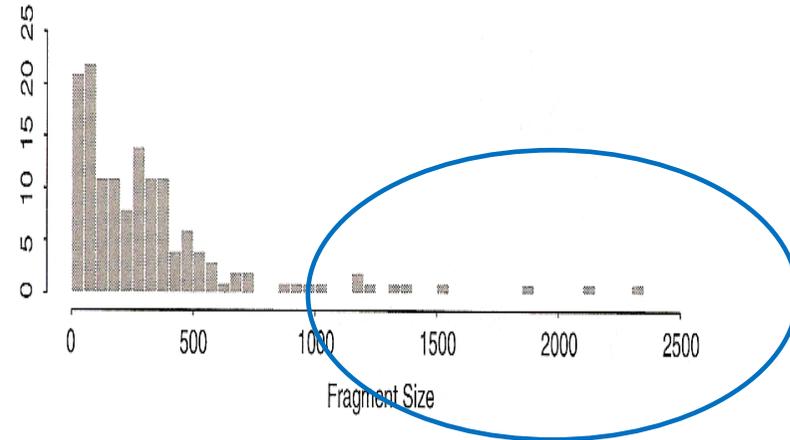
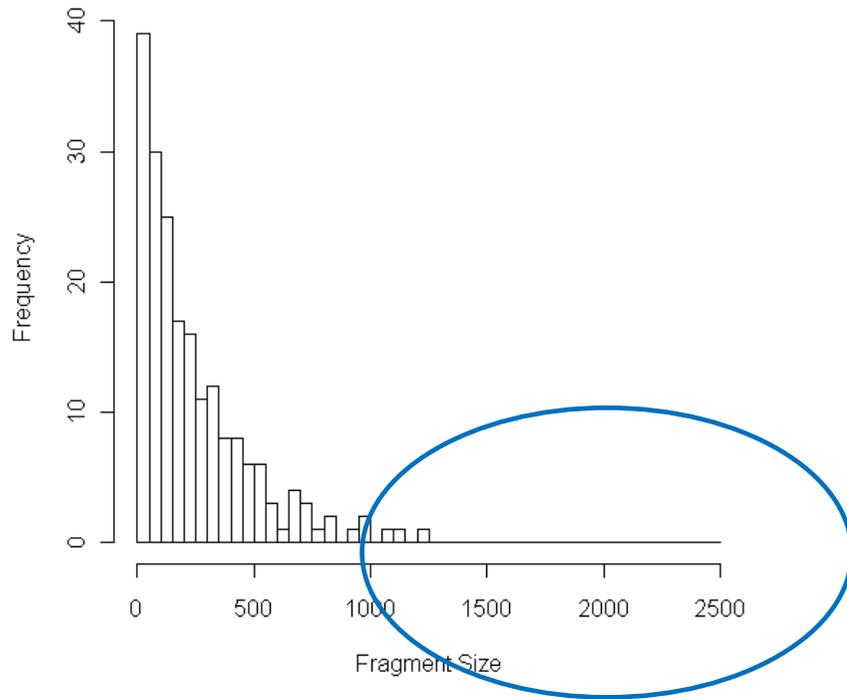
```
plot(c(0,2500),c(3,1),xlab="Fragment Size",ylab="",type="n",axes=F)
axis(1,c(0,500,1000,1500,2000,2500))
for (i in 1:length(alu.frag)){
lines(c(alu.frag[i],alu.frag[i]),c(1,3))
}
hist(alu.frag,breaks=seq(0,2500,50), freq = TRUE,xlab="Fragment Size")
```

- The main important question is:

Is our theoretical model still ok when looking at
restriction fragment lengths?

Simulating restriction fragment lengths

Histogram of alu.frag



Histogram of fragment sizes (bp) produced by AluI digestion of bacteriophage lambda DNA

Histogram based on theoretical model

Simulating restriction fragment lengths

- To determine whether the actual distribution differs significantly from the mathematical model (exponential distribution), we could break up the length axis into a series of "bins" and calculate the expected number of fragments in each bin by using the exponential density.
- We could then compare the observed with expected number of fragments (using the same bin boundaries) via for instance a χ^2 – test.

7 Sequence comparisons

7.a The biological problem

Introduction

- Sequence comparisons are important for a number of reasons.
 - First, they can be used to establish evolutionary relationships among organisms using methods analogous to those employed for anatomical characters.
 - Second, comparison may allow identification of functionally conserved sequences (e.g., DNA sequences controlling gene expression).
 - Finally, comparisons between humans and other species may identify corresponding genes in model organisms, which can be genetically manipulated to develop models for human diseases.

Evolutionary basis of alignment

- This lies in the fact that sequence alignment enables the researcher to determine if two sequences display sufficient similarity to justify the inference of homology.
- Understanding the difference between similarity and homology is of utmost importance:
 - Similarity is an observable quantity that may be expressed as a % identity or some other measure.
 - Homology is a conclusion drawn from the data that the two genes share a common evolutionary history.

(S-star Subbiah)

Evolutionary basis of alignment

- Genes are either homologous or not homologous.
- There are no degrees of homology as are there in similarity.
- While it is presumed that the homologous sequences have diverged from a common ancestral sequence through iterative molecular changes we do not actually know what the ancestral sequence was.

Homology versus similarity

- Hence, sequence similarity is not sequence homology and can occur by chance ...
- If two sequences have accumulated enough mutations over time, then the similarity between them is likely to be low.
- Consequently, homology is more difficult to detect over greater evolutionary distances

#mutations

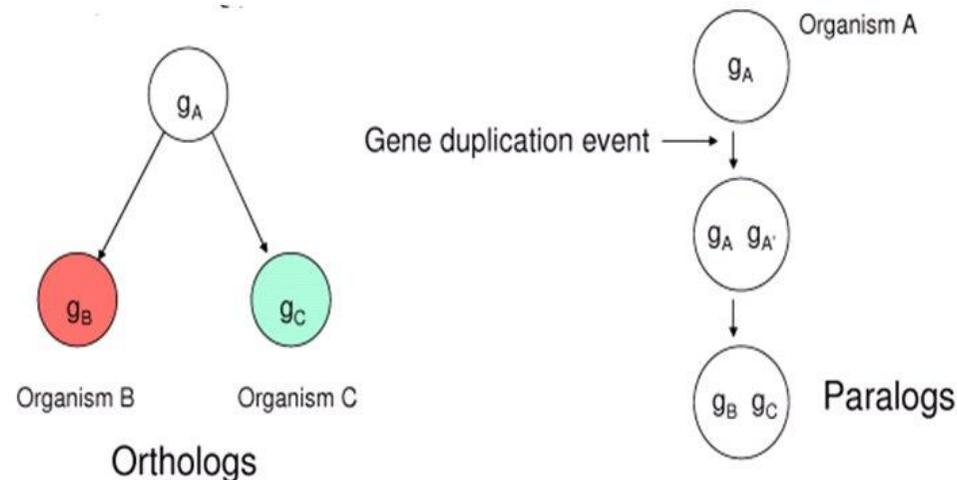
0 agt gt ccgt t aagt gcgt t c
 1 agt gt ccgt t at agt gcgt t c
 2 agt gt ccgct t at agt gcgt t c
 4 agt gt ccgct t aagggcgt t c
 8 agt gt ccgct t caaggggcgt
 16 gggccgt t cat gggggt
 32 gcagggcgt cact gagggct

#mutations

64 acagt ccgt t cgggct at t g
 128 cagagcact accgc
 256 cacgagt aagat at agct
 512 t aat cgt gat a
 1024 accct t at ct act t cct ggagt t
 2048 agcgacct gcccaa
 4096 caaac

Orthologs and paralogs

- We distinguish between two types of homology
 - Orthologs: homologs from two different species, separated by a, what is called, a *speciation event*. Orthologs typically retain the original function
 - Paralogs: homologs within a species, separated by a *gene duplication event*. A copy is free to mutate and acquire new function



7.b Pairwise alignment

Introduction

- There are many different ways that two strings might be aligned. Ordinarily, we expect homologs to have more matches than two randomly chosen sequences.
- Example (matches are indicated by . and - is placed opposite bases not aligned):

ACGTCTAG	2 matches
..	5 mismatches
ACTCTAG-	1 not aligned

Introduction

- We might instead have written the sequences

ACGTCTAG	5 matches
.....	2 mismatches
-ACTCTAG	1 not aligned

- We might also have written

ACGTCTAG	7 matches
..	0 mismatches
AC-TCTAG	1 not aligned

Which alignment is better?

What does better mean?

Introduction

- Next, consider aligning the sequence TCTAG with a long DNA sequence:

```
...AACTGAGTTTACGCTCATAGA...  
      T---CT-A--G
```

- We might suspect that if we compared any string of modest length with another very long string, we could obtain perfect agreement if we were allowed the option of "not aligning" with a sufficient number of letters in the long string.
- Clearly, we would prefer some type of parsimonious alignment. One that does not postulate an excessive number of letters in one string that are not aligned opposite identical letters in the other.

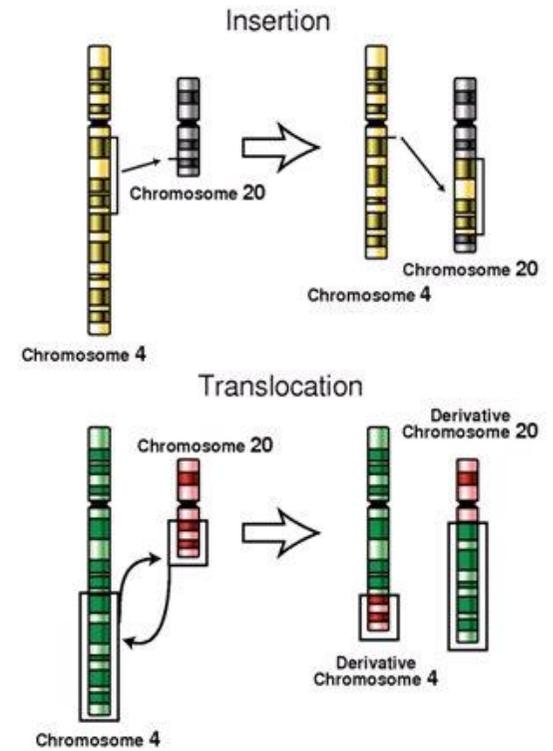
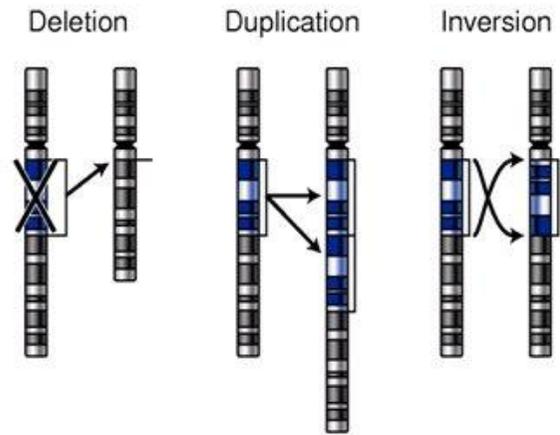
Introduction

- We have seen that there are multiple ways of aligning two sequence strings, and we may wish to compare our **target string** (target meaning the given sequence of interest) to entries in databases containing more than 10⁷ sequence entries or to collections of sequences billions of letters long.
- How do we do this?
- We differentiate between alignment of two sequences with each other
 - which can be done using the entire strings (**global alignment**)
 - or by looking for shorter regions of similarity contained within the strings that otherwise do not significantly match (**local alignment**)
 - and **multiple-sequence alignment** (alignment of more than two strings)

Introduction

- The approach adopted is guided by biology
 - It is possible for evolutionarily related proteins and nucleic acids to display substitutions at particular positions (resulting from known mutational processes).
 - Also, it is possible for these to display insertions or deletions (less likely than substitution).
 - In total, the DNA sequence can be modified by several biological processes (cfr next slide)

Types of mutations



NIH – National Human Genome Research Institute)

Introduction

- Mutations such as segmental duplication, inversion, translocation often involve DNA segments larger than the coding regions of genes.
- They usually do not affect the type of alignment that we are currently interested in
- Point mutations, insertion or deletion of short segments are important in aligning targets whose size are less than or equal to the size of coding regions of genes, and need to be explicitly acknowledged in the alignment process.

Introduction

- Consider again the following alignment

ACGTCTAG	7 matches
..	0 mismatches
AC-TCTAG	1 not aligned

- We can't tell whether the string at the top resulted from the insertion of G in ancestral sequence ACTCTAG or whether the sequence at the bottom resulted from the deletion of G from ancestral sequence ACGTCTAG.
- For this reason, alignment of a letter opposite nothing is simply described as **an indel**.

7.c Global alignment

... using a toy example

- Suppose that we wish to align WHAT with WHY.
- Our goal is to find the highest-scoring alignment. This means that we will have to devise a scoring system to characterize each possible alignment.
- One possible alignment solution is

WHAT

WH-Y

- However, we need a rule to tell us how to calculate an alignment score that will, in turn, allow us to identify which alignment is best.
- Let's use the following scores for each instance of match, mismatch, or indel:
 - identity (match) +1
 - substitution (mismatch) - μ
 - indel - δ

Toy example

- The minus signs for substitutions and indels assure that alignments with many substitutions or indels will have low scores.
- We can then define the score S as the sum of individual scores at each position:

$$S(\text{WHAT/WH} - \text{Y}) = 1 + 1 - \delta - \mu$$

- There is a more general way of describing the scoring process (not necessary for "toy" problems such as the one above).
- In particular: write the target sequence (the sequence we want to compare; WHY) and the search space as columns and rows of a matrix:

Toy example

	-	W	H	A	T
-					
W		x			
H			x	x	
Y					x

- We have placed an x in the matrix elements corresponding to a particular alignment
- We have included one additional row and one additional column for initial indels (-) to allow for the possibility (not applicable here) that alignments do not start at the initial letters (W opposite W in this case).

Toy example

- We can indicate the alignment

WHAT

WH-Y

as a path through elements of the matrix (arrows).

	-	W	H	A	T
-					
W		x			
H			x	x	
Y					x

- If the sequences being compared were identical, then this path would be along the diagonal.

Toy example

- Other alignments of WHAT with WHY would correspond to paths through the matrix other than the one shown.
- Each step from one matrix element to another corresponds to the incremental shift in position along one or both strings being aligned with each other, and we could write down in each matrix element the running score up to that point instead of inserting x.

	-	W	H	A	T
-	0				
W		1			
H			2	$2-\delta$	
Y					$2-\delta-\mu$

Toy example

- What we seek is the path through the matrix that produces the greatest possible score in the element at the lower right-hand corner.
- That is our "destination," and it corresponds to having used up all of the letters in the target string (first column) and search space (first row)-this is the meaning of *global alignment*.
- Using a scoring matrix such as this employs a particular trick of thinking.

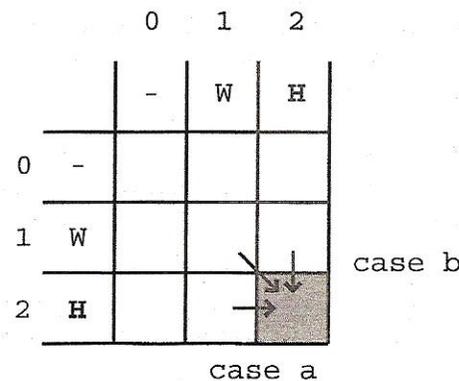
Dynamic programming

- Dynamic programming; a computer algorithmic technique invented in the 1940's, with applications to many types of problems.
- Key properties of problems solvable with DP include that the optimal solution typically contains optimal solutions to subproblems, and only a “small” number of subproblems are needed for the optimal solution.

(T.H. Cormen et al., Introduction to Algorithms, McGraw-Hill 1990).

Toy example (continued)

- The best alignment is revealed by beginning at the destination (lower right-hand corner matrix element) and working backward, identifying the path that maximizes the score at the end.



- To do this, we will have to calculate scores for all possible paths into each matrix element from its neighbouring elements above, to the left, and diagonally above.

Toy example

- There are three possible paths into element (3, 3) (aligning left to right with respect to both strings; letters not yet aligned are written in parentheses):

Case a. \longrightarrow

If we had aligned WH in WHY with W in WHAT (corresponding to element (2, 1)), adding H in WHAT without aligning it to H in WHY corresponds to an insertion of H (relative to WHY) and advances the alignment from element (2, 1) to element (2,2) (horizontal arrow):

(W) H (AT)
(WH) - (Y)

Toy example

Case b.



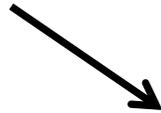
If we had aligned W in WHY with WH in WHAT (corresponding to element (1, 2)), adding the H in WHY without aligning it to H in WHAT corresponds to insertion of H (relative to WHAT) and advances the alignment from element (1, 2) to element (2, 2) (vertical arrow):

(WH)-(AT)

(W)H (Y)

Toy example

Case c.



If we had aligned W in WHY with W in WHAT (corresponding to element (1,1)), then we could advance to the next letter in both strings, advancing the alignment from (1,1) to (2,2) (diagonal arrow above):

(W)H(AT)
(W)H (Y)

- Note that horizontal arrows correspond to adding indels to the string written vertically and that vertical arrows correspond to adding indels to the string written horizontally.

Toy example

- Associated with each matrix element (x, y) from which we could have come into $(2,2)$ is the score $S_{x,y}$ up to that point.
- Suppose that we assigned scores based on the following scoring rules:

identity (match)	+1
substitution (mismatch)	-1
indel	-2

- Then the scores for the three different routes into $(2,2)$ are

$$\text{Case a : } S_{2,2} = S_{2,1} - 2,$$

$$\text{Case b : } S_{2,2} = S_{1,2} - 2,$$

$$\text{Case c : } S_{2,2} = S_{1,1} + 1.$$

Toy example

- The path of the cases a, b, or c that yields the highest score for $S_{2,2}$ is the preferred one, telling us which of the alignment steps is best.
- Using this procedure, we will now go back to our original alignment matrix and fill in all of the scores for all of the elements, keeping track of the path into each element that yielded the maximum score to that element.
- The initial row and column labeled by (-) corresponds to sliding WHAT or WHY incrementally to the left of the other string without aligning against any letter of the other string.
- Since penalties for indels are -2, the successive elements to the right or down from element (0, 0) each are incremented by -2 compared with the previous one.
- Aligning (-) opposite (-) contributes nothing to the alignment of the strings, so element (0,0) is assigned a score of zero.

Toy example

- The remaining elements of the matrix are filled in by the same procedure, with the following result:

		0	1	2	3	4
	-	0	-2	-4	-6	-8
0	-	0	-2	-4	-6	-8
1	W	-2	+1	-1	-3	-5
2	H	-4	-1	2	0	-2
3	Y	-6	-3	0	1	-1

- The final score for the alignment is $S_{3,4} = -1$.
- We just completed a **global alignment** exercise
- Global alignment efforts are not unique

Toy example

- The path through element (2,3) (upper path, bold arrows) corresponds to the alignment

WHAT

WH-Y

which is read by tracing back through all of the elements visited in that path.

- The lower path (through element (3, 3)) corresponds to the alignment

WHAT

WHY-

- Each of these alignments is equally good (two matches, one mismatch, one indel).

Main reference for this chapter:

- Deonier et al. *Computational Genome Analysis*, 2005, Springer.
(Chapters 6,7)

Preparatory Reading for next theory class:

- Balding 2006. A tutorial on statistical methods for population association studies. *Nature Reviews Genetics* 7: 781-.