Biology Background

Introduction

The application of computer science to other disciplines seems to be increasing at an exponential rate. Each day that passes sees new ways to apply the ideas of computer science to problems in all walks of life, and especially the “hard” sciences. The fields of computational biology and bioinformatics are of particular interest here at Hunter because we have a bioinformatics/quantitative biology concentration in computer science. There are many examples of simple problems in these areas in which to demonstrate programming concepts, but to do so, the student needs to know a bit about the underlying biology. The purpose of these notes is to provide that “bit of knowledge.” They are written for a prospective computer scientist rather than a biologist.

Background

A ***DNA string***, also called a ***DNA strand,*** is a finite sequence consisting of the four letters `A`, `C`, `G`, and `T` in any order[1]. The four letters stand for the four ***nucleotides***: ***adenine***, ***cytosine***, ***guanine***, and ***thymine***. Nucleotides, which are the molecular units from which DNA and RNA are composed, are also called *bases*. Each nucleotide has a ***complement*** among the four: `A` and `T` are complements, and `C` and `G` are complements. Complements are chemically-related in that when they are close to each other, they form ***hydrogen bonds*** between them.

A special enzyme called ***RNA polymerase*** uses the information in DNA to create RNA. The process of creating RNA from DNA is called ***transcription***. A ***RNA string*** or ***RNA strand*** is a finite sequence consisting of the four lowercase letters `A`, `C`, `G`, and `U`. The `A`, `C`, and `G` have the same names as they do in DNA, but the `U` represents ***uracil***. When DNA is transcribed to RNA by RNA polymerase, each thymine base is converted to uracil. Hence RNA strings have `U`’s wherever DNA has `T`’s.

RNA in turn serves as a template for the construction of ***proteins***, which are sequences of ***amino acids***. Proteins are synthesized within the ***ribosomes*** of living cells by a process called ***translation***. In translation, the RNA string is viewed as a sequence of three-letter groups called ***codons***. Each codon codes for a particular amino acid. For example, `GUU` codes for *valine*, and `UCA` codes for *cysteine*. Let us count how many possible three-letter sequences are there in which each letter can be `A`, `C`, `G`, or `T`. There are four choices for the first letter, four independent choices for the second letter, and four for the third, so there are $4^3 = 64$ different codons. On the other hand, there are only 20 different amino acids. Some amino acids are coded for by multiple codons. For example, `UCA`, `UCC`, `UCG`, and `UCU` all code for *cysteine*. Some codons do not code for any amino acids; they are *stop codes*. There are three stop codons: `UAA`, `UAG`, and `UGA`.

Stop codes are used during protein synthesis to terminate reading of the RNA string. Not all of a RNA string is translated into protein; there are large regions that act like gaps. As the RNA is read, when a gap is reached, it is skipped over until a special start codon is found that tells the ribosome to begin creating amino acids again. When it sees a stop codon it stops and keeps reading until it finds another start codon, and so on, until the entire strand is read. Sequences of RNA that are removed like this are called ***introns***. Introns also refer to the corresponding regions of the original DNA. The regions that do get used in protein synthesis are called ***exons***. Figure 1 depicts the relationship between introns and exons. Much is yet to be discovered about introns; they were once thought to be useless parts of the RNA, but now it is known that this is not true.

[1] Some sources use lowercase while others use uppercase. Here they will be used interchangeably

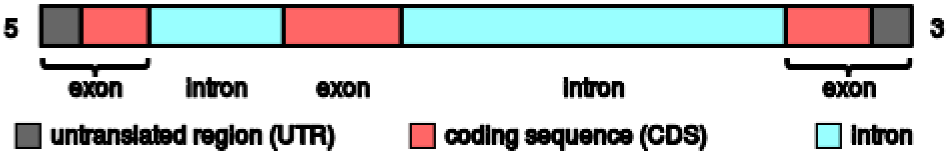

Figure 1: Illustration of introns and exons.

The process of removing introns is just like the way comments in shell scripts are treated by the shell. The `#` tells the shell to stop parsing the command, and the next newline character tells it to start again. The `#` is a stop code and the newline is a start code. As an example, the RNA strand

```
AUGGUUUAUGGUCUCUGA
```

is read as the following sequence of codons

```
AUG GUU UAU GGU CUC UGA
```

Consulting a table of these mappings, we see that `AUG` is a start codon that codes for *methionine* (Met), `GUU`, for *valine* (Val), `UAU`, for *tyrosine* (Tyr), `GGU`, for *glycine* (Gly), `CUC`, for *leucine* (Leu), and `UGA` is a stop codon. Therefore, the sequence *Met-Val-Tyr-Gly-Leu* is created from this RNA fragment.

Amino acids have long names like cysteine but they also have three-letter names such as `Cys`, for *cysteine*, and one-letter uppercase names, such as `C` for *cysteine*. It is not always true that the one-letter name is the first letter of the amino acid's long name. The above sequence would be written *MVYGL* using the one-letter names.

Direction and Shape

In its most common form, DNA is actually a double helix consisting of two strands that wrap around each other. Each DNA strand has ***direction***. Direction is usually indicated by putting a 5' at one end and a 3' at the other. The 5' and 3' refer to the names of the carbon atoms to which these ends attach. The carbons are part of a ring structure with multiple carbon atoms, so they get names for the purpose of distinguishing them from each other. For example,

```
5'-GTATCC-3'
```

is a fragment of DNA that runs from the 5' to the 3' position. The 5' end is called the ***upstream*** end, and the 3' end is the ***downstream*** end. The two strands of nucleotides are in reverse directions of each other. In other words, if you could unwind the helix so that the two strands were lying on a flat surface parallel to each other, in one strand the 5' end would be to the left, and in the other, it would be to the right. The two strands are chemically-related because the bases that would be across from each other on the table are complements of each other. For example, the two strands below

```
5'-G T A T C C A A T G C C-3'
   | | | | | | | | | | | |
3'-C A T A G G T T A C G G-5'
```

could be a fragment of the unwound double helix. The vertical lines connect complements in the forward and reverse strands to each other. Each `C` in one is matched by a `G` in the other, and each `A` is matched by a `T` in the other.

Characteristic Properties of DNA

Scientists use various heuristic rules in their study of DNA. Among the many metrics that they use are the following:

- A ***poly-T sequence*** of length N is a sequence of N or more consecutive `T` nucleotides.
- The ***GC content*** of a DNA strand is the ratio of the total number of `C` and `G` nucleotides to the length of the strand. For example, the sequence '`atcgtttgga`' is of length 10 and has a total of 4 `C`'s and `G`'s, so its GC content is 0.4.
- A ***CpG island*** is a `C` followed by a `G` in a DNA strand. (The *p* in between the `C` and `G` represents the fact that a *phosphodiester* bond connects `them`.)

Restriction Enzymes

Bacteria produce special enzymes called ***restriction enzymes*** that can cut DNA at specified sites, called ***cleavage sites***. Some theorize that these enzymes evolved to provide a defense mechanism against invading viruses. Inside a bacterial host, the restriction enzymes selectively cut up foreign DNA in a process called ***restriction***; the host DNA is protected from the restriction enzyme's activity.

The cleavage site is a position between two nucleotides in the DNA. The enzyme finds its site by a type of biological pattern-matching. These enzymes usually cut both strands of the DNA, but for simplicity we describe how the cut works on a single strand. The pattern specifies where in the DNA the enzyme will match. For example, the enzyme *EcoRI* has a recognition site defined by

```
5'-G'AATTC-3'
```

This means that it will search for a substring of the DNA consisting of the bases `GAATTC` in the 5' to 3' direction, and cut the DNA between the `G` and the first `A`. The apostrophe ' indicates the cleavage site. So, if the DNA string is

```
ATGAAAGGGTTTCCCTTTGAATTCCCCATGGTATTGTTGCCGGAATTCTTTCCGGCCCCC
```

it will be cut into the three pieces

```
ATGAAAGGGTTTCCCTTTG    AATTCCCCATGGTATTGTTGCCGG    AATTCTTTCCGGCCCCC
```

by *EcoRI*. The restriction enzyme *NotI* is defined by

```
5'-GC'GGCCGC-3'
```

which indicates that it will find all occurrences of the string `GCGGCCGC` in the 5' to 3' direction and cut the DNA between the first `C` and the second `G`.

You may have noticed that if you form the complement of `GAATTC`, you get `CTTAAG`, which is the string spelled backwards. Similarly, the complement of `GCGGCCGC` is `CGCCGGCG`, which is also the string spelled backwards. Certain types of restriction enzymes have this *complement palindromic* property.

There are four different types of restriction enzymes, differing in chemical ways and in the nature of their target sequence and the position of their DNA cleavage site relative to the target sequence. Some restriction enzymes have a cleavage site outside of the recognition site. *AceIII* is defined by

```
CAGCTCNNNNNNN'
```

The N matches any of A, C, G, or T. Therefore, this enzyme cuts the DNA between the 7th and 8th nucleotides after its recognition site. For example

```
CAGCTCAAATGCCAGGGGGGG
```

will be cut between the A and the G:

```
CAGCTCAAATGCCA GGGGGGG
```

Some types of enzymes cut the DNA very far away from the recognition site, on the order of hundreds of nucleotides away. Some of these restriction enzymes will have more than one recognition sequence, and inversely, there are different enzymes that have the same sequence and cut in the same place.

A restriction enzyme can be described in a format known as the *Staden* format. The form is

```
enzyme_acronym/recognition_sequence/recognition_sequence/.../recognition_sequence//
```

Most enzymes have a single recognition sequence. Some have two. The cut point will be denoted by an apostrophe in the recognition sequence. Some enzymes from an actual Staden enzyme file are:

```
AatI/AGG'CCT//
AatII/GACGT'C//
AbsI/CC'TCGAGG//
AccI/GT'MKAC//
AccII/CG'CG//
AccIII/T'CCGGA//
Acc16I/TGC'GCA//
BbvI/GCAGCNNNNNNNN'/'NNNNNNNNNNNNGCTGC//
BglI/GCCNNNN'NGGC//
BglII/A'GATCT//
BinI/GGATCNNNN'/'NNNNNGATCC//
```

The first line is the enzyme named *AatI* and its cut point is between the second G and the first C. You will notice that in the fourth line, there are letters other than A, C, G, and T in the recognition sequences. These letters are part of a standard set of abbreviations defined as follows:

```
R = G or A
Y = C or T
M = A or C
K = G or T
S = G or C
W = A or T
B = not A (C or G or T)
D = not C (A or G or T)
H = not G (A or C or T)
V = not T (A or C or G)
N = A or C or G or T
```

The letters act like simple patterns for matching DNA. For example, GTMKAC matches GTAGAC, GTCGAC, GTATAC, and GTCTAC since the M matches an A or a C and the K matches a G or a T. There are therefore four possible combinations that the two together can match. The enzyme *BbvI* has two recognition sites, which are reverse palindromic.