BIOINFORMATICS & COMPUTATIONAL GENETICS MSc
PROBABILITY AND STATISTICS

S-PLUS EXERCISE SHEET 5

1. Starting the Splus session

Double click the S-PLUS icon, or start the program from the **Start** menu. When the program has opened up the *Object Browser* window, start a *Commands Window* by using the **Window -> Commands Window** pull down menus.

2. **GOODNESS OF FIT TESTS FOR SEQUENCE COMPARISON**

We have seen how to compare biological sequences using a hypothesis test, either using a **chi-squared statistic** or a **likelihood ratio** statistic; for the table

	Nucleotide A	C	G	T	Total
Sequence 1	n_{11}	n_{12}	n_{13}	n_{14}	$n_{1.}$
Sequence 2	n_{21}	n_{22}	n_{23}	n_{24}	$n_{2.}$
Total	$n_{.1}$	$n_{.2}$	$n_{.3}$	$n_{.4}$	n

if we wish to test the hypothesis H_0 that the two sequences have the same (marginal) nucleotide probabilities, $p_A = p_1, p_C = p_2, p_G = p_3$ and $p_T = p_4$ we first compute the estimates of these nucleotide probabilities (using **maximum likelihood estimation**) under the assumption that H_0 **is true**;

$$\hat{p}_1 = \frac{n_{.1}}{n} \qquad \hat{p}_2 = \frac{n_{.2}}{n} \qquad \hat{p}_3 = \frac{n_{.3}}{n} \qquad \hat{p}_4 = \frac{n_{.4}}{n}$$

and then compute the **expected** or **fitted values** if H_0 **is true** are given by

$$\hat{n}_{ij} = n_{i.}\hat{p}_j = \frac{n_{i.}n_{.j}}{n} \qquad i = 1, 2,\ j = 1, 2, 3, 4$$

and then the two test statistics that may be used; the first is the **Chi-squared statistic**

$$\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{4} \frac{(n_{ij} - \hat{n}_{ij})^2}{\hat{n}_{ij}}$$

and the second is the **Likelihood Ratio** (**LR**) statistic

$$LR = 2\sum_{i=1}^{2}\sum_{j=1}^{4} n_{ij} \log \frac{n_{ij}}{\hat{n}_{ij}}$$

Recall that both of these statistics have an approximate **Chi-squared distribution** $\chi^2_{(r-1)(c-1)} = \chi^2_3$ distribution, again given that H_0 **is true.** Typically, a significance level of $\alpha = 0.05$ is used for this test, and the critical value in a one-tailed test of the hypothesis is at the 0.95 point of this distribution, that is, at 7.81.

To compute the quantities in S-PLUS proceed as follows for two simulated DNA sequences to be generated; we will assume that the nucleotide frequencies are equal

```
>n1 <- 2000
>n2 <- 2500
>n <- n1+n2
>sequence1 <- sample(c(1:4),n1,prob=c(0.25,0.25,0.25,0.25),rep=T)
>sequence2 <- sample(c(1:4),n2,prob=c(0.25,0.25,0.25,0.25),rep=T)
>obs.table <- matrix(0,nrow=2,ncol=4)
>fit.table <- matrix(0,nrow=2,ncol=4)
>obs.table[1,]_table(sequence1)
>obs.table[2,]_table(sequence2)
>chi.stat <-0
>lr.stat <- 0
>for(i in 1:2){
+ for(j in 1:4){
+ fit.table[i,j] <- (sum(obs.table[i,])*sum(obs.table[,j])/n
+ chi.stat <- chi.stat + ((obs.table[i,j]-fit.table[i,j])^2)/fit.table[i,j]
+ lr.stat <- lr.stat + 2*obs.table[i,j]*log(obs.table[i,j]/fit.table[i,j])
+ }}
>chi.stat
>lr.stat
```

EXERCISE Repeat for different sample lengths, and also by changing the probability vectors in lines 4 and 5, and attempt to get a significant result.

3. SIMULATION-BASED HYPOTHESIS TESTING FOR SEQUENCE ALIGNMENT

We have seen in Exercise 4 how to compute p-values for run-test statistics by simulating i.i.d. *Geometric*(θ) random variables and studying the distribution of the maximum in the sample. We can repeat the exercises in experiments concerned with matching or aligning sequences. We being by proceeding as above, first using a *Binomial* (N, p_{MATCH}) model under H_0 for the number of matches in two sequences of length n:

```
>n <- 5000
>pmatch <- 0.25
>sequence1 <- sample(c(1:4),n,prob=c(0.25,0.25,0.25,0.25),rep=T)
>sequence2 <- sample(c(1:4),n,prob=c(0.25,0.25,0.25,0.25),rep=T)
>b <- length(sequence1[sequence1 == sequence2])
>pvalue <- pbinom(b,n,pmatch)
```

EXERCISE Repeat for different sample lengths, and also by changing the probability vectors in lines 3 and 4, in order to get a significant result.

To allow for the possibility of mismatches in a sequence alignment, in a run-test situation, the *Negative Binomial* probability sampling model needs to be used. In fact Monte Carlo simulation is the only way that a $p-$value may be computed. The calculation may be achieved as follows, similarly to those methods used in Exercise 4. Here we allow $k = 3$ mismatches, with $p = 0.25$

```
>k <- 3
>p <- 0.25
>nits <- 100000
>xmax_rep(0,nits)
>for(i in 1:nits){
+ x <- rngeom(k,p)
+ xmax[i] <- max(x)+k
+ }
>hist(xmax)
```

The histogram produced essentially contains the frequencies with which the maximum run length is equal to a specified value. In line 7, the value k is added to the maximum value due to the way that the Negative Binomial samples are obtained.

EXERCISES (i) Repeat for different sample lengths, and also by changing the probability vectors in lines 3 and 4, in order to get a significant result

(ii) An even more accurate $p-$value can be obtained from repeated simulation of pairs of sequences, with the maximum run lengths being computed by trawling through the sequences looking for runs of consecutive matches allowing up to k mismatches. Repeated simulation of pairs and sequences can be achieved fairly easily

```
>nits <- 10
>n <- 100000
>match.count_rep(0,nits)
>for(i in 1:nits){
+ pmatch <- 0.25
+ sequence1 <- sample(c(1:4),n,prob=c(0.25,0.25,0.25,0.25),rep=T)
+ sequence2 <- sample(c(1:4),n,prob=c(0.25,0.25,0.25,0.25),rep=T)
+ match <- length(sequence1[sequence1 == sequence2])
+ match.count[i] <- sum(match)
+ }
>match.count
```

which reports successive numbers of matches overall. However, computation of the maximal run lengths (with mismatches) for each generated match sequence is not straightforward. Can you construct some code to do this ?