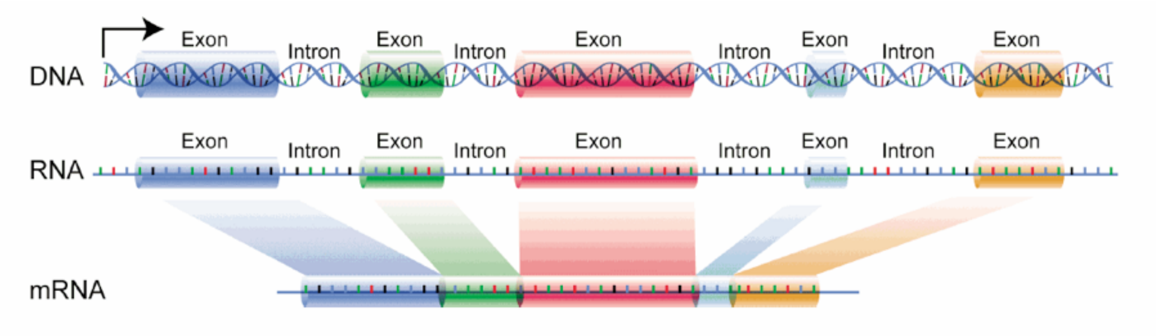
Lab 3: Using the FFT to find DNA periodicities

DSP can be applied in genetic research area. This lab, you will learn the structure of a gene and implement FFT to find introns and exons of a gene.

1. **Background**

All organisms **transcribe and translate** their DNA. This means that when a protein is needed, DNA is the code that will provide the instruction on how to make the protein. All that needs to be done is to transcribe the DNA into mRNA (then the mRNA goes on to form amino acids and the amino acids fold into proteins). In prokaryotes (primitive lifeforms without a nucleus), this is simple because the whole sequence gets transcribed. In eukaryotes (or organisms that have a nucleus in their cells), there are portions that get transcribed (**exons**) and portions that don't get transcribed (**introns**). It is thought that this exon-intron structure enables eukaryotes to turn on a part of gene and to have finer control than prokaryotes (remember ... prokaryotes transcribe have one contiguous exon and transcribe the whole thing). See the structure below:



An interesting fact is that exons (also known as **coding-regions**), have a periodic structure whereas introns (also known as **noncoding-regions**), do not. Therefore, many researchers have explored using the Fourier transform to detect these regions (such as **Akhtar M., Epps J., Ambikairajah E.** "*Signal Processing in Sequence Analysis: Advances in Eukaryotic Gene Prediction*" IEEE Selected Topics in Signal Processing, 2008.) In this lab, you will explore to use the FFT at the $\frac{N}{3}$ position to detect the protein-coding portions of a gene from the non protein-coding portions.

The National Center for Biotechnology Institute's Genbank is the US's major repository for all DNA. Unfortunately, using Genbank, is a lab completely in itself. But we do want you to get acquainted to be able to retrieve a sequence.

We will be studying the Human Hemoglobin sequence in class. Please see:
`http://ghr.nlm.nih.gov/gene/HBB`

Lab 3.1.1:

Go to Genbank's homepage (`http://www.ncbi.nlm.nih.gov`):
1. Enter **hbb** into the search - how many hits did you get?
2. Click on the nucleotide link. To the right (in the Top Organisms box), they suggest the human, mouse, chimpanzee, etc. HBB genes
3. Type **hbb AND human[orgn]** into the search. Now how many hits do you get? Note: the **AND** has to be capitalized. The **human[orgn]** means that you're narrowing your search just to humans.
4. Click on the **HBB (Homo sapiens): hemoglobin, beta** in the Gene search (you can also use the link: `http://www.ncbi.nlm.nih.gov/gene/3043`). Explore this record.
5. Search for **hbb AND human[orgn] AND RefSeqGene** in the Nucleotide search. (this should lead you to `http://www.ncbi.nlm.nih.gov/nuccore/28380636`)
6. We will actually be obtaining the whole region and doing analysis on the 81,706 bp linear DNA.

Lab 3.1.2:

FFT review: linspace, Fs, fftshift, NFFT:

1. Generate one second of a cosine of $\omega_s = 10Hz$ sampled at $F_s = 100Hz$ and assign it to x. Define a tt as your time axis.
2. Take 64 points FFT.
3. As you remember, the DFT (which the FFT implements) computes N samples of $\Omega_k = 2\pi\frac{k}{N}$, where $k = 0, 1, 2, 3, ..., N-1$. Plot the magnitude of this 64-points FFT at range 0 to 63, what do you think of this graph?
4. To get the x-axis into a Hz-frequency form, plot this 64-points FFT between -50 to 50 (the $100Hz$ sampling rate) and have N-points between them.
5. According to your figure, what frequency is this cosine wave at?
6. Remember that the FFT is evaluating from 0 to 2π. We are used to viewing graphs from $-\pi$ to π. Therefore, you need to shift your graph.
7. Now according to your shifted graph. what frequency is this at?
8. Note that the spikes have long drop-offs? Try a 1024-point DFT. Note that the peak is closer to 10 and the drop-off is quicker. Although, now sidelobes are an issue.

2. **Warm-up** Show your work to TA by the end of this lab

Ex 3.2.1: The following is an example of loading and manipulating the file you downloaded from NCBI:

```
% example of load and manipulate *.gb file in Matlab:
% The file you download from NCBI should be named as 'hbb_region_chr11.gb'
% Matlab can load *.gb as a struct
% Double click on the struct in your Worksapce to see its fields
% These fields have a wide variety of types (cell, struct, char, etc..)
hbb = genbankread('hbb_region_chr11.gb'); % load *.gb file as a struct
CDS = hbb.CDS; % extract CDS field (Coding Sequence from hbb) as a struct
CDSrange1 = CDS.indices; % get the range indices of CDS
CDSrange2 = hbb.CDS.indices; % a quicker way to get the range
A=hbb.Sequence(1); % extract the first base in this sequence
```

Lab 3.2.1:

1. How many CDS regions are there in this sequence? (A CDS sequence is a coding sequence that results in a protein product).
2. Look at **hbb.CDS(1)**, what is the length of this region? How would you identify the coding and non-coding sequences for this **hbb.CDS(1)** region on Chromosome 11?
3. Write a function that will take a sequence and CDS indices as input and outputs the corresponding coding and non-coding DNA sequences.

Hint:

The instructor will verify that it is correct by entering a random CDS region's indices into your function.

After we generate the coding sequence, we want to take the FFT of it. Given DNA is a sequence of letters, we have take the binary indicator representation of the sequence. Then, the FFT of the sequence is the summation of the FFT of each binary indicator representation of the sequence, i.e.,

$$coding\_FT[k] = |FFT(coding\_A)| + |FFT(coding\_T)| + |FFT(coding\_C)| + |FFT(coding\_G)|$$

This is similar to the equation in **Anastassiou.** "*Genomic Signal Processing*" IEEE Signal Processing Magazine, 2001., except that we do not take the power spectrum like they did (we just take the magnitude). This is Anastassiou's equation, where $U\_A[k]$ represents the DTFT of $u\_A[k]$, which is the binary indicator sequence of A:

$$S[k] = |U_A[k]|^2 + |U_T[k]|^2 + |U_G[k]|^2 + |U_C[k]|^2$$

Ex 3.2.2: This example converts a DNA sequence, called coding, into four separate binary indicator sequences for each nucleotide and takes FFT of sequence:

```
coding = hbb.Sequence(1:999); % extract 999 bases in this sequence
coding_A = (upper(coding)=='A'); % find A bases and set them to 1
coding_T = (upper(coding)=='T'); % find T bases and set them to 1
coding_G = (upper(coding)=='G'); % find G bases and set them to 1
coding_C = (upper(coding)=='C'); % find C bases and set them to 1
coding_FT = abs(fft(coding_A,1024)).^2+abs(fft(coding_T,1024)).^2 ...
    +abs(fft(coding_G,1024)).^2+abs(fft(coding_C,1024)).^2; % FFT of the sequence
```

Once you have computed your *coding_FT*$[k]$ sequence, we can now plot it. Plot it without x-axis labels. At first, it may seem like the first value (or DC value) dwarfs the other values, but zoom in to the rest of the sequence and find the next highest peaks - What are their values? You should actually find that the next-to-highest peaks have x-axis values that are close to $\frac{NFFT}{3}$ and $\frac{2\times NFFT}{3}$. This is not a coincidence! Due to biological reasons, there is a distinct 3-base periodicity in many coding regions. Do you know why there is also a peak at $\frac{2\times NFFT}{3}$? The reason that the peaks are located at NFFT/3 is because 3 is the periodicity and the DFT frequency is therefore $\frac{NFFT}{3}$.

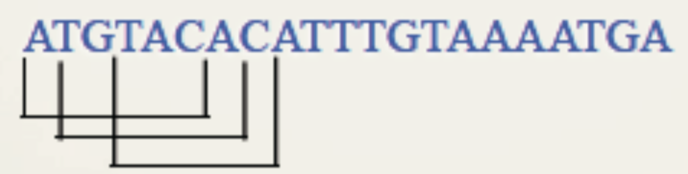
Lab 3.2.3:
Now, compute the magnitude spectrum of the noncoding sequence (use the same NFFT value as above)? what does it look like? What is the magnitude of the NFFT/3 point compared to the coding sequence? Show your two plots, explain the peaks to your instructor, and the difference between coding and noncoding regions.

3. **Exons detection** Formal report for this section due by the start of next lab
In the previous sections, we took the FFT of whole sequences. One may start to wonder if we could actually take smaller sequences (or windows) and predict a coding from a non-coding region as you go along the sequence. To do such a thing, we need the short-time discrete fourier transform:

$$STFT\{x[n]\} \equiv X(m,\omega) = \sum_{n=0}^{\infty} x[n]\omega[n-m]e^{-j\omega n}$$

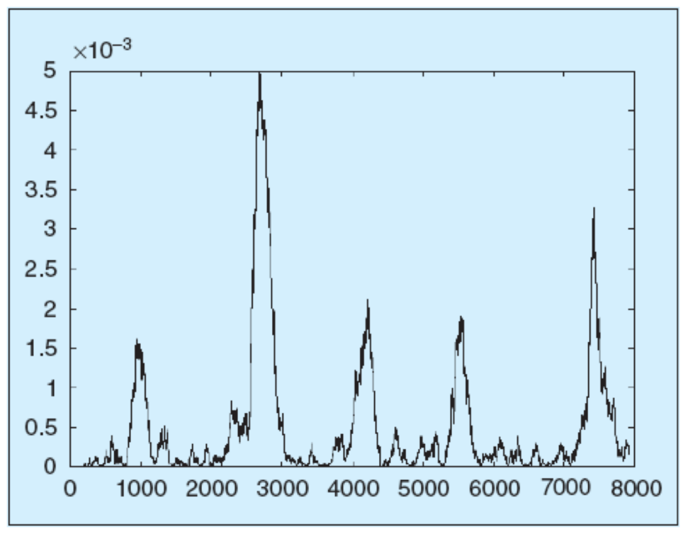
where $\omega = \frac{2\pi k}{N}$
when the FFT is used and m is the **WINDOW_LENGTH**. In this lab, we assume that all values of $\omega[n] = 1$ (so this is a square window). Say that $S = STFT\{x[n]\}$, then $S[\frac{N}{3}]$ will be the three base periodicity magnitude in the window. We want to write a function that will output $S[\frac{N}{3}]$ for all consecutive windows in the sequence.

Lab 3.3.1:
1. Make a function called **threebasefreq_stft.m** that can be called like this:
Threebaseperiodicity_vs_position = threebasefreq_stft (DNA_SEQUENCE, WINDOW_LENGTH, NFFT)
2. After you implement your function, test it on the whole 81,706 bp sequence of the HBB gene. Show two plots of the results (similar to the figure in *Hint 5*) by using a) **threebasefreq_stft(seq,100,1024)** and b) **threebasefreq_stft(seq,1000,1024)**.
3. Compare and contrast your results. Include all Matlab codes in your report.
Hint:
1. Assume that the windows have full overlap meaning that each window of **WINDOW_LENGTH** overlaps by **WINDOW_LENGTH** − 1 data points:



2. **DNA_SEQUENCE** is the DNA sequence of letters (before the binary indicator operation).
3. **NFFT** is the number of points to take in the Fourier transform.
4. The output will give you the magnitude of the $\frac{N}{3}$ point for each position (or consecutive window) in the sequence.

5. We show an example of using such a program on a sequence that has both exons and introns (**Anastassiou.** "*Genomic Signal Processing*" IEEE Signal Processing Magazine, 2001.). The following figure shows the $\frac{NFFT}{3}$ of the Spectrum (note this graph is for the magnitude squared) vs. position in the sequence. Also note that spectrums of each binary indicator sequence were specially weighted to get better results. Your results may vary. The results here show that the peaks correspond to exons in ATPase gene in C. Elegans (a worm):


×10⁻³
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
1000
2000
3000
4000
5000
6000
7000
8000