HW 4

Background

The main goal of this assignment is to enable you to model protein folding processes (that we went over in class). We will be using two forms of dimensionality reduction to model protein folding trajectories where we will use (1) second order correlations and (2) higher order statistical correlations to model such fluctuations. In the process, we will also learn a bit about identifying which parts of a protein (particularly amino acid residues) exhibit interesting fluctuations.

The maximum number of points for this assignment is 100. You have a week to submit from Sun (Oct 27, 2019) – which means your assignment is due on Nov 3, 2019 (11.59 PM Central Time).

What to expect?

You will be expected to set up your own Python environment and provide documentation on whether you were successful. This assignment can be run on your laptop. There is no need for a special Google Collab environment. Note that as part of your home work; much of this can be done with less than 40 lines of code.

What to hand in?

You are expected to hand in a python notebook (usually with the extension: `.ipynb`) or a Python script that generates all of the results for the questions below. Within your Python notebook, you can easily document your results, visualize plots, and also add comments using Markdown (`https://en.wikipedia.org/wiki/Markdown`). If you are not familiar with Markdown, you can easily learn it from the link above.

Characterizing the statistical properties of atomistic fluctuations [10 points]

You have seen that proteins have some statistical diversity in their atomistic fluctuations. Your task as part of this section is to characterize the statistical diversity. You will use two forms of representations: (1) Cartesian coordinates (x, y, z) of individual atomic positions and (2) Dihedral coordinates (ϕ, ψ, χ) of the backbone torsion angles. You have access to three `numpy` compressed arrays: for Cartesian coordinates, you will download `Fs-peptide-cartesian-fit.npz` and for dihedral coordinates, you will download `Fs-peptide-dihedral.npz` and `Fs-peptide-dihedral-transform.npz`. In addition to this, you will also download a secondary file: `Fs-peptide-RMSD.npz`.

All of the data that you need for this assignment are in these four arrays. You will find that the Cartesian coordinates will have a shape of $(3, 21, 280000)$ – corresponding to the $3 \times N_a \times N_f$, where N_a corresponds to the number of atoms in Fs-peptide, and N_f corresponds to the number of frames; dihedral coordinates – $(42, 280000)$ and $(84, 280000)$, for the `Fs-peptide-dihedral.npz` and `Fs-peptide-dihedral-transform.npz` respectively; and RMSD – $(280000, 1)$.

Question 1: Characterizing the statistical diversity of atomic fluctuations? [10 points]

One way to characterize the statistical diversity of atomic fluctuations is to use histograms and understand the primary statistical characteristics of the data. To achieve this, we need to do something quite simple. Compute the mean of all the rows of the Cartesian coordinates and subtract it from each column. This gives you the deviations across the entire trajectory.

Now plot a histogram of the deviations that you have computed. Once you plot the histogram, you should compute the mean (μ), standard deviations (σ), and kurtosis (κ) values of the distributions you see. For computing κ, you can use the `scipy.stats` library `kurtosis` function (`https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kurtosis.html`).

Question 2: Finding regions of Fs-peptide that exhibit super-Gaussian characteristics [15 points]

This section requires you to now characterize the deviations that you computed from Question 1 grouped by residues. Recall that the deviations that you computed also has a shape $3 \times N_a \times N_f$. This question has three sub-questions:

- What would be the dimensions of the κ values for the data that you have? [5 points]
- How many atoms in the data are super-Gaussian in all the three (x, y, z) directions, and how many of them are sub-Gaussian in all the three directions (x, y, z)? [5 points]
- Do you find any atoms that have purely Gaussian behaviors? What proportion of the data exhibits higher order statistics? [5 points]

Question 3: What to do with dihedral coordinates? [15 points]

The dihedral coordinates represent a new challenge; how would you characterize the statistical nature of these deviations? For an idea, let's first think of a way in which you can examine the values. Find the (min, max) of the dihedral array: `Fs-peptide-dihedral.npz`. You should do a spot check to see if the values are distributed between -π and +π (-180 and +180 degrees).

This requires you to think a bit harder about working with circularly distributed data. Can you come up with suggestions on how we can convert the data into an array where the data is not distributed between $\pm\pi$, but rather, you can use the same functions (that you used in the previous question) to calculate (μ, σ, κ)? Note that there are no right answers/ wrong answers here; just come up with a reasonable suggestion for how you'd overcome this problem with circular statistics. **[5 points]**

There are many ways in which we can transform our data such that you can use our regular statistical approaches. One way to think about this is to say instead of working directly with (ϕ, ψ) values, we will instead use the following transformation: $[\sin(\phi(\vec{\theta})), \cos(\phi(\vec{\theta})), \sin(\psi(\vec{\theta})), \cos(\psi(\vec{\theta}))]$, where $\vec{\theta}$ is the individual angles you have in the dihedral array. For this part of the question, you will be using the `Fs-peptide-dihedral-transform.npz`.

- What would be the dimensions of the κ values for the data that you have? [2 points]
- How many dihedral angles in the data are super-Gaussian, and how many of them are sub-Gaussian? [4 points]
- Do you find any dihedrals that have purely Gaussian behaviors? What proportion of the data exhibits higher order statistics? [4 points]

Question 4: Using second-order statistics to characterize atomic fluctuations [30 points]

One of the ways you can understand the ways in which atomic fluctuations are coupled to each other is through the use of principal component analysis (PCA). PCA removes dominant second-order spatial correlations by trying to reduce a larger dimensional dataset of correlated variables into small number of transformed uncorrelated variables. While considering the analysis of MD simulation data, PCA provides visualization of residues that are harmonically resolved.

This section requires you to remove spatial dependencies by using the supplied PCA (called SD2) function which takes as input: the data matrix based on (i) cartesian coordinates coordinates of size $3N \times T$, where N refers to the number of residues, T indicates the number of snapshots of MD trajectory and m corresponds to the dimensionality of the subspace or (ii) dihedral coordinates of size $4N \times T$, where the four coordinates for each residue at a particular time-stamp is obtained from the angular transformation of angles ϕ and ψ obtained from Q3. Download the SD2 module here.

Following is a *Python* code snippet which essentially performs three things: SD2 implementation, generate cumulative variances based on eigenvalues and eigenvectors from PCA, and obtain second-order spatially resolved matrix by projecting the input data onto the whitening matrix obtained after doing PCA analysis on m components. The SD2 function returns four parameters: a matrix of spatially uncorrelated components Y of size $m \times T$, eigenvalues of the data covariance matrix S, eigenvectors of the data covariance matrix B, and the whitening matrix U.

```python
import SD2
import numpy as np
import matplotlib.pyplot as plt

# perform SD2
Y,S,B,U = SD2.SD2(X.T,m=X.shape[0])

# obtain cumulative variances
[pcas,pcab] = np.linalg.eig(np.cov(X))
si = np.argsort(-pcas.ravel())
pcaTmp = pcas
cumvar = np.cumsum(pcaTmp.ravel()/np.sum(pcaTmp.ravel())) * 100
```

```
# a 3D plot of the projections obtained from SD2 modes
Y_sd2 = np.asarray(Y)

fig = plt.figure()
ax3D = fig.add_subplot(111, projection='3d')
p3d = ax3D.scatter(Y_sd2[0,::], Y_sd2[1,::], Y_sd2[2,::], s=30, c=RMSD,
   marker='o')

ax3D.set_xlabel(r'$\bf{SD2_1}$');
ax3D.set_ylabel(r'$\bf{SD2_2}$');
ax3D.set_zlabel(r'$\bf{SD2_3}$');
cbar = fig.colorbar(p3d, ax=ax3D)
```

Based on the results of your implementation by referring to the code block above, please answer the following questions. (**Note**: While making pretty plots please make sure to include axis labels.)

- In order to generate a scree plot, plot the variable cumvar. What can you infer from the cumulative variance plot? Once you get the cumulative variance plots, the next part of this is to project the data onto the eigenvectors.

  Recall that in the class when we went over PCA, you get the matrices: $X = U\lambda V^T$, where X is the data matrix. The projection matrix is given by Y_sd2 [5 points]

- Based on the scatter plot $p3d$ what can you gather from the arrangement of the $3D$ projections of the modes and can you comment if that is sufficient to characterize the conformational landscape of the MD simulated molecular data? For answering this question, you will have to plot the RMSD values on the projection matrix using the 3D scatter plot. Read the documentation of `scatter()` in the matplotlib library (`https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.axes.Axes.scatter.html?highlight=scatter#matplotlib.axes.Axes.scatter` and use this listing as your guideline. [15 points]

- After implementing the entire workflow for cartesian coordinates and the transformed dihedral coordinates, now you will have two sets of results. Based on this, please compare and contrast the outcome of your analysis in both coordinate spaces. You may use the help of generated plots in order to compare the two results (cumulative variance plot and 3D scatter plot of projections). [5 points]

- List some of the drawbacks of PCA on this dataset. [Hint: refer to the histogram of deviations plot]. [5 points]

Question 5: Going beyond second-order statistics to characterize atomic fluctuations [30 points]

In this section, you will get a chance to explore the non-orthogonal behavior of protein motions through fourth-order statistics based on *kurtosis* by applying Independent Component Analysis (ICA). The main idea here is for you to appreciate the anharmonic behavior of atomistic fluctuations and harness the power of higher-order statistics to deal with such datasets.

Based on the theory that you learnt in class, for your convenience, we will be providing you with a python script that performs spatial decorrelation in fourth-order (*SD4* module). You can download the SD4 module here. The second-order projections Y and the whitening matrix U obtained from the SD2 module will be used to build a fourth-order spatially correlated cumulant tensor. The SD4 module approximately diagonalizes this tensor to return a fourth-order spatially resolved matrix.

Upon executing the SD4 module, you will obtain modes that are statistically ordered based on the kurtosis of the projected coordinates. Similar to the previous question, your task is to generate a $3D$ scatter plot for visualizing the top 3 dominant ICA modes. Here, we will be choosing RMSD to paint the reaction coordinates just like we did in the previous question. Following is a code snippet to perform SD4. We will be working with $m = 20$ as the dimensionality of subspace that is of interest.

```
import SD4

# perform SD4
W = SD4.SD4(Y[0:m,:], m=20, U=U[0:m,:])

# obtain data projections on SD4 modes
Z = W.dot(X)

# a 3D plot of the projections obtained from SD4 modes
Y_sd4 = np.asarray(Z)

fig = plt.figure()
ax3D = fig.add_subplot(111, projection=’3d’)
p3d = ax3D.scatter(Y_sd4[0,::], Y_sd4[1,::], Y_sd4[2,::], s=30, c=RMSD,
    marker=’o’)

ax3D.set_xlabel(r’$\bf{SD4_1}$’);
ax3D.set_ylabel(r’$\bf{SD4_2}$’);
ax3D.set_zlabel(r’$\bf{SD4_3}$’);
cbar = fig.colorbar(p3d, ax=ax3D)
```

Following your understanding based on the implementation above, please answer the following questions.

- After generating the scatter plot $p3d$ based on SD4 modes of motion, what can you gather from the spatial arrangement of the $3D$ projections? Please provide an explanation based on your visual interpretation of the plot. Next, relate your visual interpretation of the scatter plot to the collective behavior of residues in Fs-peptide. [5 points]
- Please compare the two $3D$ scatter plots generated in Q4. and Q5. Can you provide a reasonable explanation behind the differences in the spatial arrangement of conformations projected in three-dimensional space in both the scatter plots. Next, can you analyze and explain how did the usage of fourth-order statistics help in dealing with the shortcomings of second-order statistics? [10 points]
- Once again you will be having two sets of results after implementing SD4 on cartesian coordinates and transformed dihedral coordinates. Please compare the two results from SD4 module like you did for SD2 in Q4. What are you learnng from SD4? [10 points]
- In this assignment we only looked at the fluctuations of atoms at a spatial scale (in second-order by doing SD2 and fourth-order by performing SD4). Do you think it is necessary to think in terms of other scales? If so, what would be an ideal approach to address this? [5 points]

Note: The entire workflow presented in Q5. should be performed both on cartesian coordinates and dihedral coordinates.