dPCA and Cluster Analysis of CPs

.....

1. Template file location

A completed dPCA and cluster analysis is located at our GitHub repo: https://github.com/ysl-lab/Lab-tools/tree/main/dPCA_and_Clustering"

- In this guide this directory will be referred to as the Reference Directory
- Use this directory as a reference/to get files that perform analysis

1.1 Template file location

- Create a dPCA folder for the cyclic peptide you are simulating
- cd into dPCA
- Create a folder for the time span you want to analyze
    - For an analysis of the trajectory from 50-100ns create a folder called: 50-100ns

2. Dihedral Principal component analysis

Note 1: This part is performed on the cluster.
Note 2: Currently this is set up to do dPCA for s1 and s2 using 5 neutral replicas. The numbering of the replicas will be different for the size of the CP used (the neutral replicas for a 5mer are numbered 10-14 (indexed from zero))

2.1 Create a raw_traj directory in the 50-100ns folder

- cd raw_traj
- Make a directory for s1 and s2. I normally name them s1cPROT and s2cPROT (where PROT is replaced with your protein sequence)
- Go to your production run folder for s1 and s2 [Edit by Francini: check the completion of

your bemeta simulation by following Part 5: Checking that your simulation completed successfully in the tutorial Running Be-Meta Simulations on SLURM HPC with Gromacs/4.6.7]

    - Create a trimmed trajectory for your neutral replica .xtc files using the following command:
        - gmx_mpi trjconv -f prod1?_100ns.xtc -s prod1?_100ns.tpr -o prod1?_50_100ns.xtc -b 50001 -e 100000 -pbc mol
            - This command will rewrite your inputted trajectory from time 50001 ps to 100000 ps using just the protein molecule [Edit by Francini: use the .tpr files created after the last equilibration run]
            - Repeat this command for all neutral replicas for both s1 and s2

- Copy the trimmed trajectory files (i.e. protein only) into s1cPROT and s2cPROT for the five neutral replicas
    - Naming: prod10_50_100ns.xtc, prod11_50_100ns.xtc, etc
- In each directory, trajectory cat the five neutral replicas together
    - gmx_mpi trjcat -f prod1?_50_100ns.xtc -cat -nosort -o s1cPROT_all.xtc (or s2cPROT_all.xtc)

2.2 Main dPCA_3D directory

- cd back into the 50-100ns directory
- Write the dPCA.ndx file for PCA (Refer to http://www.gromacs.org/Documentation/How-tos/Dihedral_PCA for additional information on why this is done)
    - Example: For 10 dihedrals, each dihedral as a sin and cos element, so total we have 10x2 = 20 coordinates. Since these are stored in the x,y,z components of the dpca files, we need at least 20/3 = 7 atoms to store these coordinates
    - dPCA.ndx file contents:
        - [dummy]
        - 1 2 3 4 5 6 7
- Copy a .gro file of your cyclic peptide and delete all of the water molecules to create a protein only .gro file
- Copy the protein only .gro file (prot.gro) to this directory (/50-100ns) and create a dpca.gro file using the following command:
    - gmx_mpi trjconv -f prot.gro -n dpca.ndx -s prot.gro -o dpca.gro
- Use VMD_GenPhiPsiIndex.sh (copied from the Reference Directory) to generate the output index.ndx file containing the backbone phi/psi angles (vmd -e VMD_GenPhiPsiIndex.sh)
    - Reads in prot.gro, need to change the number of residues in the script (“set numRes 5”) to match the number of residues in your cyclic peptide [Edit by Francini: creating an index file that contains each atom number involved in the 10 phi and psi dihedrals of interest]
    -

- Edit driver_GenDihedTraj.sh (copied from the Reference Directory) and subsequently run this shell script
    - Change "`prot`" to your protein sequence
    - Make sure your replica numbering is correct (`it` is set to the index of the first neutral replica - 10 in this case and `NT` is set to total number of replicas - 15 in this case)
    - This will create the dihed_traj directory and calculate both s1 and s2 and output files in dihed_traj [Edit by Francini: a .trr file is created containing cos and sin of selected dihedral angles, which subsequently can be used as input for a principal components analysis]

2.3 dihed_traj directory

- Edit Sh_combine_trr.sh (copied from the Reference Directory) and subsequently run this shell script
    - Change "PROT" to your sequence
    - This outputs s1PROT_all.trr s2PROT_all.trr and all.trr

2.4.1 Create a phipsi/ directory in the 50-100ns/ directory

2.4.2 Create a covar/ directory inside 50-100ns/phipsi/

- cd into phipsi/covar
- Run CalcCOVAR.sh (copied from the Reference Directory)
    - Outputs the eigenvec.trr file needed in phipsi/projection directory
    - NOTE: For GROMACS versions 4.6.6, 4.6.7, and 5.0.2 the g_covar_mpi command will result in a Segmentation Fault if your dpca.gro file has more atoms than your trajectory has frames

2.4.3 Create a phipsi/projection/ directory

- Edit CalcProject.sh and subsequently run this shell script (copied from the Reference Directory)
    - Change PROT to your sequence
    - Creates pc1_pc2_pc3 directory and two xvg files for all/s1/s2
- Copy driver.sh and Py_combine.py from the Reference Directory
- Edit driver.sh and subsequently run this shell script
    - Change PROT to your sequence
    - Combines xvg files for all/s1/s2 and outputs all.txt/s1cPROT.txt/s2cPROT.txt, which are the inputs for cluster analysis
- cd back into 50-100ns/ and create a directory called cluster_analysis
- Copy phipsi/projection/pc1_pc2_pc3/*.txt to cluster_analysis/

3. Cluster analysis

- Transfer the cluster_analysis directory to your laptop

- NOTE: If you load the matlab module on the cluster, this section can be done without downloading files to your laptop

3.1 cluster_analysis directory

- Run Py_write_dPCA_min_max.py (copied from the Reference Directory)
    - python Py_write_dPCA_min_max.py all.txt PROT TIME CLEAN
        - PROT is your peptide sequence, TIME is the time segment, and CLEAN is the density value below which is cleaned off
    - Example: python Py_write_dPCA_min_max.py all.txt GNSRV '150-200ns' 0.1 [Edit by Francini: make sure to pass the arguments]
    - Outputs driver_s1.sh and driver_s2.sh
- In the cluster_analysis/ directory create an s1/ and s2/ directory
- For s1 and s2, copy the corresponding txt file from dPCA and driver into the s1 (s2) directory
- In both the s1 and s2 directories, copy *.py, *.gplt, and *.m from the Reference Directory/s*/ folder
- In both the s1 and s2 directories, bash the driver_s*.sh
    - NOTE: you may need to edit the first line of the driver_s*.sh script to point to the actual location of the bash shell

3.2 Matlab

In matlab, first change to your current working cluster_analysis/s1 (or cluster_analysis/s2) directory

- This step needs to be completed for both s1 and s2.
- In the command window, type Mt_cluster_dp. It will prompt you for the name of the distance matrix file (either s1cPROT_kept.dmtx or s2cPROT_kept.dmtx)
- A window will pop up and allow you to pick your cluster centers. I usually pick all points that are above a value of 0.5 on the y-axis
- Outputs CLUSTER_ASSIGNATION

3.3 cd back into the cluster_analysis/ directory

- Run Py_write_dPCA_assign_fortran.py (copied from the Reference Directory)
    - python Py_write_dPCA_assign_fortran.py all.txt
    - Outputs Assign.f90
- In the s1 and s2 directories create a struct/ directory
- Copy Assign.f90 to s1/struct/ and s2/struct/
- cd into your s1 and s2 directories
- For both s1 and s2, complete the following:
    - Edit the driver_s*.sh. On the bottom, comment out the two lines that say "calc_den #step1" and "clean #step2". Uncomment the line that says "calc_pop &> populations.txt"
    - bash driver_s*.sh
        - Outputs populations.txt, which contains the population of all clusters

(Note: these are not sorted)

- cd to the the struct/ directory in your respective s*/ folder:
  - Create executable for Assign.f90
    - g95 Assign.f90 -o assign
      [Edit by Francini: here you might need to module load g95 first and then run gfortran Assign.f90 -o assign. That will create the executable assign.]
  - Run the fortran file: ./assign ../s*cPROT.txt assignments.txt (assignments.txt is the output; an assignment for each frame of your original xtc file)
  - Copy driver.sh and GenGromacsIndex.py from the Reference Directory/s*/struct/) to your struct/ directory
  - bash driver.sh
    - Outputs cluster.ndx, which is used to generate the xtc files for each cluster using gromacs

- On the cluster, create a 50-100ns/cluster_traj/ directory
- In your cluster_traj directory, create an s1cPROT/ directory and an s2cPROT/ directory
- Transfer the cluster.ndx for s1 and s2 back to the cluster in cluster_traj/s1/ and cluster_traj/s2/ directories, respectively

4. Make cluster trajectory files and ramachandran plots

Move back to working on the cluster for the following steps

4.1 cd into the cluster_traj directory

- For both s1 and s2, complete the following:
  - Copy Sh_make_cluster_xtc.sh from the Reference Directory to cluster_traj/s*cPROT/ and run Sh_make_cluster_xtc.sh
    - Change “max_cluster” to the number of clusters from cluster analysis
    - Outputs cluster1.xtc, cluster2.xtc...cluster5.xtc (cluster1.xtc is the most populated cluster)
    - [Edit by Francini: Replace NUMCLUSTERS by the actual number of clusters you are working with. Check the .txt files for eventual errors if output is not cluster1.xtc, cluster2.xtc...cluster5.xtc]

4.2 In the 50-100ns/ directory create the rama/ directory

- Generate an ndx file (index.ndx) containing the backbone phi/psi angles for all residues
  - This should be the same index file generated in step 2.2, located in your 50-100ns/ directory
- Edit driver_calc_dihed.sh (copied from the Reference Directory) and subsequently run the shell script
  - Change “PROT” to your peptide sequence
  - Outputs directories for s1cPROT_phipsi and s2cPROT_phipsi, containing the xvg

files for the five most populated clusters

- Run Clean.sh (copied from the Reference Directory)
    - This removes the comment lines from all the xvg files in s1cPROT_phipsi and s2cPROT_phipsi, and outputs txt files in both directories
- Copy calc_rama.sh and calcFreeEnergy_v2_5mers.py from the Reference Directory
- Edit calc_rama.sh and subsequently run the shell script
    - Change “PROT” to the protein sequence
    - Outputs png files containing the ramachandran plots for a 5 residue CP (calcFreeEnergy_5mers.py)

Note 1: The ramachandran plots are actually density, not free energy.
Note 2: If you would like to edit the calcFreeEnergy*.py file that outputs the ramachandran plot for a different number of residues, you need to change two lines in the python:

1) “cvvals = np.loadtxt…” to use the correct number of columns. I.e. 2–11 for a 5mer, 2–13 for a 6mer, etc.
2) MakeFigure(10,2, INP, OUT). The 10 and 2 are the width and the height of the output figure, respectively. (10, 2) works nicely for a 5mer. You will want to scale it appropriately so each of your ramachandran plots is (2 x 2 in). I.e. (12,2) for a 6mer and (14,2) for a 7mer.