# B I O I N F O R M A T I C S


## Kristel Van Steen, PhD[2]

**Montefiore Institute - Systems and Modeling**

**GIGA - Bioinformatics**

**ULg**

**kristel.vansteen@ulg.ac.be**

# CHAPTER 4: 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

# 1 The biological problem

## 1.a Questions

### What does sequence analysis enclose?

Sequence analysis includes:

- Multiple sequence alignment,
- sequence searches and clustering;
- prediction of function and localisation;
- novel domains and motifs;
- prediction of protein, RNA and DNA functional sites and other sequence features.

(Bioinformatics Journal scope)

## Types of analyses

- GC content
- Pattern analysis
- Translation (Open Reading Frame detection)
- Gene finding
- Mutation
- Primer design
- Restriction map

## GC content studies:

- Stability
  - GC: 3 hydrogen bonds
  - AT: 2 hydrogen bonds
- Codon preference
- GC rich fragment has increased probability to point towards a gene

What is a CpG island?

## CpG island

- The CG island is a short stretch of DNA in which the frequency of the CG sequence is higher than other regions. It is also called the CpG island, where "p" simply indicates that "C" and "G" are connected by a phosphodiester bond.
- CpG islands are often located around the promoters of housekeeping genes (which are essential for general cell functions) or other genes frequently expressed in a cell.
- At these locations, the CG sequence is not methylated.

(http://www.web-books.com/MoBio/Free/Ch7F2.htm)

## CpG island

- By contrast, the CG sequences in inactive genes are usually methylated to suppress their expression. The methylated cytosine may be converted to thymine by accidental deamination. Unlike the cytosine to uracil mutation which is efficiently repaired, the cytosine to thymine mutation can be corrected only by the mismatch repair which is very inefficient.
- Hence, over evolutionary time scales, the methylated CG sequence will be converted to the TG sequence. This explains the deficiency of the CG sequence in inactive genes.

(http://www.web-books.com/MoBio/Free/Ch7F2.htm)

## Relevant questions

Given the DNA sequence

AATCGGATGCGCGTAGGATCGGTAGGGTAGGCTTTAAGATCATGCTATTTTCGAGA
TTCGATTCTAGCTAGGTTTAGCTTAGCTTAGTGCCAGAAATCGGATGCGCGTAGGAT
CGGTAGGGTAGGCTTTAAGATCATGCTATTTTCGAGATTCGATTCTAGCTAGGTTTT
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?
  - Cenromere / Telomere?

# Tools to answer the questions

# Tools to answer the questions


BioEdit.zip (Full install)

Bug fixes / changes

BioEdit General information

BioDoc.pdf (pdf format help doc)

View Screenshots

**BioEdit** is a biological sequence alignment editor written for Windows 95/98/NT/2000/XP. An intuitive multiple document interface with convenient features makes alignment and manipulation of sequences relatively easy on your desktop computer. Several sequence manipulation and analysis options and links to external anaylsis programs facilitate a working environment which allows you to view and manipulate sequences with simple point-and-click operations.

BioEdit's features include: *New version is WinXP compatible*

- Several modes of hand alignment
- Automated ClustalW alignment
- Automated Blast searches (local and WWW)
- Plasmid drawing and annotation
- Accessory application configuration
- Restriction mapping
- RNA comparative analysis tools
- Graphical matrix data viewing tools
- Shaded alignment figures
- Translation-based nucleic acid alignment
- ABI trace viewing, editing and printing
- Customizable ... other features

**Note:** Although BioEdit was recently updated, it is no longer being reliably maintained, and the documentation is out of date and no longer maintained. It is being updated slowly, but there is no guaranteed finish date. Until documentation is complete, play with the menus and see what happens, or email with a question.

Note: Scoring in pairwise alignment functions does not agree with the example in the help document. This will be investigated if/when I have time. I had considered removing the functions, but decided no to on the grounds that they generally work OK for quick manipulations and can be useful as a time-saving utility. They should not, however, be used as a teaching tool.

## Recall: Different cell types

- *Eukaryotes*: organisms with a rather complex cellular structure. In their cells we find organelles, clearly discernable compartments with a particular function and structure.
  - The organelles are surrounded by semi-permeable membranes that compartmentalize them further in the cytoplasm.
  - The Golgi apparatus is an example of an organelle that is involved in the transport and secretion of proteins in the cell.
  - Mitochondria are other examples of organelles, and are involved in respiration and energy production

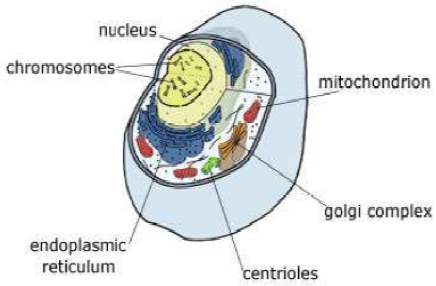
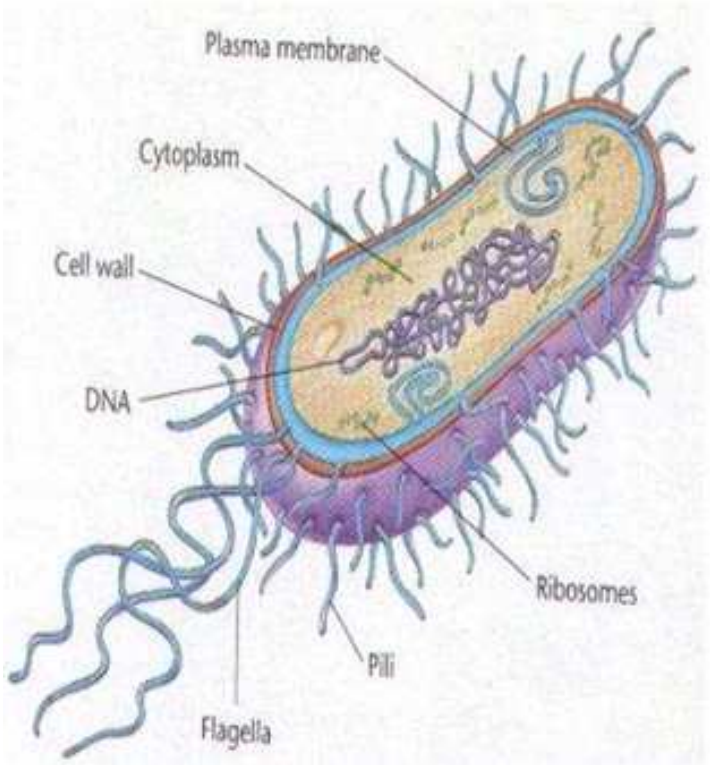


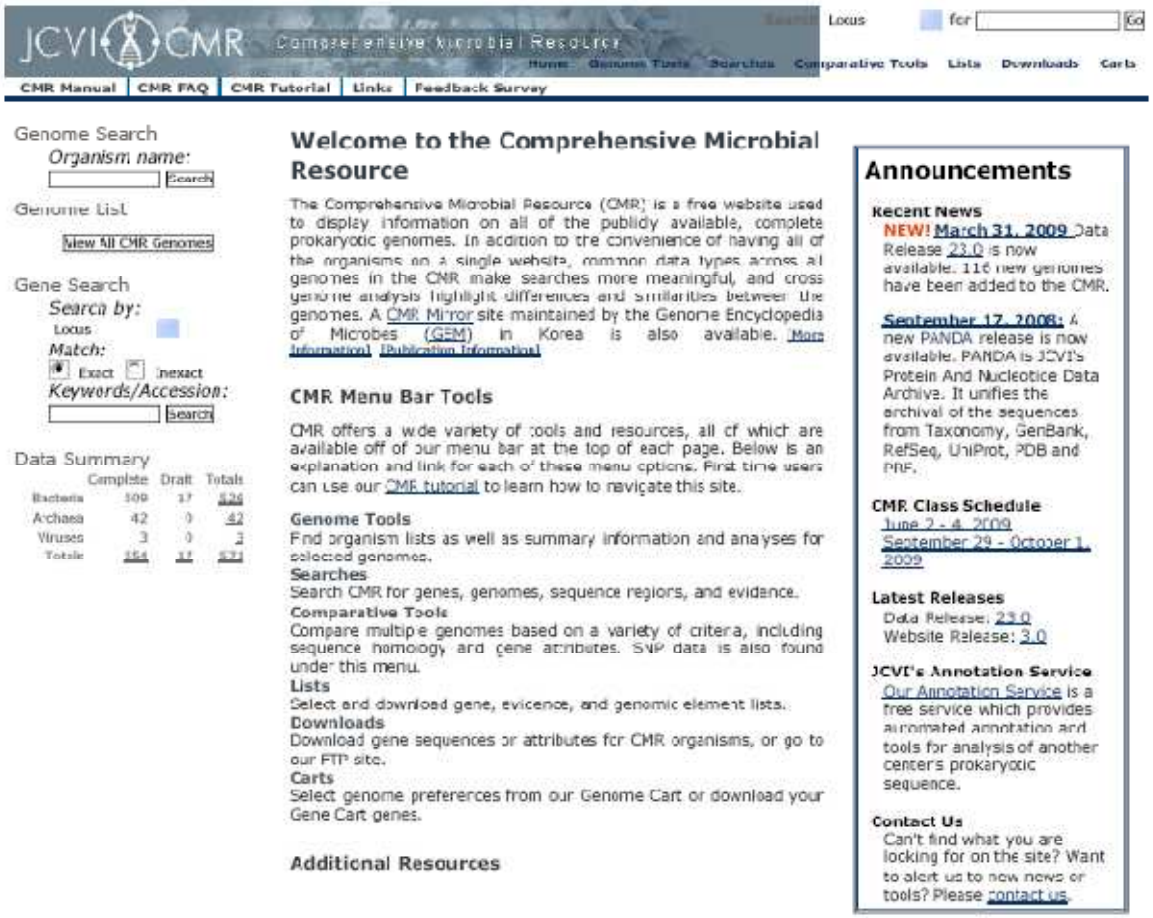

Image adapted from: National Human Genome Research Institute.
A typical eukaryotic cell.

## Recall: Different cell types

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



# For instance, base composition for bacterial data:




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

# Focus on human data and Bioconductor / R Environment

## Methods to answer the questions

For instance by investigating frequencies of occurrences of words

## 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

- For free-living organisms, DNA is typically duplex
  - Every A (G) on one strand is matched by a T (C) on the complementary strand
- Note the difference with bacteriophages (viruses that infect bacteria)
  - Bacterium, the singular form of the word bacteria, is a one-celled living organism, with complete sets of both ribonucleic acid (RNA) and deoxyribonucleic acid (DNA) genetic codes.
  - A virus is little more than a section of RNA or DNA strand covered by a protein shell.

## Biological words of length 1

- There are constraints on base composition imposed by the genetic code: fr(C+G)=0?
- 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 (strands whose 5’ to 3’ direction corresponds to the direction of replication fork movement)
  - This can be described by the “GC skew”, characterized by:
    - (#G - #C) / (#G + #C)
    - # = nr of

# 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

- Statistical intervals and hypothesis tests are often based on specific distributional assumptions. 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 usually rely on random numbers generation. These generations are from a specific probability distribution …

## Introduction

- Discrete probability functions are referred to as probability mass functions and continuous probability functions are referred to as probability density functions.
    - The term probability function covers both discrete and continuous distributions. When we are referring to probability functions in generic terms, we may use the term probability density functions to mean both discrete and continuous probability functions.

## Assumptions

- Consider the nucleotide sequence on a single strand written in a given 5' to 3' direction
- 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. A, C, T or G is generated with the probabilities above
- Notation for the output of a random string of *n* bases: $L_1, L_2, \ldots, L_n$ ($L_i$ = base inserted at position *i* of the sequence)

## Probability distributions

- Suppose the "machine" we are using produces an output X that takes exactly 1 of the J possible values in a set $\chi = \{x_1, x_2, \ldots, x_n\}$
  - In the DNA sequence J=4 and $\chi = \{A, C, T, G\}$
  - X is a discrete random variables (since its values are uncertain)
  - If $p_j$ is the prob that the value $x_j$ occurs, then
    - $p_1, \ldots, p_J \geq O$ and $p_1 + \ldots + p_J = 1$
- The probability distribution (probability mass function) of X is given by the collection $p_1, \ldots, p_J$
  - $P(X=x_j) = p_j$, $j$=1, …, J
- The probability that an event S occurs (subset of $\chi$) is P(X ∈ $S$) = $\sum_{j:x_j \in S} (p_j)$

## Meaning of the probability distribution

- A discrete probability function is a function that can take a discrete number of values (not necessarily finite). This is most often the non-negative integers or some subset of the non-negative integers.
- There is no mathematical restriction that discrete probability functions only be defined at integers, but in practice this is usually what makes sense.
  - For example, if you toss a coin 6 times, you can get 2 heads or 3 heads but not 2 1/2 heads.
  - Each of the discrete values has a certain probability of occurrence that is between zero and one. That is, a discrete function that allows negative values or values greater than one is not a probability function. The condition that the probabilities sum to one means that at least one of the values has to occur.

## Probability distributions

- What is the probability distribution of the number of times a given pattern occurs in a random DNA sequence $L_1, \ldots, L_n$?
  - New sequence $X_1, \ldots, X_n$:

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

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

$$N=X_1+\ldots+X_n$$

  - The prob distr of each of the $X_i$:

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

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

- What is a "typical" value of N?
  - Depends on how the individual $X_i$ (for different *i*) are interrelated

## Independence

- Discrete random variables $X_1$, …, $X_n$ 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

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

## 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 $x_1$, $x_2$, . . . (denoted EX, 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
  - $E(X_1 + \ldots + X_n) = EX_1 + \ldots + EX_n$
- Because $X_i$ are assumed to be independent and identically distributed (iid):

$$E(X_1 + \ldots + X_n) = nEX_1 = np_A$$

## Expected values and variances

- 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.

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

  - The positive square root of the variance of X is called its standard deviation sd(X)

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

## 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 Var(X) can also be written as:

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

  - If Y=cX then VarY = $c^2$VarX
  - The variance of a sum of independent random variables is the sum of the individual variances

- For the random variables $X_i$:
  $\mathrm{Var}X_i = [1^2 \times p_A + 0^2 \times' (1 - p_A)] - p_A^2 = p_A(1 - p_A)$
  $\mathrm{Var}N = n\mathrm{Var}X_1 = np_A(1 - p_A)$

## 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 :

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

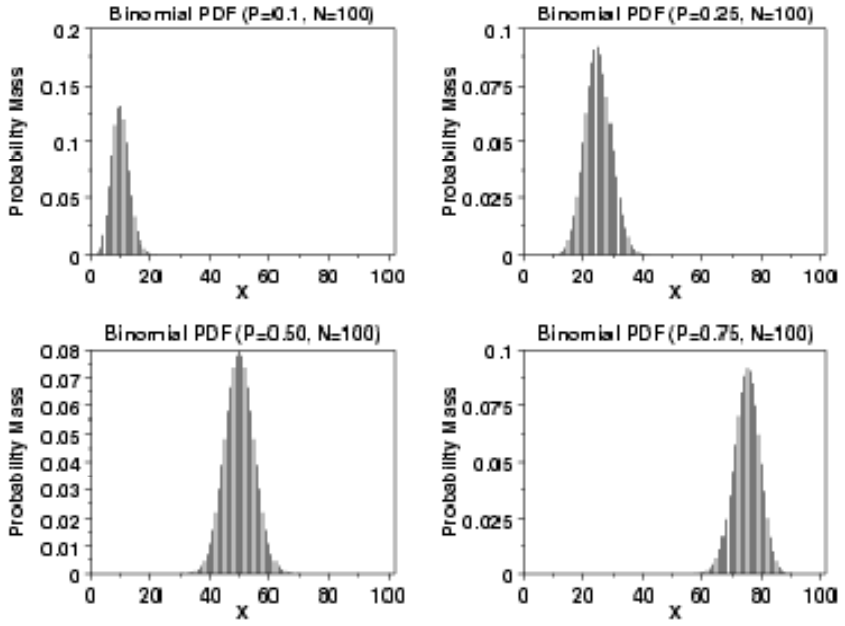
with the binomial coefficient $\binom{n}{j}$ determined by

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

and j!=j(j-1)(j-2)…3.2.1, 0!=1

# The binomial distribution

- The mean is np and the variance is np(1-p)
- 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 is that we can study the properties of the distribution of N when we can get our computer to output numbers $N_1$, …, $N_n$ having the same distribution as N
  - We can use the sample mean to estimate the expected value EN:

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

  - 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 observations?
  - Access to pseudo-random numbers: random variables that are uniformly distributed on (0,1): any number between 0 and 1 is a possible outcome and each is equally likely
- Simulating an observation with the distribution of $X_1$:
  - Take a uniform random number u
  - 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 $X_1$, …, $X_n$ from which N can be computed by adding the X's

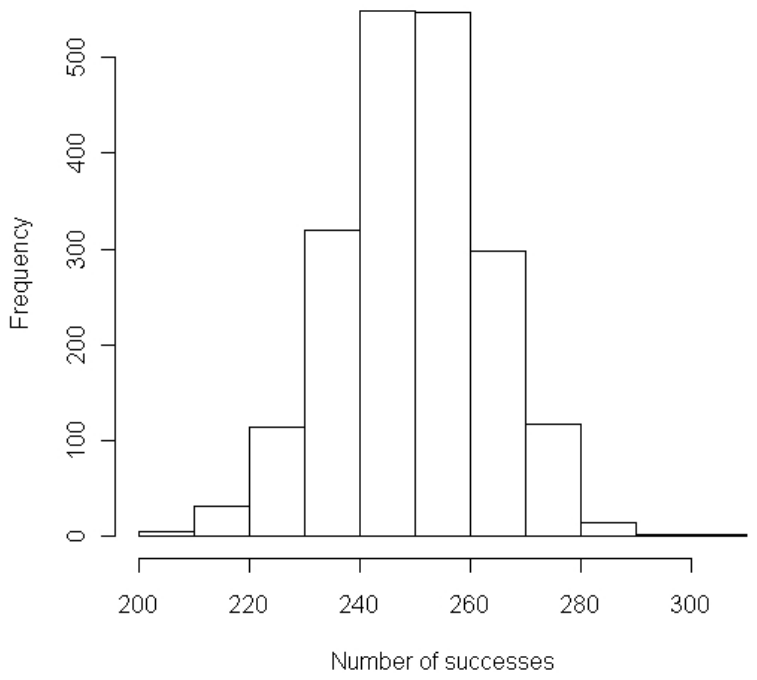
## Simulating from probability distributions

- Simulate a sequence of bases $L_1$, …, $L_n$:
  - 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 $L_1$, …, $L_n$
- 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 :

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

and therefore

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

$$= 0.00019359032194965841$$

| | 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 $X_1$, …, $X_n$ are iid random variables with mean $\mu$ and variance $\sigma^2$, then we know that for the sample average

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

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

- Hence,

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

## Approximate via Central Limit Theory

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

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

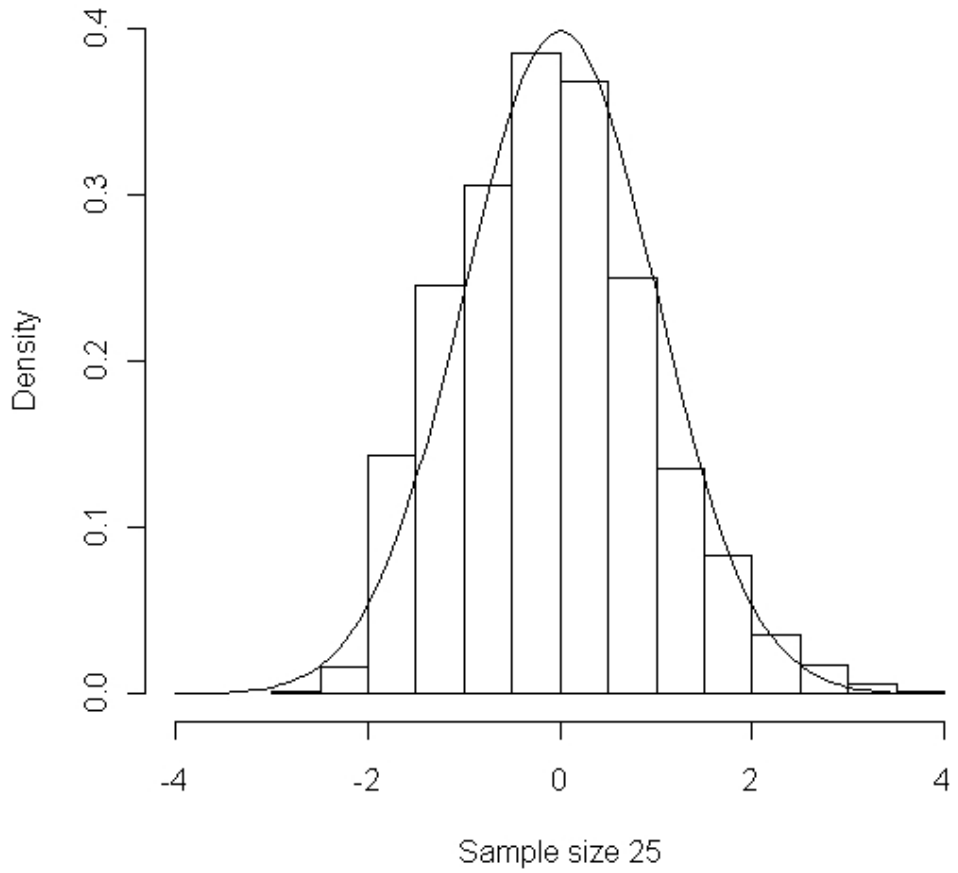
with $\phi(.)$ the standard normal distribution defined as

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

- 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=TRUE,xlab="Sample size
25",main="")
x<-seq(-4,4,0.1)
lines(x,dnorm(x))
```

# Approximate via Central Limit Theory



## 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 $L_1$, …, $L_n$:

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

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

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

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

Why (n-1) as factor?

## Comparing to the reference

- How to determine which values of $\chi^2$ 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 $\chi^2$ 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

## Comparing to the reference

- How to determine which values of $\chi^2$ are unlikely or extreme?
  - Simulate percentage points of the distribution of the statistic $\chi^2$:
    - Generate strings of 1000 letters having distribution e.g., (0.3,0.2,0.3,0.2) for (A,C,T,G)
    - Calculate O, the number of times the pair AC is observed
    - Calculate E and $\frac{\chi^2}{c}$
    - Plot a histogram of these values
    - Compare

What is the theoretical distribution?

# 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 then 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 Ω-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 $B_1$, …, $B_k$ forming a partition of Ω, this is the Bi are disjoint and the Bi are exhaustive (their union Is Ω), the law of total probability holds:

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

## The Markov property

- The property will be explained via studying a sequence of random variables $X_t$, t=0,1,… 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:

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

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:
  $P(X_{t+1} = j | X_t = i) = p_{ij}$ (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.
  - Note that $\sum_j \ p_{ij} = 1$ for DNA sequences
  - How does a one-step transition matrix look like for DNA sequences?
  - Which situation corresponds to the iid model?
- 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 Markov property

- The probability distribution for the states at position 1 can be obtained as follows:

$$P(X_1 = j) = \sum_{i \in \chi} P(X_0 = i, X_1 = j)$$
$$= \sum_{i \in \chi} P(X_0 = i)\, P(X_1 = j | X_0 = i)$$
$$= \sum_{i \in \chi} \pi_i p_{ij}$$

## 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=$P^2$

$$P(X_2 = j|X_0 = i) = \sum_{k \in \chi} P(X_2 = j, X_1 = k|\, X_0 = i)$$
$$= \sum_{k \in \chi} P(X_2 = j|X_1 = k, X_0 = i) P(X_1 = k|\, X_0 = i)$$
$$= \sum_{k \in \chi} P(X_2 = j|X_1 = k) P(X_1 = k|\, X_0 = i)$$
$$= \sum_{k \in \chi} p_{ik} p_{kj} = (PP)_{ij}$$

- Therefore

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

## The Markov property

- In a similar way it can be shown that

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

- 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 $\sum_{i \in \chi} \pi_i p_{ij} = \pi_{j,}$ for all j, or stated differently when $\pi = \pi P$
  - With $\pi_i = P(X_0 = i)$, also $\pi = \pi P^t$ and $P(X_t = j) = \pi_j$

## Creating our own Markov chain simulation in practice

- 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 |

- How to compute the individual base frequencies?
- 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 $L_1$, $L_2$, ... , $L_n$ 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
- It follows that among those codons making up the amino acid Phe, the expected proportion of TTT is

$$\frac{P(TTT)}{P(TTT) + P(TTC)}$$

## Predicted relative frequencies

- 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.

| | | | 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 |

## Predicted relative frequencies

- Medigue et al. (1991) clustered the different genes based on such codon usage patterns.
- They 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

| | Codon | Predicted | Observed Gene Class I (502) | Observed 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 |

## The codon adaptation index

- A statistic that can describe each protein-coding gene for any given organism is the codon adaptation index, or CAI (Sharp and Li, 1987).
- This statistic compares the distribution of codons actually used in a particular protein with the preferred codons for highly expressed genes.
- One might also compare them to the preferred codons based on gene predictions for the whole genome, but the CAI was devised prior to the availability of whole-genome sequences.

.

## The codon adaptation index

- Consider a sequence of amino acids X = $x_1$, $x_2$, ... , $x_L$ 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

- The CAI is defined as

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

- 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

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

- 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.

- 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

Example of codon usage patterns in E. coli for computation of the codon adaptation index of a gene. 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 | ... |
| 1.000 | 0.469 | 0.018 | 0.451 | 0.798 | 0.469 | 0.794 | 1.000 | 0.428 | 0.794 | 0.193 | 0.018 | 0.228 | |
| | 0.057 | 0.018 | 0.468 | 0.202 | 0.057 | 0.206 | | 0.319 | 0.206 | 0.807 | 0.018 | 0.772 | |
| | 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 | | |
| | | 0.888 | | | | | | 0.176 | | | 0.888 | | |
| 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

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

- 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)

## Word distribution and occurrences - The biological problem

- Suppose that we wanted to obtain a sample of DNA that contained a specific gene or portion of a gene with very little other DNA.
- How could we do this?
  - Today, given a genome sequence, we could design PCR primers flanking the DNA of interest and could amplify just that segment by PCR.
  - Prior to the development of rapid genomic sequencing technologies, the process was much more complicated.

## The biological problem

- DNA is a macromolecule: DNA molecules can have very high molecular weights.
- Because DNA can be long but is very thin, it is easily broken by hydrodynamic shear (e.g. due to physical stress induced by nebulisation).
  - Note that the DNA in human chromosome 1, at 245,000,000bp, is 8.33cm long and only 20 x$10^{-8}$ cm thick
- Such a long molecule cannot be transferred from one sample tube to another without breakage during pipetting.
- The result of shearing is a collection of DNA fragments that are not broken at the same position: molecules containing the gene of interest intact might be very rare.

## The biological problem

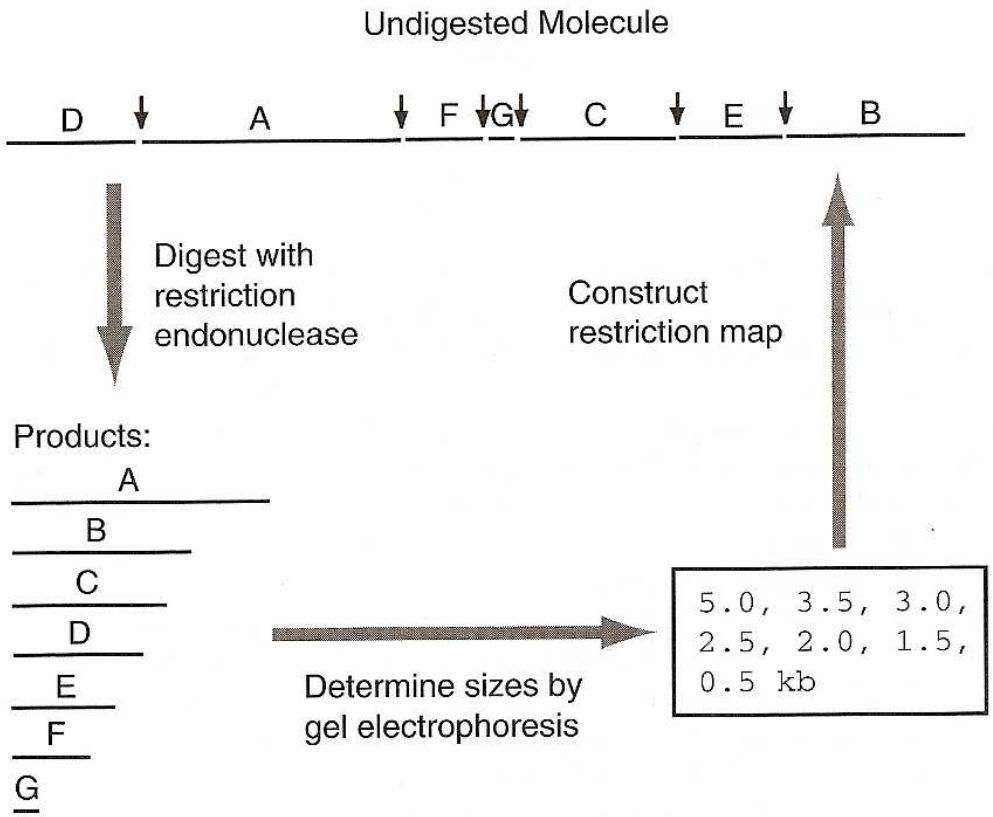
- Restriction endonucleases provides the means for precisely and reproducibly cutting the DNA into fragments of manageable size (usually in the size range of 100s to 1000s of base pairs), and
- molecular cloning provides the method for amplifying the DNA of interest
- Cloning puts DNA of manageable size into vectors that allow the inserted DNA to be amplified, and the reason for doing this is that large molecules cannot be readily manipulated without breakage.

## 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

- Although restriction mapping is not as central as it once was for genome analysis, workers at the bench still use restriction mapping to evaluate the content of clones or DNA constructs of interest
- Hence, being able to determine locations and distributions of restriction endonuclease recognition sites is still relevant.
    - A probabilistic basis is needed for analyzing this kind of problem.
    - **In** addition, word occurrences can be used to characterize biologically significant DNA subsequences.

# 6 Modeling the number of restriction sites in DNA

## Introduction

- Modelling the number of restriction sites in DNA is important when addressing the following questions:
  - 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?
  - Suppose that we observed 761 occurrences of the sequence 5'-GCTGGTGG-3' in a genome that is 50% G+C and 4.6 Mb in size.
    - How does this number compare with the expected number?
    - How would one find the expected number?
    - Expected according to what model?

## Introduction

- We will model the underlying sequence as a string of iid letters and will use this model to find the probability distribution of the number of restriction endonuclease cleavage sites and the distribution of fragment sizes of a restriction digest.
- Because of their occurrence in promoter regions, it is also relevant to inquire about the expected frequencies of runs of letters (such as AAAAAA⋯A tracts).

## Introduction

- While doing so we assume, as before, that the genome of an organism can be represented as a string $L_l, \ldots, L_n$ drawn from the alphabet $\chi = \{a_1, a_2, a_3, a_4\} = \{A, C, G, T\}$, where *n* is the number of base pairs
- Note that if we are given a DNA sample, we usually know something about it; at least which organism it came from and how it was prepared.
  - This means that usually we know its base composition (%G+C) and
  - its approximate molecular weight, useful pieces of information

## 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 *Eco*RI, 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.

## 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

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

- 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 $X_1, X_2, \ldots, X_n$,
- The number of restriction sites is $N = X_1 + X_2 + \text{'''} + X_m$, 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 $X_1, X_2, \ldots, X_n$ 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

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

## Poisson approximation to the binomial distribution

- In preparation for looking at the distribution of restriction fragment lengths, we introduce an approximate formula for P($N = j$) when $N$ has a binomial distribution with parameters $n$ and $p$.
- Using the example for *Eco*RI before with $p = 0.00024$ for DNA that has equal frequencies of the four bases, a molecule that is 50,000 bp long would have 50,000 x 0.00024 = 12 expected sites according to our model.
- Notice that because $p$ is very small, the number of sites is small compared to the length of the molecule.
  - This means that Var$N = np(1 - p)$ will be very nearly equal to E$N = np$.
  - Contrast this with a fair coin-tossing experiment, where $p = 0.5$. With 300 coin tosses, we would have E$N$ = 300 x 0.5 = 150, and Var$N$ = 300 x 0.5 x (1 - 0.5) = 75.

## Poisson approximation to the binomial distribution

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

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

## 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,

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

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

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

- This is the formula for the Poisson distribution with parameter $\lambda = np$ = 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 *Eco*RI 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 *Eco*RI two times or less
- R code:

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

## The Poisson process

- There is a more general version of the Poisson distribution that is very useful. It generalizes *n* into "length" and *p* into "rate." The mean of the corresponding Poisson distribution is length x rate. We suppose that events (which were restriction sites above) are occurring uniformly on a line at rate μ then

$$\mathbb{P}(k \text{ events in } (x, x+l)) = \frac{\mathrm{e}^{-\mu l}(\mu l)^k}{k!}, \ k = 0, 1, 2, \ldots \ .$$

- If there is more than one interval, the lengths of the intervals simply add,

$$\mathbb{P}(k \text{ events in } (x, x+l_1) \cup (y, y+l_2)) = \frac{\mathrm{e}^{-\mu(l_1+l_2)}(\mu(l_1+l_2))^k}{k!}, \ k = 0, 1, 2, \ldots,$$

as long as the intervals are disjoint (i.e., $x < x + l_1 \leq y < y + l_2$).

## Distribution of restriction fragment lengths

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

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

- 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):

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

## Distribution of restriction fragment lengths

- The previous has some important consequences:

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

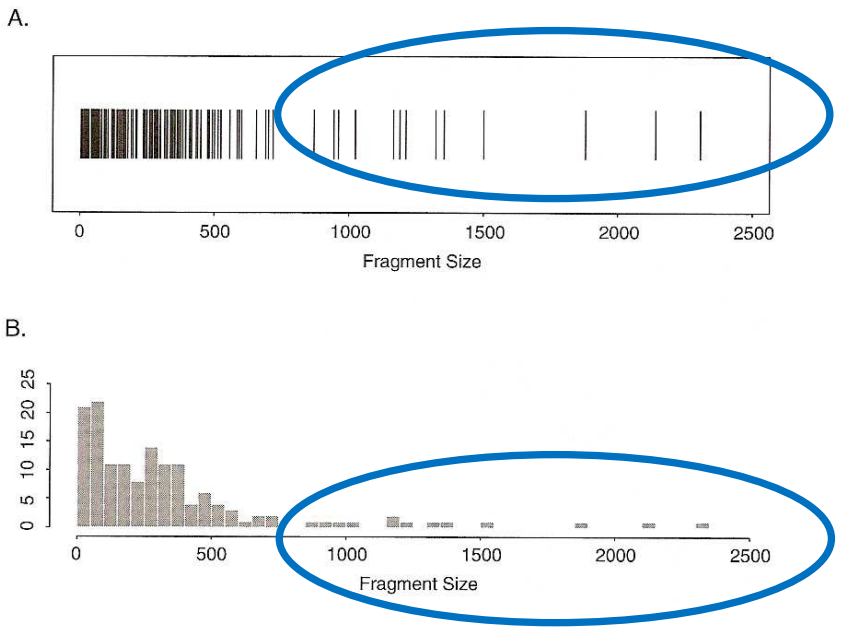
so that the density function for X is given by

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

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

## Simulating restriction fragment lengths

- From the previous, the restriction fragment length (fragment size) distribution should be approximately exponential
- But what would we actually see for a particular sequence conform to the iid model (A*lu*I enzyme with recognition sequence AGCT)?




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

## Simulating restriction fragment lengths

- In other words, if we *simulated* a sequence using the iid model, we could compute the fragment sizes in this simulated sequence and visualize the result in a manner similar to what is seen in the actual case in the figure on the previous slide (Fig. 3.3. Deonier et al 2005)
- 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 A*lu*I, AGCT.
- R code envoking the appropriate function:

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

How close is the actual number of restriction sites to the number predicted by our mathematical model?

## Simulating restriction fragment lengths

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

## Simulating restriction fragment lengths

- R code continued ….

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

What is the largest or smallest fragment?

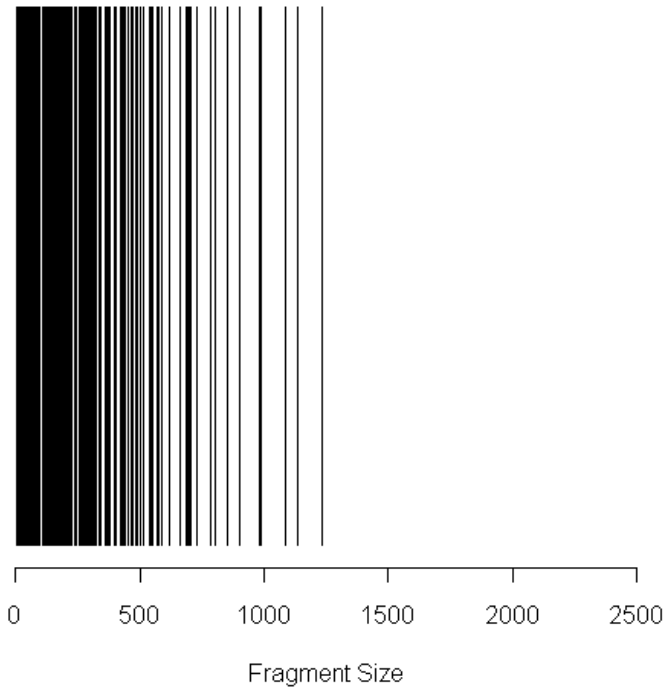
```
max(alu.frag[])
min(alu.frag[])
```

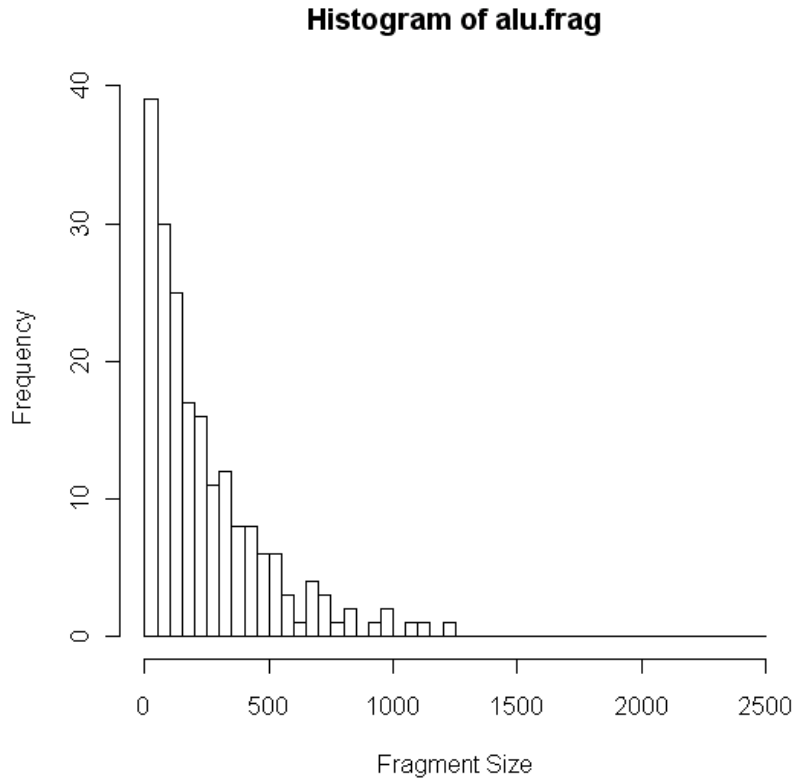
Internal checks

```
length(alu.frag[])
sum(alu.frag[])
```

How come that the length of alu.frag is one more than the length of alu.map?

# Simulating restriction fragment lengths



## Simulating restriction fragment lengths

- The previous plots were obtained via the R code:

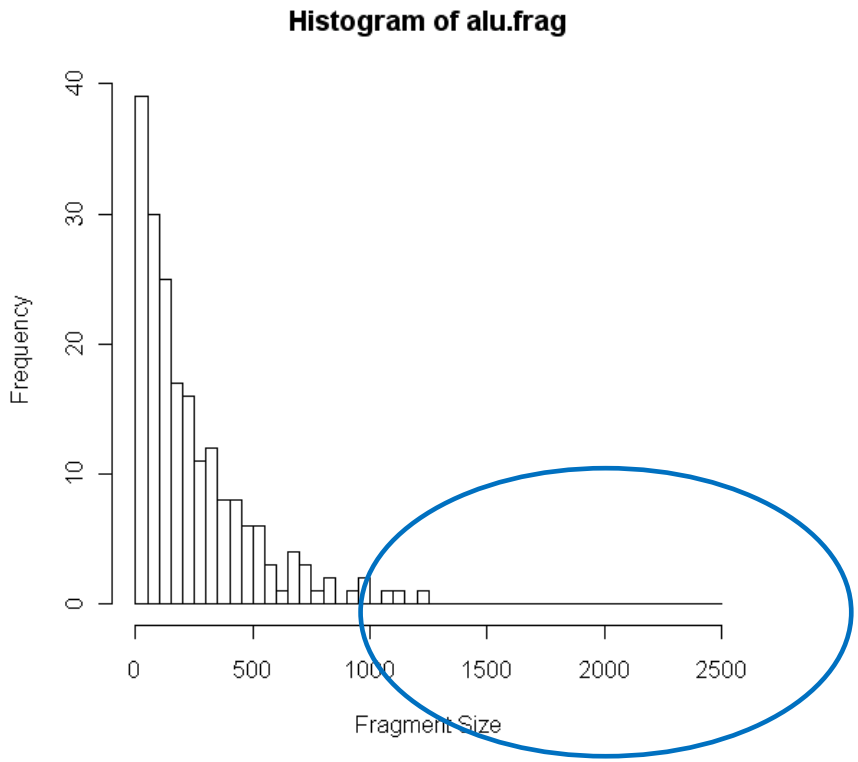
```
plot(c(0,2500),c(3,1),xlab="Fragment Size",ylab="",type="n",axes=F)
axis(1,c(0,500,1000,1500,2000,2500))
for (i in 1:length(alu.frag)){
lines(c(alu.frag[i],alu.frag[i]),c(1,3))
}
hist(alu.frag,breaks=seq(0,2500,50), freq = TRUE,xlab="Fragment Size")
```

- The main important question is:

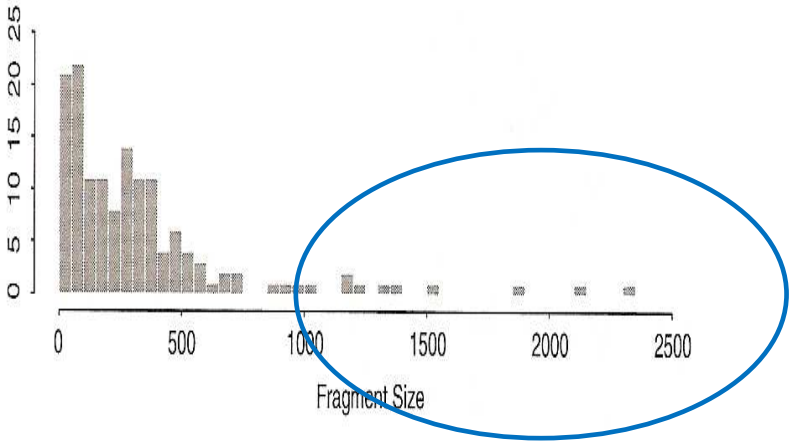
Is our theoretical model still ok when looking at

restriction fragment lengths?

# Simulating restriction fragment lengths




Histogram based on theoretical model




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

## Simulating restriction fragment lengths

- To determine whether the distribution in case of lambda DNA 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.
- This would create the entries for a histogram based on the mathematical model.
- We could then compare the observed distribution of fragments from lambda DNA (using the same bin boundaries) to the expected distribution from the model by using for instance a $\chi^2$ – test.

# Occurrences of k-words (home reading)

## Introduction

- The aforementioned statistical principles can be applied to other practical problems, such as discovering functional sites in DNA.
- We will use promoter sequences as an example.
  - Promoters are gene regions where RNA polymerase binds to initiate transcription.
  - We wish to find k-words that distinguish promoter sequences from average genomic sequences.
  - Because promoters are related by function, we expect to observe k-words that are over-represented within the promoter set compared with a suitable null set.

## Introduction

- Using already known methods, we will determine expected *k*-word frequencies and compare them to the observed frequencies.
- Via theoretical distributions, it can be tested whether over-represented *k*-words appear with significantly higher frequencies than the reference

## Counting k-words in promoter sequences

- Consider *N* promoter sequences of length *L* bp, which we denote by$S_1, \ldots, S_n$.
- The null set might consist of *N* strings of L iid letters, each letter having the same probability of occurrence as the letter frequencies in genomic DNA as a whole.
- Here, we take a small word size, k = 4, so that there are 256 possible k-words. With no a priori knowledge of conserved patterns, we must examine all 256 words.
- Question: Are there an unusual number of occurrences of each word in the promoter region?

## Counting k-words in promoter sequences

- 8 promoter sequences (-75 - +25 to transcriptinal start site) are given in the file promseqex.txt
- The expectation of each 4-word according to the null (iid) model is easily computed:

$$P(w = ACGT) = p_A p_C p_G p_T$$

$$E(\text{nr of times } w \text{ appears in } S\_i\ ) = (L - 4 + 1)\, p_A p_C p_G p_T$$

$$E(X_w) = N(L - 4 + 1)\, p_A p_C p_G p_T$$

with $X_w$ the number of occurrences in N sequences

## Counting k-words in promoter sequences

- R code:

```
ec.prom <- read.table("promseqex1234.txt",sep="",header=F)
ec.prom <- as.matrix(ec.prom)
ec.prom <- ec.prom[,-ncol(ec.prom)]
ncol(ec.prom)
w <- 4 # restricting attention to 4-words

prob.ec <- c(0.246,0.254,0.254,0.246) # base frequencies for the E coli sequence
expect4.ec <- array(rep(0,4^w),rep(4,w)) # 4 is the max value in each dim for w
                                         # there are w dimensions
for (i in 1:4){
  for (j in 1:4){
    for (k in 1:4) {
      for (m in 1:4) {
        expect4.ec[i,j,k,m] <- 8*97*prob.ec[i]*prob.ec[j]*prob.ec[k]*prob.ec[m]
                               # 8 is the number of sequences in this example
```

```
                    # L-w+1 = 100 - 4 + 1 = 97
      }
    }
  }
}


Ncount4 <- function(seq,w){
  # w is length of word
  tcount <- array(rep(0,4^w),rep(4,w))
  # array[4 times 4 times 4 times 4] to hold word counts, elements set to zero
  N <- length(seq[1,]) # length of each sequence
  M <- length(seq[,1]) # number of sequences



##
```

```
 # Count total number of word occurrences
 for (j in 1:M){ # looping over sequences
  jcount <- array(rep(0,4^w),rep(4,w))
  # array to hold word counts for sequence j
  for (k in 1:(N-w+1)){ # looping over positions
  jcount[seq[j,k],seq[j,k+1],seq[j,k+2],seq[j,k+3]] <-
   jcount[seq[j,k],seq[j,k+1],seq[j,k+2],seq[j,k+3]] +1
   # adds 1 if word at k, k+1, k+2, k+3 appears in sequence j
  }
  tcount <- tcount + jcount
  # add contribution of j to total
 }
 return(tcount)
}

 prom.count <- Ncount4(ec.prom,4)
```

## Counting k-words in promoter sequences

## Internal check

sum(prom.count)

What is the most frequent word?

max(prom.count)

How many words occur more at least 10 times?

```
length(prom.count[prom.count[,,,]>=10])
(1:256)[prom.count[,,,]>=10]        # the actual positions
prom.count[prom.count[,,,]>=10] # the actual values
```

## Counting k-words in promoter sequences

How to know to which 4-words the positions refer to?

```
kwordseq <- NULL
for (i in 1:4){
  for (j in 1:4){
    for (k in 1:4) {
      for (m in 1:4) {
        kwordseq <- c(kwordseq,paste(i,j,k,m,sep=""))
      }
    }
  }
}
kwordseq[(1:256)[prom.count[,,,]>=10]]
```

- The actually observed word frequencies need to be compared with those obtained via our mathematical model

## Counting k-words in promoter sequences

| Word | Observed Freq | Expected Freq |
|---|---|---|
| "1111" | 14 | 2.841857 |
| "1144" | 12 | 2.841857 |
| "2124" | 10 | 3.029698 |

- R code :

```
kwordseq[(1:256)[prom.count[,,,]>=10]]
prom.count[prom.count[,,,]>=10]
expect4.ec[(1:256)[prom.count[,,,]>=10]]
```

- Are these abundances significant ?
  - So what is the expected number of occurrences of the k-word?
  - How are these numbers distributed?

## Counting k-words in promoter sequences

- Previously, we counted all occurrences of a k-word in the whole set of N regions
- Alternatively, we can count the number of promoter sequences in which the word occurs at least once.
  - Why is it sufficient to look at “at least one”, without further specification? → Only one occurrence at a particular location may be sufficient for functioning
- This will lead to an alternative statistic of which the distribution conforms to the normal approximation of the binomial distribution

## Counting k-words in promoter sequences

- In practice:
  - Simulate 5000 sequences with letter probabilities corresponding to the E *coli* genome
  - Use the simulated data to estimate

$$p_w = \mathbb{P}(w \text{ occurs at least once in a 51-letter sequence}) \approx \frac{\#\text{ of sequences in which } w \text{ appears at least once}}{5000}.$$

- For the number $N_w$ of promoter sequences in which w appears at least once,

$$Z_w = \frac{N_w - 8p_w}{\sqrt{8p_w(1-p_w)}} \sim N(0,1)$$

when there would be 8 trials.

Note that this is in fact too small for the normal approximation to hold, but in other situations it may actually perform pretty well.

# Counting k-words in sequences

```
Ncount4b <- function(seq,w){
  # w is length of word
  tcount <- array(rep(0,4^w),rep(4,w))
  # array[4 times 4 times 4 times 4] to hold word counts, elements set to zero
  ncount <- array(rep(0,4^w),rep(4,w))
  # array[4 times 4 times 4 times 4] holds number of sequences with one or more of
each k-word
  N <- length(seq[1,]) # length of each sequence
  M <- length(seq[,1]) # number of sequences

  # Count total number of word occurrences
  for (j in 1:M){ # loopiing over sequences
    jcount <- array(rep(0,4^w),rep(4,w))
    # array to hold word counts for sequence j
    for (k in 1:(N-w+1)){ # looping over positions
    jcount[seq[j,k],seq[j,k+1],seq[j,k+2],seq[j,k+3]] <-
     jcount[seq[j,k],seq[j,k+1],seq[j,k+2],seq[j,k+3]] +1
     # adds 1 if word at k, k+1, k+2, k+3 appears in sequence j
```

```
}
tcount <- tcount + jcount
# add contribution of j to total

 # plug-in: add 1 to ncount if word occurs W= once in j
  for (k in 1:4){
   for (l in 1:4){
    for (m in 1:4) {
     for (n in 1:4) {
      if (jcount[k,l,m,n]!=0){
      ncount[k,l,m,n] <- ncount[k,l,m,n]+1
      }
     }
    }
   }
 }
 return(tcount,ncount)
}
```

## Counting k-words in sequences

- R code for computation of $p_w$

```
ec.sim <- matrix(nrow=5000,ncol=51)
for (i in 1:5000){
  ec.sim[i,] <- sample(x,51,replace=T,prob.ec)
}
sim.count <- Ncount4b(ec.sim,4)
```

- R code for computation of $N_w$

```
prom.count$ncount[1,1,1,1]
prom.count$ncount[1,1,4,4]
prom.count$ncount[2,1,2,4]
```

## Counting k-words in sequences

| Word | Observed Freq | Expected Freq | $N_w$ | $p_w$ | p-value |
|---|---|---|---|---|---|
| "1111" | 14 | 2.841857 | 7 | 0.1356 | 5.027e-10 |
| "1144" | 12 | 2.841857 | 6 | 0.1474 | 7.627e-07 |
| "2124" | 10 | 3.029698 | 4 | 0.1642 | 5.176e-03 |

**A=1; C=2; G=3; T=4**

- R code for computation of p-values

```
Nw <-
c(prom.count$ncount[1,1,1,1],prom.count$ncount[1,1,4,4],prom.count$ncount[
2,1,2,4])
pw <-
c(sim.count$ncount[1,1,1,1]/5000,sim.count$ncount[1,1,4,4]/5000,sim.count$n
count[2,1,2,4]/5000)
options(digits=4)
Zw <- (Nw-8*pw)/sqrt(8*pw*(1-pw))
1-pnorm(Zw)
```

## References:


- Deonier et al. *Computational Genome Analysis*, 2005, Springer.
  (Chapters 2,3)


## Background reading:


- HT_BioC_manual: http://htseq.ucr.edu/ (part of R BioConductor Manual)
- Morgan et al 2009. ShortRead: a Bioconductor package for input, quality assessment, and exploration of high throughput sequence data. Bioinformatics Advance Access published August 3.
- I/0 and Quality Assessment using ShortRead. R document May 31, 2009
- Bangham 2005. The (computational) means and the motif. Nature Reviews Genetics – Bioinformatics 6:161.


- Key “for your library” reference: Venter et al 2001. The sequence of the human genome. Science 291: 1304-.

# In-class discussion document

- Gregory 2005. Synergy between sequences and size in large-scale genomics. Nature Reviews Genetics 6: 699-.

Questions: In class reading_4.pdf

## Preparatory Reading:

- Bioinformatics explained: BLAST
- Bioinformatics explained: Smith-Waterman
- Bioinformatics explained: BLAST versus Smith-Waterman