1)

Answers:

A1)

```
CGCA-T
ATCACT
```

A2) score=5

		C	G	C	A	T
	0	-3	-6	-9	-12	-15
A	-3	**-2**	-2	-5	-5	-8
T	-6	-2	**-4**	-1	-4	-1
C	-9	-2	-4	**0**	-3	-3
A	-12	-5	-3	-1	**4**	1
C	-15	-8	-4	3	1	**5**
T	-18	-11	-7	0	1	**5**

A3) It is the minimum number of edit operations- insertions, deletions and substitutions – needed to transform the first sequence into the second.

A4) Compare all pairs of sequences to obtain a similarity matrix; Based on the similarity matrix, make a guide tree relating all the sequences; Perform progressive alignment where the order of the alignments is determined by the guide tree;

A5) Maximum parsimony

---2)--

a. unsupervised learning can be used to:

cluster similar genes together -- which genes have a similar profile. Can then find identity of new genes having a similar function to known genes. e.g. cell cycle experiments -- genes can be clustered together that have peak expreession at similar points in cell cycle. This helps us identify new genes that have similar profiles to known genes.

cluster similar arrays together -- find out which chips cluster together; presumably cells with similar RNA should have similar expression profiles and so cluster together. This can be used as a quality control, to test whether the uarray data is sensible. e.g. clustering data from two types of cancer, would expect the chips from one type of cancer to group together.

b. partioning vs hierachical methids

Both are good at clustering data, but take different approaches to the problem.

partioning methods:
(+) easy to understand the results
(-) need to specify the number of clusters in advance.
(-) Kmeans e.g. can give different results on different runs.

hierachical methods
(+) may be faster than partioning methods.
(+) no need to specify number of clusters, since you can cut the tree at any level
(-) divisive tree-methods are better than agglomerative methods, but harder to implement. Agglomerative methods are greedy, thus may make bad clusters.
(-) many ways of drawing the same tree, which may lead to interpretation errors.

Both techniques enforce some structure onto the data, when there may be none (e.g. each gene will be clustered into one of K clusters, regardless of whether it is really appropriate.

When deciding which method to use, first off, these methods are quick enough that you can just run both methods and compare results. However, sometimes you may know that you want a predefined number of clusters, in which case, you could use a partioning method with the given number of K.

c. [see lecture notes for pseudo code].

d. The silhoutte widths method is a good example of a way to evaluate the number of clusters. see lecture notes for definition of the method. We try a range of values of K, and chose the value that maximises silhoutte width.

If the number of clusters is too few, a_i will typically be quite large, since the variance within each cluster will be high; this leads to small silhoutte width.

If the number of clusters if too large, a_i will be quite small, but b_i will also be quite small since there is likely to be another cluster close to the one being examined. Hence if b_i is small, the silhoutte width will be small.

3)

Model answers for Systems Biology question

(a) $\frac{\partial p(x)}{\partial t}=\lambda p(x)-\lambda p(x-1)+\beta(x+2)^2 p(x+2)-\beta x^2 p(x)$

(b) Motivation: The first step simulates the time for the next event. The system above leaves state x with total rate $\lambda_{tot}=\lambda+\beta x^2$ which is a contact for given x. The probability P that the even has not yet occurred at time t is thus governed by $\partial P/\partial t=-\lambda_{tot}P$., which means that the probability that the reaction has occurred is $F(t)=1-P=1-e^{\lambda_{tot}t}$. The waiting time is thus exponentially distributed. Given that some reaction occurred, the second step is to simulate which one it was. The probability for each event is simply the rate of that event divided by the total rate, e.g. $\beta x^2/\lambda_{tot}$ for the second event. The algorithm itself should be of this type (% signifies a comment):

```
For i: sim                            % Loop over number of simulations
   rates = [ $\lambda$ $\beta x^2$ ]                    % Vector of rates
   cumr=cumsum(rates)                 % Cumulative sum, cumr(2) = $\lambda_{tot}$.
   u_event=rand                       % Uniform random number on [0,1]
   tsim(i+1)=-log(1-u)/cumr(2)        % Exp distr. number for reaction time
   if cumr(1)/cumr(2) >u_event
      x (i+1) =x(i)+1
   else
      x (i+1) =x(i)-1
   end_if
end_for
```

(c) $\frac{\partial\langle x\rangle}{\partial t}=\lambda-2\beta\langle x^2\rangle\approx\lambda-2\beta\langle x\rangle^2$

(d) The FDT is accurate for nonlinear processes as long as the fluctuations are not so large that the responses are strongly nonlinear. It is thus exact for linear processes regardless of the size of the fluctuations, and close to exact for strongly nonlinear processes as long as the fluctuations are very small.

(e) $\langle x\rangle=\sqrt{\frac{\lambda}{2\beta}}$ and $H=\frac{\partial\ln\left(2\beta\langle x\rangle^2/\lambda\right)}{\ln\langle x\rangle}=2$

$$\tau=\frac{\langle x\rangle}{2\beta\langle x\rangle^2}=\frac{1}{2\beta\langle x\rangle}=\frac{1}{\sqrt{2\beta\lambda}} \text{ and } \langle r\rangle=\frac{1\times\lambda+2\times2\beta\langle x\rangle^2}{\lambda+2\beta\langle x\rangle^2}=\frac{\lambda+2\lambda}{2\lambda}=\frac{3}{2}$$

(f) Stationary normalized FDT: $M\eta+(M\eta)^T=D$ where $M=H\tau^{-1}$ and $D=2\tau^{-1}\langle r\rangle/\langle x\rangle$. In this scalar case, we get $\eta=\frac{D}{2M}=\frac{3}{4}\frac{1}{\langle x\rangle}=\frac{3}{4}\sqrt{\frac{2\beta}{\lambda}}$. If instead $x\xrightarrow{\beta x^2}x-2$ changes to $x\xrightarrow{\beta x}x-1$, we get $\eta=\frac{D}{2M}=\frac{1}{\langle x\rangle}$. In the original example, the quadratic nonlinearity corrects fluctuations more effectively and the increased event size increases fluctuations. The first effect is stronger, resulting in a net suppression of fluctuations.

(g) Observation (1) means that the noise does not come from having a low number of x_2 molecules. Element D_{22} can thus be set to zero, which simplifies the solution of the FDT. The first step is to calculate H_{22}, the only nonlinear part of the equation:

$$H_{22}=\frac{\partial\ln\left[\beta_2x_2/\left(\lambda_2x_1x_2^{-h}\right)\right]}{\partial\ln x_2}=1+h$$

This gives