



Taller 3

Unidad 3

Librerías especializadas para Bioingeniería

1. Objetivo

Afianzar las habilidades básicas en el uso de librerías de Python especializadas en el manejo, procesamiento y visualización de información.

2. Para tener en cuenta

Los ejercicios propuestos en el presente documento serán desarrollados por el monitor. A partir de lo visto en el taller, los estudiantes deberán realizar el entregable que se encuentra al final del documento, este deberá ser enviado a más tardar el viernes de la semana siguiente a las 24:00. El archivo con la solución se debe adjuntar por medio de la sección Recurso Tarea que cada taller tiene asignado, con el siguiente asunto: *NumeroGrupo_Nombre*, el archivo *.py* debe estar rotulado con los apellidos de los integrantes.

Nota: tenga en cuenta que este entregable tiene un valor del **4%** del total del curso.

3. Ejercicios:

Uso de las librerías Numpy, Matplotlib y Scipy:

3.1. Realice un algoritmo que cargue una señal (en formato mat) y extraiga la información que contiene el archivo. Obtenga características de la señal como lo son las dimensiones y número de muestras. Por último, grafique la señal para observar su forma a través del tiempo.

3.2. Realice un algoritmo que permita hacer el análisis de Welch a uno de los canales de la señal del numeral anterior. Dicho análisis consiste en los siguientes pasos:

3.2.1. Dividir la señal en segmentos que se pueden solapar.

3.2.2. Cada segmento que se obtiene se multiplica por un segmento conocido como ventana de Hamming.

3.2.3. Sobre cada segmento se calcula la transformada rápida de Fourier. A esta transformada se le extrae la magnitud y se eleva al cuadrado. Cada termino resultante se divide sobre la cantidad de puntos que hay en el segmento (Graficar).

3.2.4. Se promedian los segmentos que se obtienen del paso anterior (Graficar).

Uso de la librería OpenCV:

3.3. Realice un algoritmo que permita la segmentación de los tejidos hiperintensos e hipointensos a una imagen médica. Siga estos pasos:



3.3.1. Crear una máscara para los valores de tejido hiperintensos. Considere que todo tejido es hiperintenso si los valores de los píxeles son superiores a 125.

3.3.2. Crear una máscara para los valores de tejido hipointensos. Considere que todo tejido es hipointenso si los valores de los píxeles son inferiores a 125.

3.3.3. Con las máscaras obtenidas en los puntos anteriores, realice la segmentación. La segmentación se consigue multiplicando la máscara por la imagen original, eliminando así los píxeles que no cumplen las condiciones de interés (hipointenso e hiperintenso).

4. Entregable:

Realice un módulo en Python que permita el tratamiento básico de señales. Para hacerlo tenga en cuenta lo siguiente:

4.1. Un objeto que contenga toda la información relevante de una señal, como lo son las dimensiones (número de canales, épocas y puntos), frecuencia de muestreo y la señal. Dicha señal es cargada desde un archivo con extensión mat.

4.2. Un método que permita redimensionar, o extraer partes de la señal. Especifique por medio de argumentos los canales y épocas que se deseen utilizar.

4.3. Un método que permita calcular algunas variables estadísticas de la señal, tales como media, desviación estándar, valor máximo y valor mínimo. Especifique por medio de argumentos los canales y épocas que se desean analizar.

4.4. Un método que permita aplicar el análisis espectral de Welch, especificando por medio de argumentos los canales y épocas que se deseen analizar. Mostrando al final el gráfico de potencia VS. frecuencias.

4.5. Un método que permita graficar cualquier segmento de la señal, utilizando etiquetas en los ejes (tiempo vs amplitud), un título que contenga el canal y épocas graficadas. Es necesario especificar por medio de argumentos el canal o época que se desee graficar.

4.6. Realizar un código donde se muestra en funcionamiento cada uno de los métodos del objeto que contiene el módulo.