BIOINFORMATICS

Kristel Van Steen, PhD^2

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)

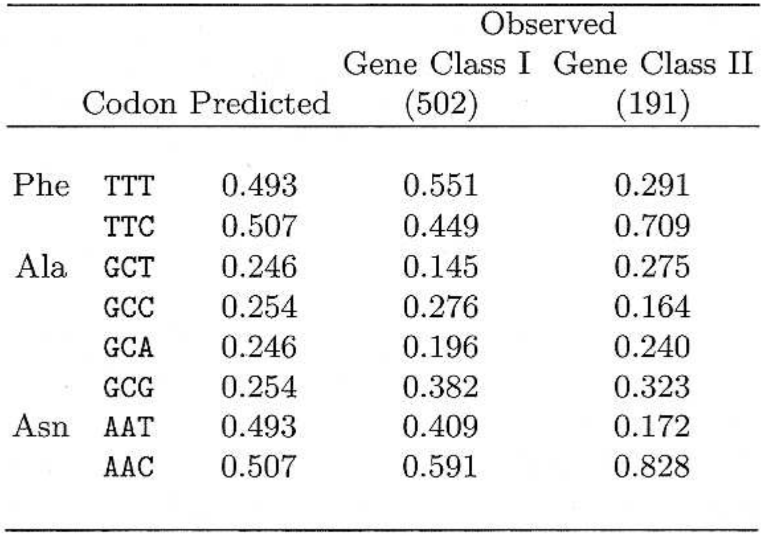
6 Modeling the number of restriction sites in DNA

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.

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



			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

Moderate expr

High expression levels

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

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

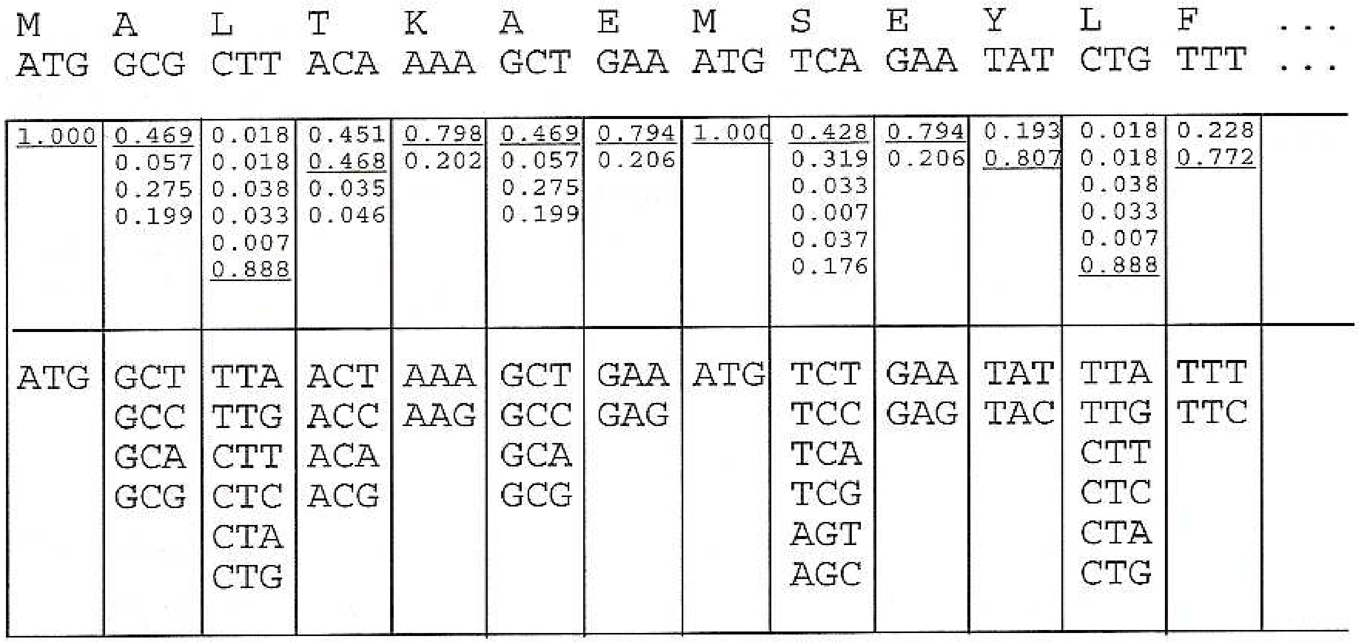
- It is the geometric mean of the ratios of the probabilities for the codons *actually* used to the probabilities of the codons *most frequently* used in highly expressed genes.
- An alternative way of writing this is

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

- 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

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

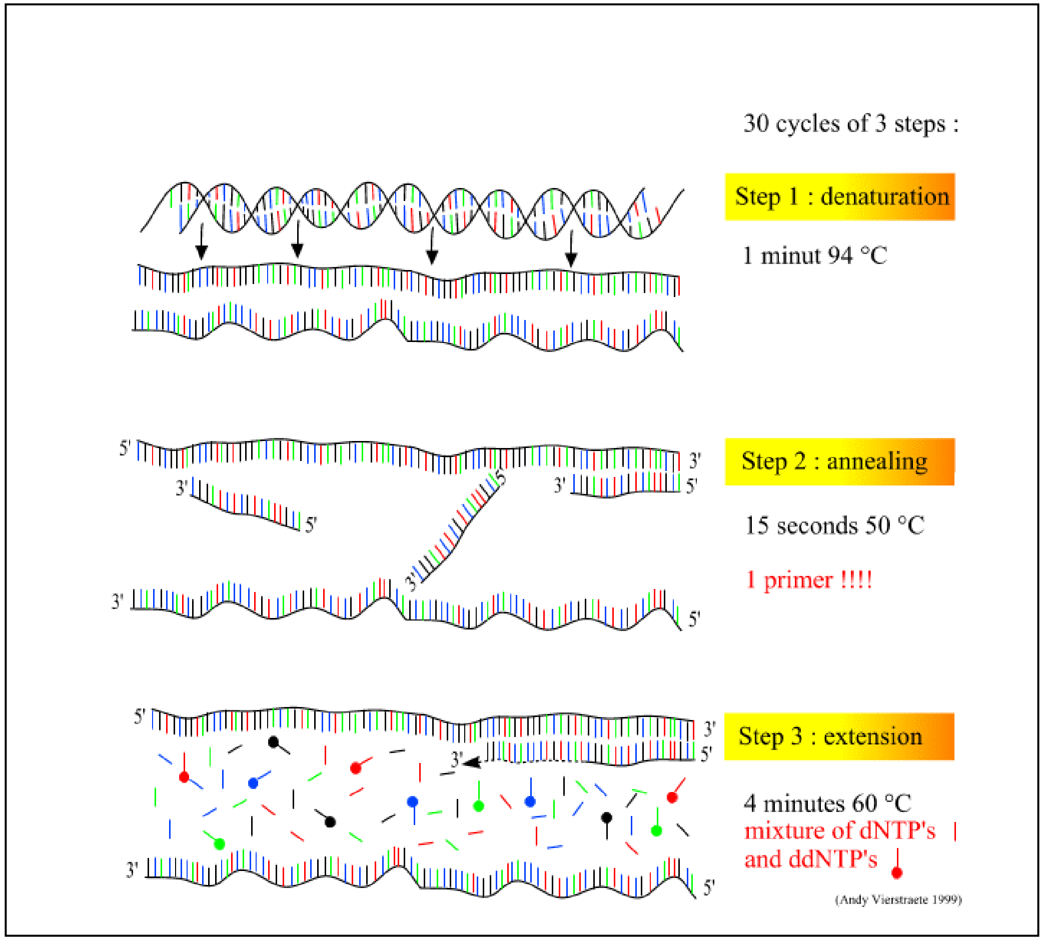


M	A	L	T	K	A	E	M	S	E	Y	L	F	...
ATG	GCG	CTT	ACA	AAA	GCT	GAA	ATG	TCA	GAA	TAT	CTG	TTT	...
1.000	0.469 0.057 0.275 0.199	0.018 0.018 0.038 0.033 0.007 0.888	0.451 0.468 0.035 0.046	0.798 0.202	0.469 0.057 0.275 0.199	0.794 0.206	1.00(	0.428 0.319 0.033 0.007 0.037 0.176	0.794 0.206	0.193 0.807	0.018 0.018 0.038 0.033 0.007 0.888	0.228 0.772	
ATG	GCT GCC GCA GCG	TTA TTG CTT CTC CTA CTG	ACT ACC ACA ACG	AAA AAG	GCT GCC GCA GCG	GAA GAG	ATG	TCT TCC TCA TCG AGT AGC	GAA GAG	TAT TAC	TTA TTG CTT CTC CTA CTG	TTT TTC	

Word distribution and occurrences - The biological problem

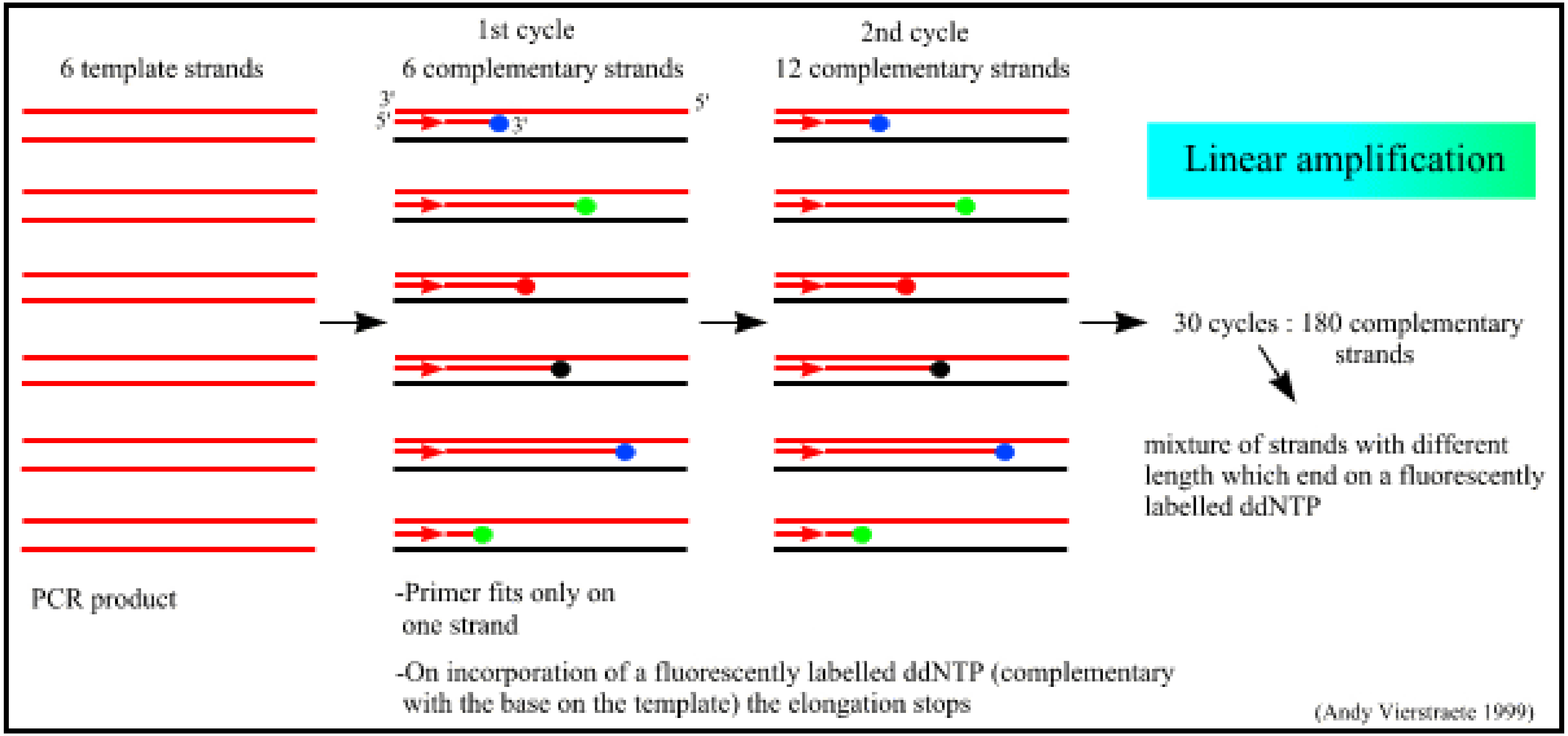
- Suppose that we wanted to obtain a sample of DNA that contained a specific gene or portion of a gene with very little other DNA.
- How could we do this?
  - Today, given a genome sequence, we could design PCR primers flanking the DNA of interest and could amplify just that segment by PCR.

The sequencing reaction



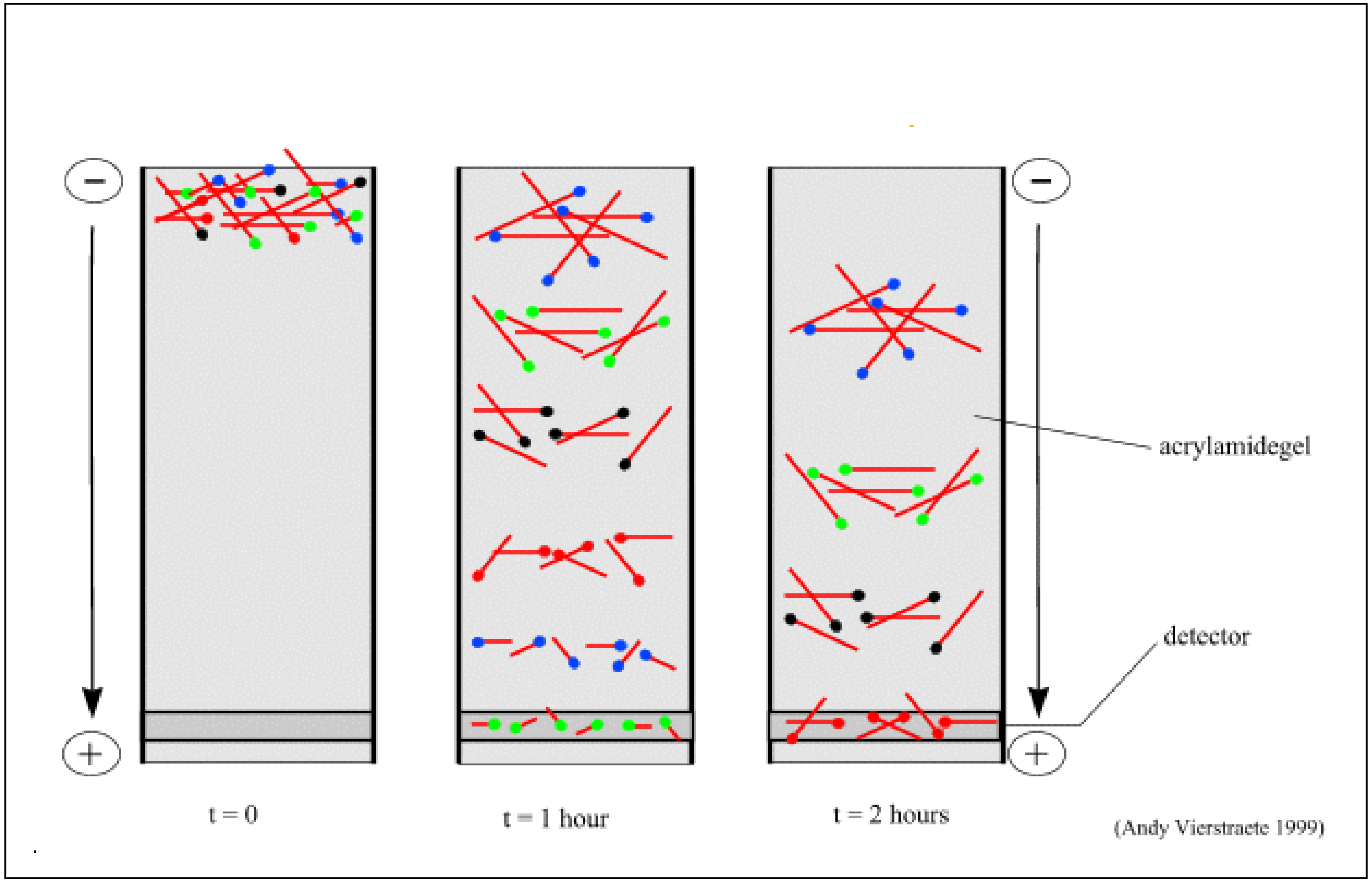
(http://users.ugent.be/~avierstr/principles/seq.html)

The sequencing reaction


1st cycle
2nd cycle
6 template strands
6 complementary strands
12 complementary strands
3'
5'
5'
3'
Linear amplification
30 cycles : 180 complementary strands
mixture of strands with different length which end on a fluorescently labelled ddNTP
PCR product
-Primer fits only on one strand
-On incorporation of a fluorescently labelled ddNTP (complementary with the base on the template) the elongation stops
(Andy Vierstraete 1999)

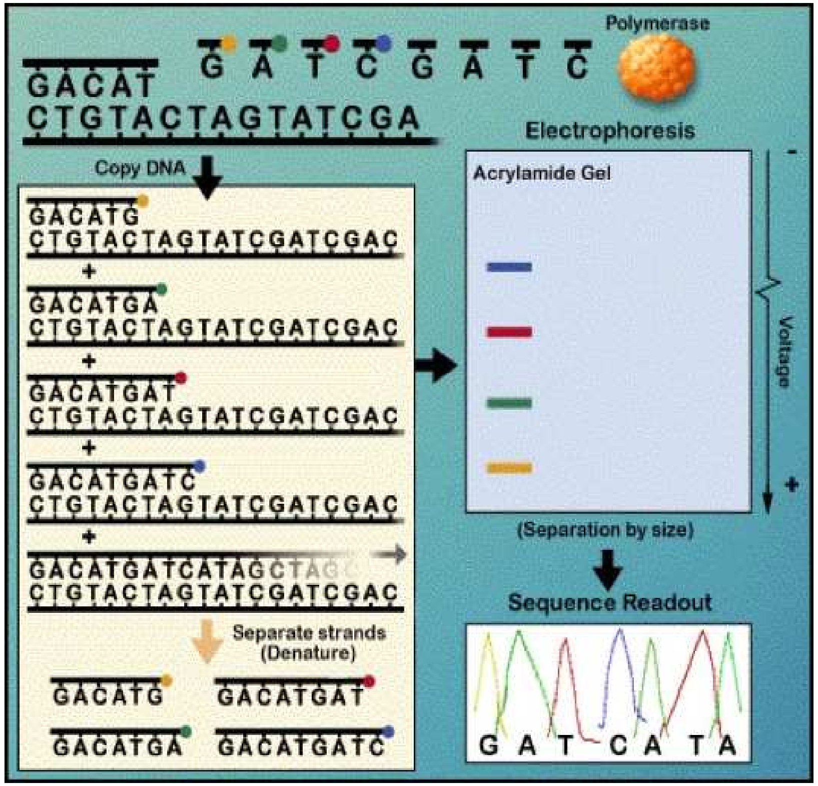
(http://users.ugent.be/~avierstr/principles/seq.html)

Separation of the molecules via gel electrophoresis



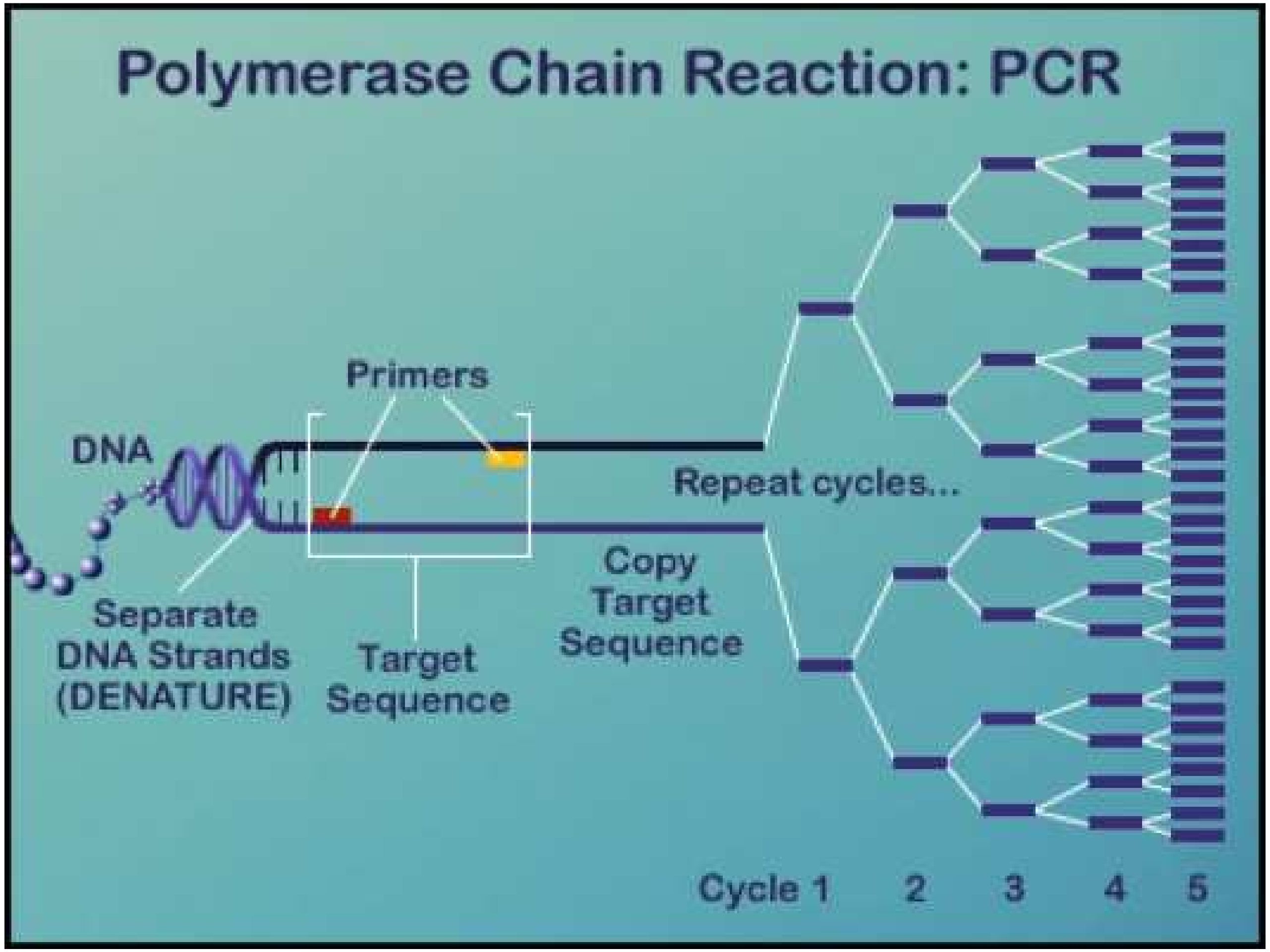
(http://users.ugent.be/~avierstr/principles/seq.html)

Sequencing of DNA



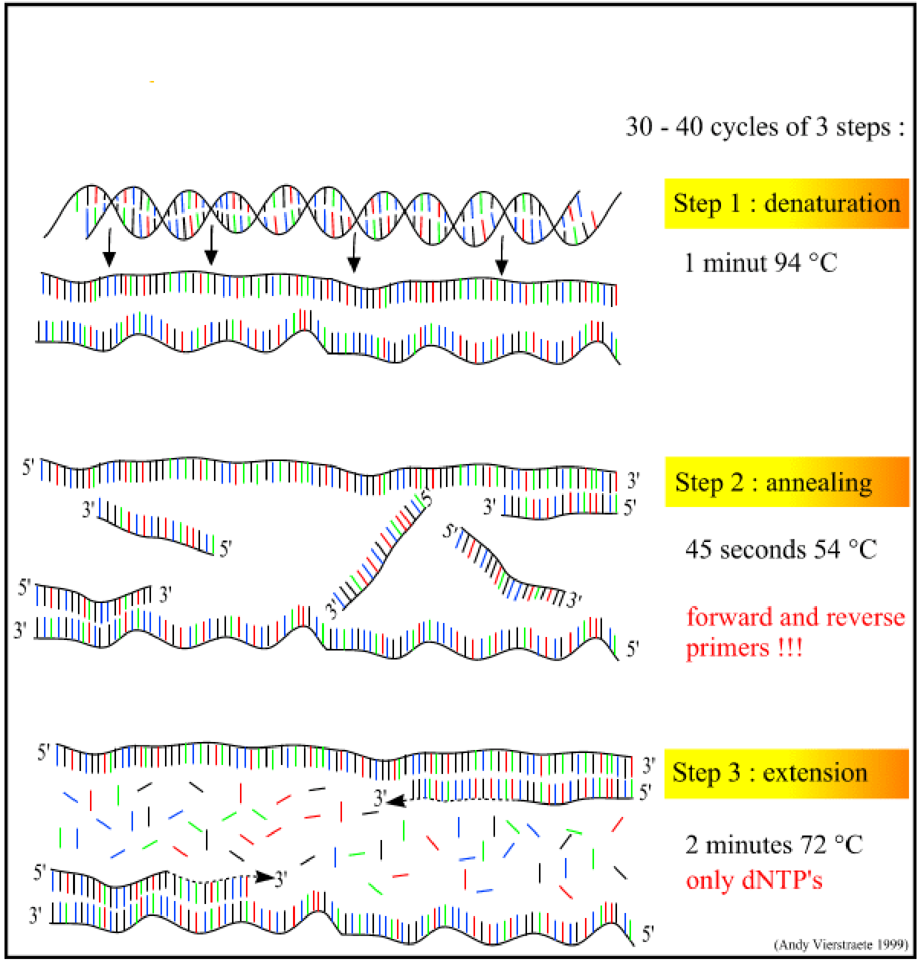
- The result is an electropherogram showing the fluorescence units over time and fragment positions.

Polymerase chain reaction (PCR): In vitro DNA replication



(Roche Genetics)

The cycling reactions of a PCR



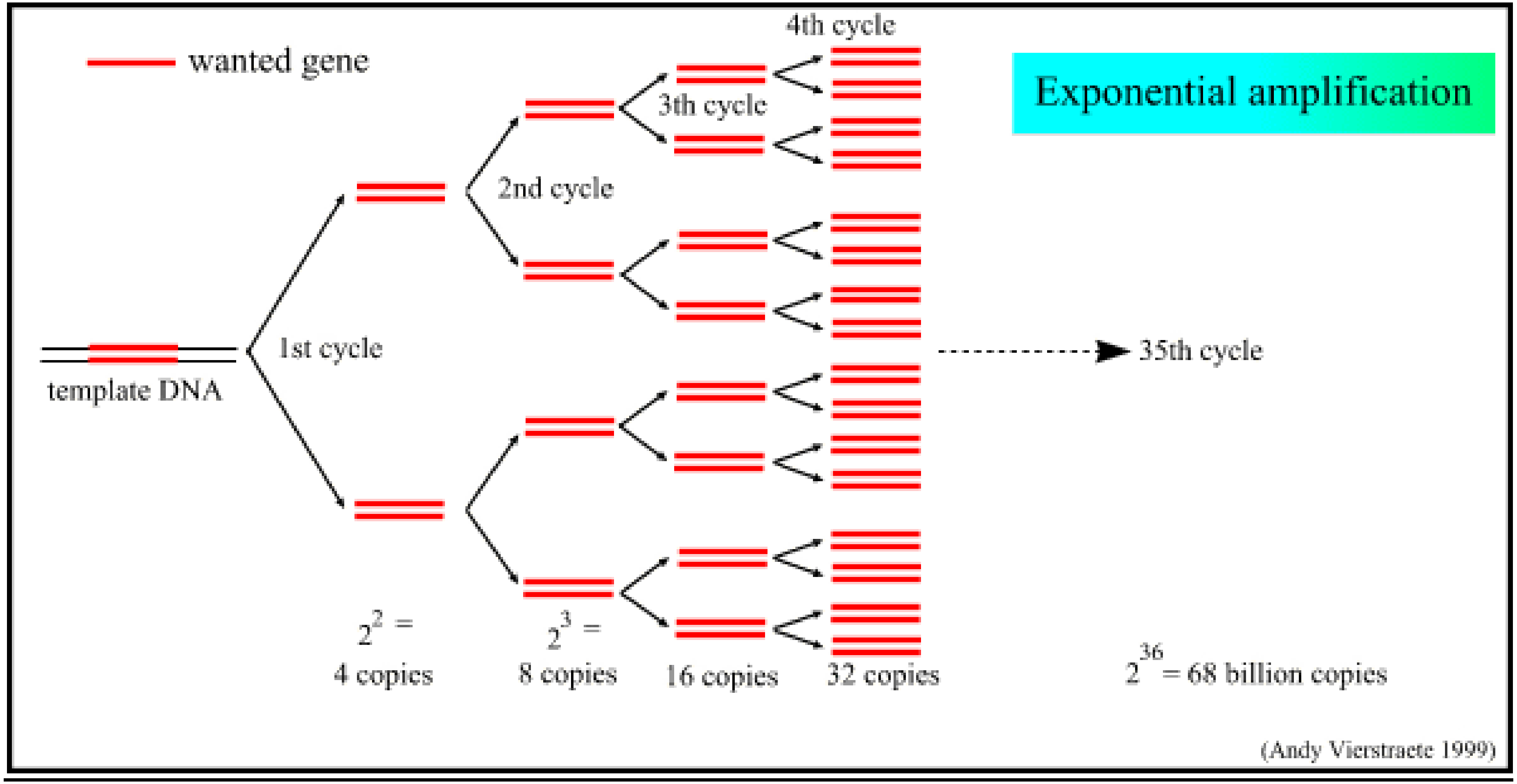
(http://users.ugent.be/~avierstr/principles/pcr.html)

The cycling reactions of a PCR

- Because both strands are copied during PCR, there is an *exponential* increase of the number of copies of the gene.
- Suppose there is only one copy of the wanted gene before the cycling starts,
    - after one cycle, there will be 2 copies,
    - after two cycles, there will be 4 copies,
    - three cycles will result in 8 copies and so on.

(http://users.ugent.be/~avierstr/principles/pcr.html)

The cycling reactions of a PCR



(Andy Vierstraete 1999)

(http://users.ugent.be/~avierstr/principles/pcr.html)

The biological problem

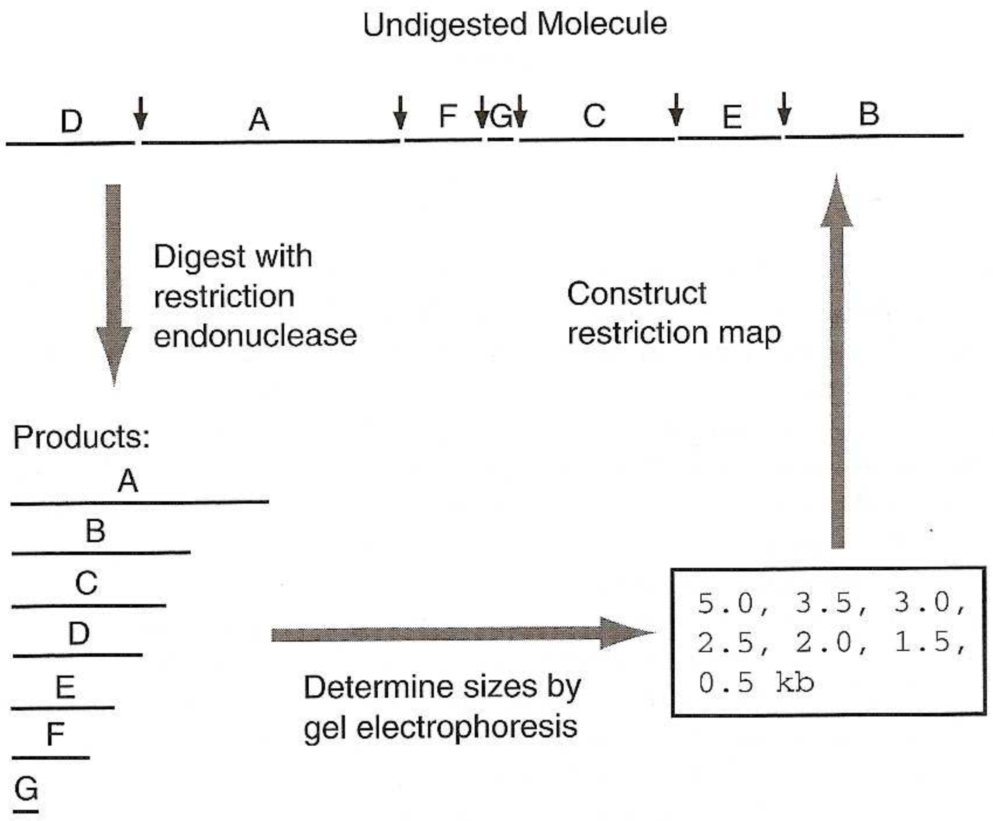
- Prior to the development of rapid genomic sequencing technologies, the process was much more complicated.
- Restriction endonucleases provides the means for precisely and reproducibly cutting the DNA into fragments of manageable size (usually in the size range of 100s to 1000s of base pairs), and
- molecular cloning provides the method for amplifying the DNA of interest

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.


Undigested Molecule
D
A
F
G
C
E
B
Digest with restriction endonuclease
Construct restriction map
Products:
A
B
C
D
E
F
G
Determine sizes by gel electrophoresis
5.0, 3.5, 3.0, 2.5, 2.0, 1.5, 0.5 kb

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, ... , 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, ... , X_n,
- The number of restriction sites is $N = X_1 + X_2 +$ '" $+ X_m$, 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, …, 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 also 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)
```

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

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

- 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):

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

Distribution of restriction fragment lengths

- The previous has some important consequences:

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

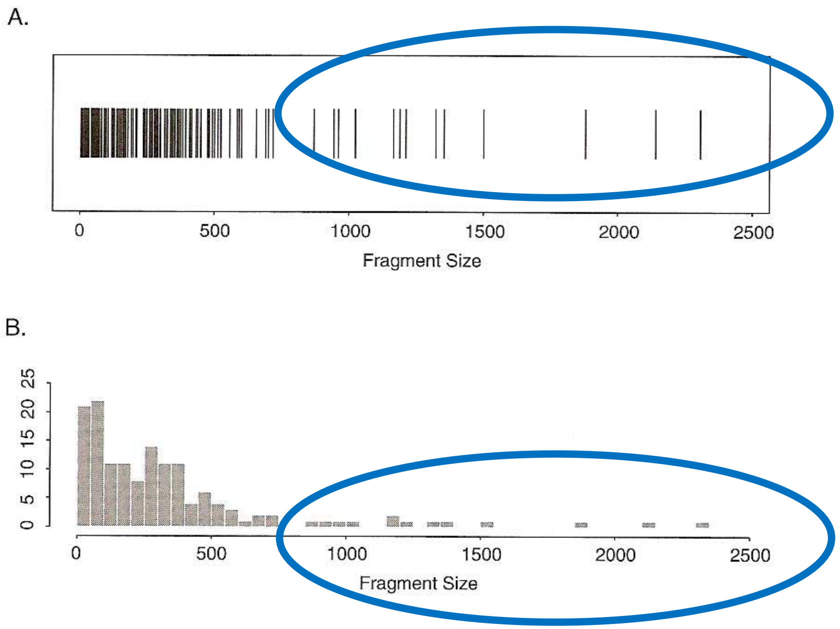
so that the density function for X is given by

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

- The distance between restriction sites therefore follows an exponential distribution with parameter λ
  - 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)?



Actual 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)
}
```

Example

```
inseq <- c(1,1,2,3,4,1,2,4,3,2,1)
seq <- c(1,2)
rsite(inseq,seq)
```

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?

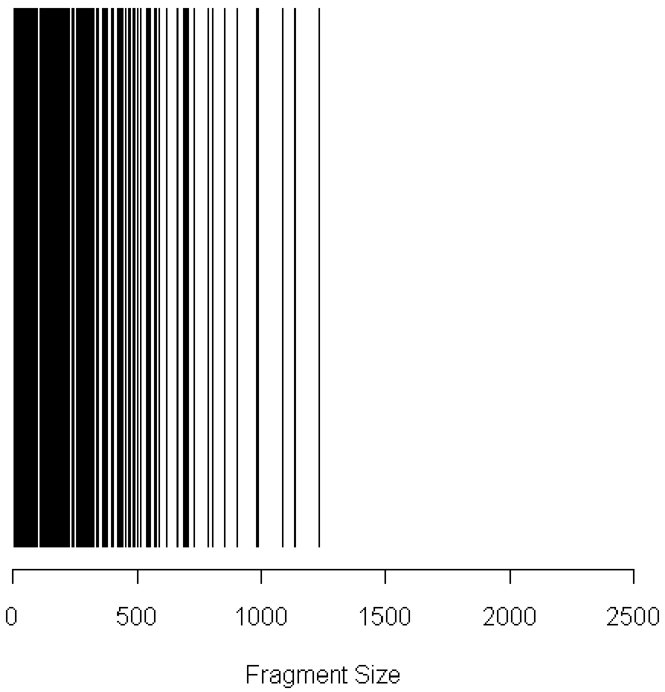
```
max(alu.frag[])
min(alu.frag[])
```

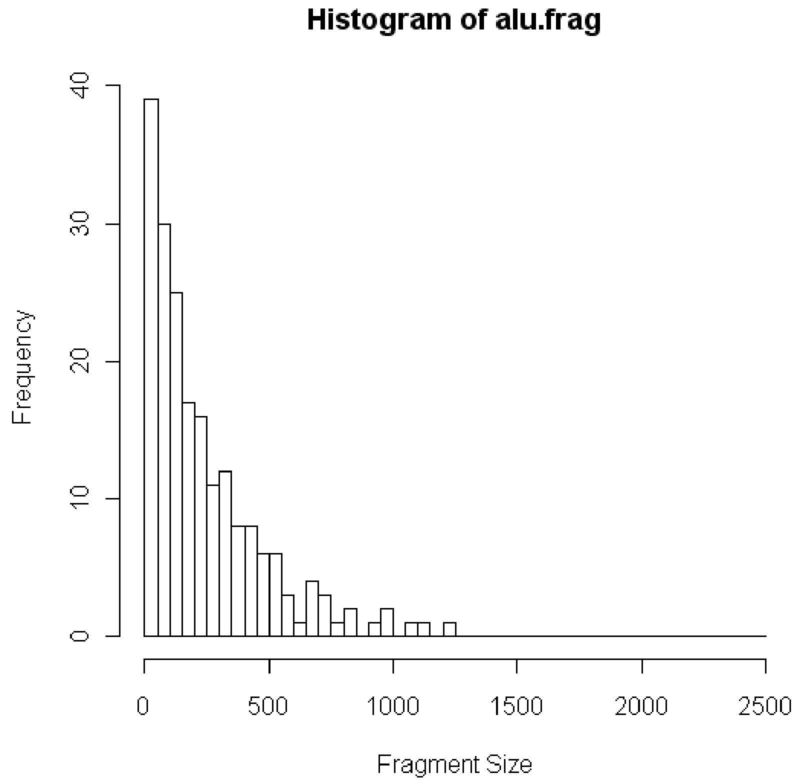
Internal checks

```
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:

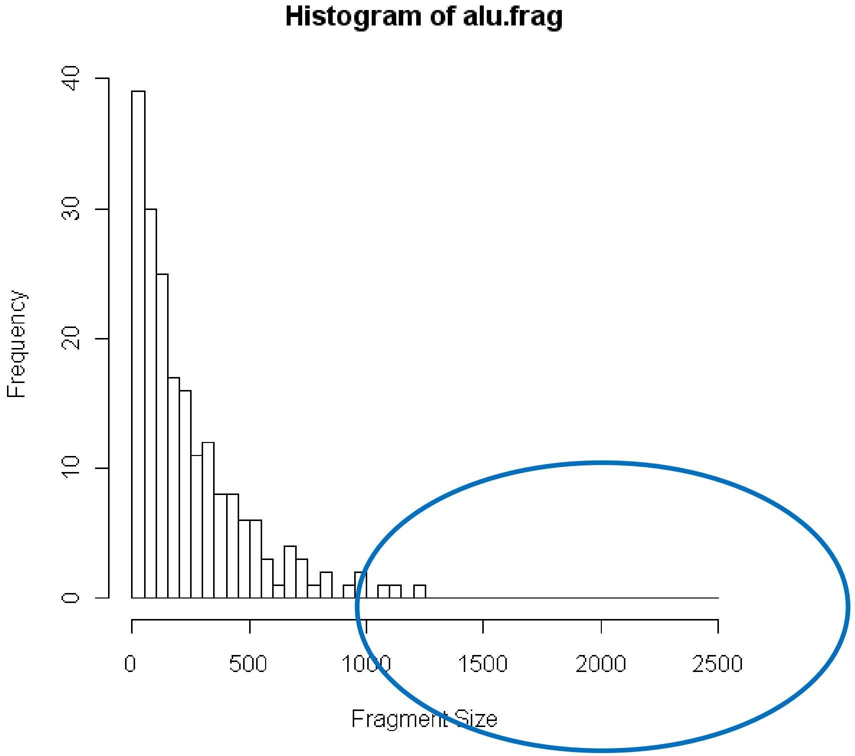
```
plot(c(0,2500),c(3,1),xlab="Fragment Size",ylab="",type="n",axes=F)
axis(1,c(0,500,1000,1500,2000,2500))
for (i in 1:length(alu.frag)){
lines(c(alu.frag[i],alu.frag[i]),c(1,3))
}
hist(alu.frag,breaks=seq(0,2500,50), freq = TRUE,xlab="Fragment Size")
```

- The main important question is:

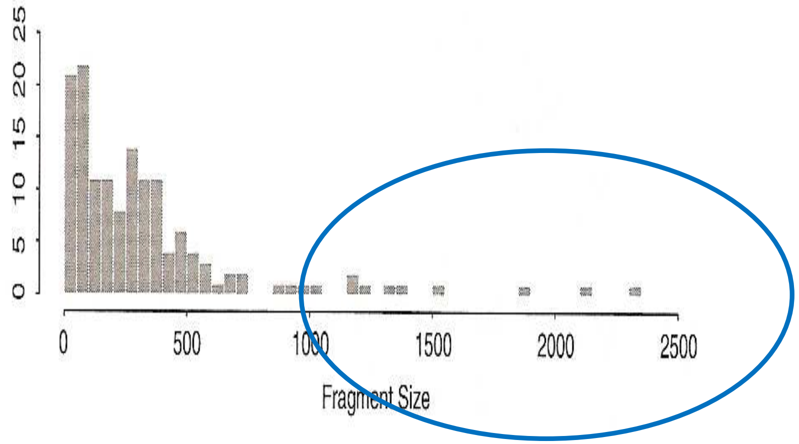
Is our theoretical model still ok when looking at

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.

(http://www.stat.yale.edu/Courses/1997-98/101/chigf.htm)

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 transcriptional 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?

- 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 :

```
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?

Main reference:

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

Background reading: **- useful to better understand theory**

- http://www.stat.yale.edu/Courses/1997-98/101/chigf.htm
- Abdurashitov 2008. A physical map of human Alu repeats cleavage by restriction endonucleases. BMC Genomics 2008, 9:305

In-class discussion document

- Gregory 2005. Synergy between sequences and size in large-scale genomics. Nature Reviews Genetics 6: 699-.
- Key “for your library” reference: Venter et al 2001. The sequence of the human genome. Science 291: 1304-.

Questions:

- In Class reading_Gregory2005.pdf
- In Class reading_Venter2001.pdf