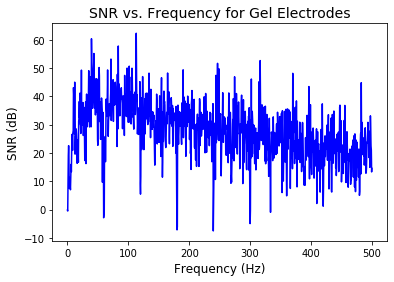
%reset  
%matplotlib inline  
import matplotlib.pyplot as plt  
  
#IPython is what you are using now to run the notebook  
import IPython  
print ("IPython version: %6.6s (need at least 6.1.0)" % IPython.\_\_version\_\_)  
  
# Numpy is a library for working with Arrays  
import numpy as np  
print ("Numpy version: %6.6s (need at least 1.13.1)" % np.\_\_version\_\_)  
  
# SciKit Learn implements several Machine Learning algorithms  
import sklearn  
print ("Scikit-Learn version: %6.6s (need at least 0.19.0)" % sklearn.\_\_version\_\_)

Once deleted, variables cannot be recovered. Proceed (y/[n])? y  
  
  
IPython version: 7.8.0 (need at least 6.1.0)  
Numpy version: 1.17.2 (need at least 1.13.1)  
Scikit-Learn version: 0.22.1 (need at least 0.19.0)

# Set font sizes  
SMALL\_SIZE = 10  
MEDIUM\_SIZE = 12  
BIGGER\_SIZE = 14  
plt.rc('font', size=SMALL\_SIZE) # controls default text sizes  
plt.rc('axes', titlesize=SMALL\_SIZE) # fontsize of the axes title  
plt.rc('axes', labelsize=MEDIUM\_SIZE) # fontsize of the x and y labels  
plt.rc('xtick', labelsize=SMALL\_SIZE) # fontsize of the tick labels  
plt.rc('ytick', labelsize=SMALL\_SIZE) # fontsize of the tick labels  
plt.rc('legend', fontsize=SMALL\_SIZE) # legend fontsize  
plt.rc('figure', titlesize=BIGGER\_SIZE) # fontsize of the figure title

# import data  
gel\_relaxed1 = np.loadtxt('./csv/gel\_relaxed1.csv',delimiter=',',usecols=(0,1,3,4),skiprows=21)  
gel\_relaxed2 = np.loadtxt('./csv/gel\_relaxed2.csv',delimiter=',',usecols=(0,1,3,4),skiprows=21)  
gel\_clenched1 = np.loadtxt('./csv/gel\_clenched1.csv',delimiter=',',usecols=(0,1,3,4),skiprows=21)  
gel\_clenched2 = np.loadtxt('./csv/gel\_clenched2.csv',delimiter=',',usecols=(0,1,3,4),skiprows=21)  
silver\_relaxed = np.loadtxt('./csv/silver\_relaxed.csv',delimiter=',',usecols=(0,1,3,4),skiprows=21)  
silver\_clenched\_BAD = np.loadtxt('./csv/silver\_clenched\_BAD.csv',delimiter=',',usecols=(0,1,3,4),skiprows=21)  
silver\_clenched\_better = np.loadtxt('./csv/silver\_clenched\_better.csv',delimiter=',',usecols=(0,1,3,4),skiprows=21)

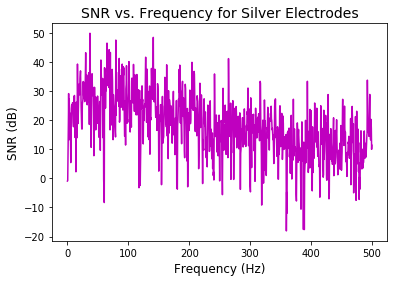
# Calculate SNR for gel  
SNR\_gel1 = gel\_clenched1[:,3] - gel\_relaxed1[:,3]  
SNR\_gel2 = gel\_clenched2[:,3] - gel\_relaxed2[:,3]  
  
# plot SNR data  
plt.figure(1)  
plt.title("SNR vs. Frequency for Gel Electrodes", fontsize=14)  
plt.xlabel("Frequency (Hz)")  
plt.ylabel("SNR (dB)")  
plt.plot(gel\_clenched2[:2000,2],SNR\_gel2[:2000], 'b-')  
plt.savefig('SNR\_Gel.pdf')



# Calculate the mean SNR from 0-500 Hz for gel  
  
gel\_mean\_500Hz\_1 = np.mean(SNR\_gel1[:2000])  
gel\_mean\_500Hz\_2 = np.mean(SNR\_gel2[:2000])  
gel\_mean\_500Hz = np.mean([gel\_mean\_500Hz\_1, gel\_mean\_500Hz\_2])  
  
gel\_std\_500Hz\_1 = np.std(SNR\_gel1[:2000])  
gel\_std\_500Hz\_2 = np.std(SNR\_gel2[:2000])  
gel\_std\_500Hz = np.mean([gel\_std\_500Hz\_1, gel\_std\_500Hz\_2])  
  
print('Mean SNR from 0-500 Hz for gel electrodes: ' + str(gel\_mean\_500Hz) + ' dB')  
print('Standard deviation of SNR from 0-500 Hz for gel electrodes: ' + str(gel\_std\_500Hz) + ' dB')

Mean SNR from 0-500 Hz for gel electrodes: 24.053194375 dB  
Standard deviation of SNR from 0-500 Hz for gel electrodes: 9.85431908596908 dB

# Calculate SNR for silver  
SNR\_silver = silver\_clenched\_better[:,3] - silver\_relaxed[:,3]  
  
# plot SNR data  
plt.figure(1)  
plt.title("SNR vs. Frequency for Silver Electrodes", fontsize=14)  
plt.xlabel("Frequency (Hz)")  
plt.ylabel("SNR (dB)")  
plt.plot(silver\_relaxed[:2000,2],SNR\_silver[:2000], 'm-')  
plt.savefig('SNR\_Silver.pdf')



# Calculate the mean SNR from 0-500 Hz for silver  
  
silver\_mean\_500Hz = np.mean(SNR\_silver[:2000])  
silver\_std\_500Hz = np.std(SNR\_silver[:2000])  
  
print('Mean SNR from 0-500 Hz for silver electrodes: ' + str(silver\_mean\_500Hz) + ' dB')  
print('Standard deviation of SNR from 0-500 Hz for silver electrodes: ' + str(silver\_std\_500Hz) + ' dB')

Mean SNR from 0-500 Hz for silver electrodes: 17.87521615 dB  
Standard deviation of SNR from 0-500 Hz for silver electrodes: 9.952967824917057 dB